Elaboració de pàgines web amb llenguatges de marques

Isidre Guixà i Miranda IES SEP Milà i Fontanals, d'Igualada

Manteniment de portals d'informació

Setembre del 2007 © Isidre Guixà i Miranda IES SEP Milà i Fontanals C/. Emili Vallès, 4 08700 - Igualada

En cas de suggeriment i/o detecció d'error podeu posar-vos en contacte via el correu electrònic iguixa@xtec.cat

Cap part d'aquesta publicació, incloent-hi el disseny general i de la coberta, no pot ser copiada, reproduïda, emmagatzemada o tramesa de cap manera ni per cap mitjà, tant si és elèctric, com químic, mecànic, òptic, d'enregistrament, de fotocòpia, o per altres mètodes, sense l'autorització prèvia per escrit dels titulars del copyright.

Índex

| Índex | | 3 |
|-----------|---|------|
| Introduce | ió | 7 |
| Objectius | | 9 |
| 1. Ll | lenguatge HTML per al disseny de pàgines web | 11 |
| 1.1. | L'hipertext | 11 |
| 1.2. | Marques | 12 |
| 1.3. | Definicions en l'entorn de l'HTML | 13 |
| 1.4. | Marques per a estructurar la pàgina web | 15 |
| 1.5. | Marques per al format bàsic del text | 17 |
| 1.5.1. | Marques per al format de paràgrafs | 17 |
| 1.5.2. | Marques per a definir apartats: capçaleres i separado | or18 |
| 1.5.3. | Marques per a gestionar la font i l'estil del text | 20 |
| 1.5.4. | Marques per a gestionar caràcters especials | 21 |
| 1.6. | Marques per a gestionar enllaços | 25 |
| 1.7. | Marques per a gestionar imatges | 28 |
| 1.8. | Marques per a gestionar taules o graelles | 30 |
| 1.9. | Marques per a gestionar llistes | 33 |
| 1.10. | Una eina d'edició de pàgines web: Nvu | 35 |
| 1.10.1 | l. Interfície | 36 |
| 1.10.2 | 2. Disseny de pàgines web | 40 |
| 1.10.3 | 5. Format bàsic del text | 45 |
| 1.10.4 | 4. Gestió d'enllaços | 50 |
| 1.10.5 | 5. Gestió d'imatges | 51 |
| 1.10.6 | 6. Gestió de taules o graelles | 54 |
| 1.10.7 | 7. Gestió de llistes | 59 |
| 1.10.8 | 3. Ortografia | 59 |
| 2. Aj | profundiment en el disseny de pàgines web | 60 |
| 2.1. | Estils i fulls d'estil | 60 |
| 2.1.1. | Sintaxi bàsica dels fulls d'estil | 62 |
| 2.1.2. | Assignació de fulls d'estil externs a documents | 64 |
| 2.1.3. | Definició de fulls d'estil interns en un document | 66 |
| 2.1.4. | Definició de fulls d'estil en Nvu | 67 |
| 2.1.5. | L'efecte "cascada" dels fulls d'estil | 72 |
| 2.2. | Recursos multimèdia tradicionals | 73 |
| 2.2.1. | Mapes sensibles | 73 |
| 2.2.2. | Incorporació de so via enllaços | 75 |
| 2.3. | Objectes genèrics | 78 |
| 2.3.1. | Com utilitzar la marca <object></object> | 78 |
| 2.3.2. | Tipus MIME | 80 |
| 2.3.3. | Connectors | 82 |
| 2.3.4. | Exemples d'incorporació d'objectes genèrics | 83 |

| | 2.4. | Formularis | 85 |
|----|--------|--|-----|
| | 2.4.1. | Propietats d'un formulari | 87 |
| | 2.4.2. | Camps de text | 90 |
| | 2.4.3. | Botons d'opció | 91 |
| | 2.4.4. | Caselles de selecció | 92 |
| | 2.4.5. | Àrea de text | 92 |
| | 2.4.6. | Botons | 93 |
| | 2.4.7. | Llistes de selecció | 94 |
| | 2.4.8. | Observacions finals | 97 |
| | 2.5. | Plantilles | |
| | 2.5.1. | Dissenyar plantilles en Nvu | |
| | 2.5.2. | Utilització de plantilles en Nvu | 102 |
| | 2.6. | Disseny d'interfícies | 103 |
| | 2.6.1. | Patrons | 104 |
| | 2.6.2. | El model 3P | 105 |
| | 2.6.3. | Mides de la interfície | 107 |
| | 2.7. | Marcs | 107 |
| | 2.7.1. | Marca <frameset></frameset> | 108 |
| | 2.7.2. | Marques < frame > i < noframes > | 110 |
| | 2.7.3. | Marcs en Nvu | 111 |
| 3. | . Н | TML dinàmic amb JavaScript | 112 |
| | 3.1. | JavaScript bàsic | 112 |
| | 3.1.1. | El primer programa JavaScript | 113 |
| | 3.1.2. | Per què programar? Necessitats que se'n deriven | 115 |
| | 3.2. | Variables i tipus de dades | 116 |
| | 3.3. | Operadors | 118 |
| | 3.4. | Estructures repetitives | 121 |
| | 3.4.1. | Bucle for | 121 |
| | 3.4.2. | Bucle while | 124 |
| | 3.4.3. | Bucle dowhile | 125 |
| | 3.4.4. | Instruccions break i continue | 125 |
| | 3.5. | Estructures condicionals | 127 |
| | 3.5.1. | Estructura condicional if simple | 127 |
| | 3.5.2. | Estructura condicional if composta | 130 |
| | 3.5.3. | Estructura condicional switch | 131 |
| | 3.6. | Funcions | 132 |
| | 3.6.1. | Com escriure, situar i cridar una funció? | 133 |
| | 3.6.2. | Utilització de paràmetres | 134 |
| | 3.6.3. | Valors de retorn | 136 |
| | 3.7. | Àmbit de les variables. Variables locals i globals | 138 |
| | 3.8. | Esdeveniments | 140 |
| | 3.9. | Objectes i mètodes | 143 |
| | 3.9.1. | Objecte Date | 143 |
| | 3.9.2. | Objecte Math | 144 |
| | 3.10. | El Model d'Objectes del Document (DOM) | 145 |
| | 3.11. | Acció de JavaScript en formularis | 146 |
| | | | |

| 3.12. | Exemples d'scripts molt emprats | |
|--------|----------------------------------|--|
| 3.12.1 | Darrera modificació d'una pàgina | |
| 3.12.2 | Botó per a tornar | |
| 3.12.3 | Data catalana | |
| 3.12.4 | Hora actual | |
| 3.12.5 | Rellotge | |

Introducció

Internet, la xarxa de comunicacions que posa en contacte milions de sistemes informàtics repartits per tot el planeta, facilita diversos serveis als seus usuaris: correu electrònic (e-mail), xerrades (xat), pujar i baixar fitxers (FTP), taulers d'anuncis (news), vídeo conferència i, el seu servei més utilitzat, la teranyina de documents (pàgines web), formada per un immens conjunt de documents que poden tenir contingut textual, gràfic, àudio, animacions, etcètera.

En les unitats didàctiques precedents heu anat treballant els conceptes vinculats amb l'explotació d'un lloc web així com els procediments a emprar per actualitzar-ne els continguts i la normativa legal vigent. En aquesta unitat didàctica aprendreu com mantenir i dissenyar, d'una forma correcta, pàgines web que siguin atractives i entenedores, de forma que hi hagi un alt grau de probabilitat de ser visitades per molts usuaris de la xarxa.

En el nucli d'activitat "Llenguatge HTML per al disseny de pàgines web" coneixereu el llenguatge informàtic HTML, que és el que ha d'utilitzar el dissenyador de la pàgina web per tal informar al navegador encarregat de visualitzar la pàgina, de com vol que aquesta es visualitzi. El llenguatge HTML permet anar inserint, en els continguts, marques que els navegadors saben interpretar, de forma similar a la situació que es produeix quan es dicta a viva veu un document a una persona que el transcriu a màquina i se li indica "inicia subratllat", "inicia negreta", "en majúscula"... (indicacions de visualització).

El primer nucli d'activitat finalitza amb la presentació d'un editor actual de pàgines web, *Nvu*, que facilita la tasca del dissenyador ja que va inserint automàticament les marques del llenguatge HTML des d'una interfície gràfica sense necessitat que el dissenyador les expliciti.

En el nucli d'activitat "Aprofundiment en el disseny de pàgines web" es presenten recursos més sofisticats utilitzats en el disseny de pàgines web: gestió de fulls d'estil, integració d'informació multimèdia, gestió de formularis, disseny d'interfícies,...

I per últim, en el nucli d'activitat "HTML dinàmic amb JavaScript" es veu com aprofitar les facilitats que els navegadors aporten per aconseguir cert dinamisme en les pàgines web (mostri la data del dia, aparegui un missatge segons la franja horària de l'usuari que visualitza la pàgina – estigui en el país que estigui-, ...). No és objecte d'aquesta unitat didàctica, sinó de la següent, aprofitar el dinamisme que els propis servidors web són capaços de facilitar, construint la pàgina web en el moment en que aquesta és sol·licitada des d'un navegador, podent-hi incloure informació actualitzada (és el dinamisme que es necessita en les pàgines web que permeten l'adquisició de productes en línia).

Per aconseguir els nostres objectius és necessari que aneu reproduint en el vostre ordinadors tots els exemples incorporats en el text així com les activitats i els exercicis d'autoavaluació. També és aconsellable, com no, la lectura dels annexos.

Objectius

En acabar la unitat didàctica heu de ser capaços del següent:

- 1. Aplicar el llenguatge de programació de pàgines web adient a les característiques del maquinari, els requisits de la comanda i la tecnologia disponible.
- 2. Elaborar pàgines web amb llenguatges de marques, amb l'ajut d'eines específiques per desenvolupar pàgines web, incloent-hi guions *(scripts)* de navegador, i realitzant la verificació del funcionament.

1. Llenguatge HTML per al disseny de pàgines web

En el disseny de pàgines web s'utilitza el llenguatge de marques HTML que és reconegut per tots els navegadors i, per tant, és l'eina de que disposa el dissenyador de pàgines web per informar al navegador sobre l'estructura de la pàgina web o la seva presentació.

Un llenguatge de marques és una forma de codificar un document consistent en que el text va acompanyat d'etiquetes o marques que contenen informació addicional sobre l'estructura del text o la seva presentació.

Històricament el marcatge s'utilitzava i s'usa en la indústria editorial i de la comunicació així com entre autors, editors i impressors.

1.1. L'hipertext

L'**HTML**, acrònim anglès de *Hyper Text Markup Language* (llenguatge de marques hipertextuals), ha estat dissenyat per estructurar textos i presentar-los en forma d'hipertext, que és el format estàndard de les pàgines web. Gràcies a Internet i als navegadors com *Mozilla Firefox*, *Internet Explorer*, *Opera* o *Netscape*, l'HTML s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents i també dels més fàcils d'aprendre.

L'hipertext és un sistema d'organització de la informació basat en la possibilitat de moure's per dins d'un text i cap a textos diferents per mitjà de paraules clau.

El mot hipertext el va inventar Ted Nelson en els anys 60 quan va iniciar un ambiciós projecte inacabat anomenat *Xanadu* que havia de ser un enorme arxiu a escala mundial amb múltiples connexions entre els documents.

En informàtica, l'**hipertext** és una fita a assolir en la interfície de l'usuari, que té com a finalitat presentar documents que puguin, segons la definició de Ted Nelson, "bifurcar-se o executar-se quan sigui sol·licitat" (*branch or perform on request*). La forma més habitual d'hipertext en documents és la d'hipervincle o referències encreuades automàtiques que van a altres documents. Si l'usuari selecciona un hipervincle, fa que el programa informàtic mostri el document enllaçat en un curt període de temps.

La versió actual del llenguatge HTML és la 4.01 (any 1999) i, segons el *Word Wide Web Consortium (W3C)*, organisme regulador dels estàndards de la web, sembla que no ha d'aparèixer cap nova versió. El *W3C* va proposar, l'any 2000, el nou estàndard XHTML (acrònim anglès de *eXtensible HyperText Markup Language*), basat en el llenguatge XML, com a successor del llenguatge HTML. És molt possible que en els propers anys el llenguatge XHTML s'imposi però, de moment, el llenguatge HTML encara té corda per molt de temps.

Per conèixer l'evolució històrica del llenguatge HTML des dels seus inicis fins l'actualitat del XHTML, vegeu la secció "Annexos" del web d'aquest crèdit.

1.2. Marques

Les pàgines web escrites en HTML són fitxers plans editables amb qualsevol editor de text pla (no confondre amb els processadors de textos) que contenen marques.

Diferència entre processador de text i editor de text

Els textos enregistrats amb un processador de textos incorporen marques de format (negreta, cursiva, salt de pàgina, tipus de font, grandària de font,...) que són necessàries per saber com s'ha de visualitzar aquell text. Aquests textos només poden ser gestionats amb el processador de textos amb què s'han creat o amb processadors compatibles.

Els textos enregistrats amb un editor de textos no incorporen cap marca i, per això, també s'anomenen fitxers plans i són llegibles per qualsevol aplicació. Fins i tot, des de la consola del sistema operatiu es pot visualitzar el seu contingut amb la corresponent instrucció del sistema (*cat* en els sistemes operatius Linux, *more* o *type* en els sistemes operatius basats en DOS,...)

Exemples de processadors de textos són *OpenOffice Writer* i *Microsoft Word*. La majoria de processadors de textos també permeten enregistrar textos en format pla (opció "guardar com") però no és l'opció d'enregistrament predeterminada. Exemples d'editors de text són *gedit* i *vi* en sistemes operatius Linux i *notepad* i *edit* en sistemes Windows.

Les marques (altrament anomenades etiquetes o elements, *tags* en anglès) utilitzades en el llenguatge HTML són textos tancats entre els signes < >. Generalment, funcionen per parelles, una en començar i l'altra al final. Normalment, la marca de tancar és la mateixa que la d'obrir amb el caràcter / al davant.

És aconsellable complir les següents normes que faran que el nostre codi HTML sigui més proper a l'estàndard XHTML:

• Sempre que sigui factible utilitzarem les minúscules per a escriure les marques.

- Els atributs que puguin acompanyar a algunes marques els escriurem entre cometes ('simples' o "'dobles" indistintament).
- L'obertura i tancament de marques es pot anar imbricant però s'ha de ser acurat i no tancar una marca si després de la seva obertura se n'ha obert una altra que encara no està tancada.

Aquestes normes no són obligatòries en HTML però la seva utilització facilita la interpretació dels navegadors i el pas a XHTML es més senzill. Exemples d'utilització correcta de marques:

<X> text1 </X> <Y> text2 </Y> <X> text3 <Y> text4 </Y> </X>

Observeu que:

- text4 està afectat per les marques X i Y simultàniament.
- La marca Y està imbricada dins la marca X doncs quan s'obre Y encara no s'ha tancat X
- La imbricació de la tercera línia és correcta doncs Y es tanca abans de tancar-se X.

El següent exemple mostra una imbricació incorrecta doncs x es tanca quan encara no s'ha tancat Y:

<X> text3 <Y> text4 </X> </Y>

1.3. Definicions en l'entorn de l'HTML

En l'entorn HTML s'acostuma a utilitzar uns determinats mots pels que cal conèixer la seva definició.

Autor és una persona o programa que escriu o genera documents HTML.

Les eines de creació (programes generadors d'HTML) són autors.

Agent d'usuari HTML és qualsevol dispositiu que interpreti documents HTML.

El concepte "agent d'usuari" inclou navegadors visuals (textuals o gràfics), navegadors no visuals (audio, Braille), robots de recerca,...

Usuari és una persona que interactua amb un agent d'usuari per a veure, escoltar o utilitzar de qualsevol altra manera un document HTML representat.

Cal tenir en compte aquestes definicions (extretes de la recomanació de *W3C*) malgrat que en moltes ocasions utilitzem el terme "navegador" enlloc de "agent d'usuari" i el terme "visualització" enlloc de "representació".

Un element o atribut desaprovat és aquell que ha quedat antiquat per la presència d'estructures noves.

Els elements desaprovats es defineixen en el manual de referència en llocs ben visibles. Cal tenir en compte que poden declarar-se obsolets en versions futures d'HTML.

Els agents d'usuari haurien de continuar donant suport als elements desaprovats per raons de compatibilitat amb versions anteriors.

En la primera part d'aquest nucli d'activitat emprareu marques que ja estan desaprovades però la seva utilització és necessària mentre no conegueu les noves estructures que han permès la seva desaprovació.

Una marca (altrament anomenada element) o atribut obsolet és aquell pel que no hi ha garantia de suport pels agents d'usuaris.

Els elements obsolets acostumen a estar enumerats, per motius històrics, en el manual de referència del llenguatge.

I ara que ja esteu força introduïts en el llenguatge HTML, comentem que les marques que facilita es poden classificar en diferents tipus segons la seva funcionalitat:

- 1) Marques per a estructurar la pàgina web
- 2) Marques per al format bàsic del text (paràgrafs, font, color,...)
- 3) Marques per a gestionar enllaços
- 4) Marques per a gestionar imatges

El concepte de desaprovació de certes funcionalitats es va començar a introduir en la versió 4 d'HTML.

- 5) Marques per a gestionar taules o graelles
- **6)** Marques per a gestionar llistes

El llenguatge HTML és molt ampli i el que presentem en aquest text n'és simplement una bona introducció. Per qualsevol dubte en la utilització dels elements d'aquest llenguatge cal dirigir-se a la documentació oficial proposada pel *World Wide Web Consortium*.

1.4. Marques per a estructurar la pàgina web

El llenguatge HTML facilita un conjunt de marques bàsiques per estructurar una pàgina web. La taula l ens les presenta. S'hi pot observar l'obligatorietat de la seva utilització en la versió 4.01 d'HTML.

Taula 1. Marques bàsiques per a estructurar una pàgina web

| Marques | Significat Marca inicial en HTML 4.01 | | Marca final en HTML 4.01 |
|------------------|--|-------------|-----------------------------|
| <html> </html> | Inici i final de la pàgina web | Opcional | Opcional |
| | Inici i final de capçalera | | |
| <head> </head> | La capçalera és l'espai destinat a contenir informació sobre el propi document. Així, per exemple, el títol va dins seu. | Opcional | Opcional |
| (hody) | Inici i final del cos de la pàgina web | Opeienel | Opcional |
| | És l'espai que conté tota la informació que visualitza la pàgina web. | Opcional | |
| | Inici i final del títol de la pàgina web. | | |
| <title> </title> | És el text que apareix a la barra del títol de la pàgina web quan es visualitza en el navegador i també és el text que s'emmagatzema en els <i>bookmarks</i> . | Obligatòria | Obligatòria |

L'HTML incorpora la possibilitat d'introduir comentaris dins la pàgina web que no seran interpretats per cap agent d'usuari, és a dir, no seran visualitzats en cap navegador. Els comentaris són observacions que el dissenyador de la pàgina pot voler incloure en el codi per a diferents motius: deixar una marca en algun lloc per a detectar la ubicació en un futur, explicar quan s'ha inclòs un determinat apartat en una pàgina,...

La marca per introduir un comentari en HTML és <!-- ... --> on els punts suspensius indiquen la ubicació del comentari.

Així doncs ja esteu en condicions de dissenyar la vostra primera pàgina web. Seguiu les passes següents:

1) Preneu un editor de textos i escriviu el següent:

```
<html>
<head>
<title> La meva primera pàgina web!!! </title>
</head>
<body>
Hola <!-- Això és un comentari que no apareixerà -->Món!!!
</body>
</html>
```

Bookmarks

Terme anglès utilitzat per referir-se als enllaços a pàgines web que hem emmagatzemat en el nostre navegador pel seu interès o per la seva posterior visualització.

Vegeu una còpia del fitxer HolaMon.html a la secció "Recursos" del web d'aquest crèdit.

. I

Per consultar la documentació oficial proposada pel W3C, vegeu la secció "Annexos" del web d'aquest crèdit.

2) Enregistreu el fitxer amb un nom qualsevol amb extensió *.html* o *.htm* (com *HolaMon.html*). No utilitzeu accents ni caràcters especials.

3) Visualitzeu aquesta pàgina web, aconseguint un resultat similar al de la figura 1. Ho podeu fer de diferents maneres:

- Des del navegador amb l'opció "obrir arxiu" que acostumen a incorporar i cercant l'arxiu dins l'indret on l'hagueu enregistrat.
- Des de l'explorador d'arxius, seleccionant i obrint l'arxiu. Normalment els sistemes operatius estan configurats per a que els arxius amb extensió .htm o .html s'obrin amb algun dels navegadors instal·lats.

Figura 1. Visualització de la meva primera pàgina web.

| 😻 La meva primera pàgina web!!! - Mozilla Firefox | × |
|--|---|
| Eitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda | 0 |
| The second seco | |
| 🗋 Hotmail gratuito 🗋 Personalizar vínculos 🚰 Windows Media 📄 Windows | |
| (Hola Món!!) | |
| Títol de la pàgina | |
| Text inclòs en | |
| l'apartat BODY | |
| | |
| | |
| Fet | |

El codi HTML anterior s'ha escrit en diferents línies per a fer-lo llegible, però s'hagués pogut escriure tot seguit:

<html><head><title> La meva primera pàgina web!!! </title></head><body>Hola <!-- Això és un comentari que no apareixerà -->Món!!!</body></html>

| | | 0 | |
|--|---|---|--|
| 🕹 La meva primera pàgina web!!! - Mozilla Firefox | | | |
| Etxer Edita Yusuitza b Bitxer Edita Yusuitza b Bigres deii Barra dest Barra dest Barra latgr Hola Mónili Atuga Actualitza | Istorial Adreces d'Inter Istorial Adreces d'Inter nes ion tat al inter Esc Ctrl+R | Irefox as Eijes Ajuda aMon.htm | |
| (hida del te Esti) de pà Codificad <mark>Codi fonc d Pantalla co</mark> | xxt gina de caràcters de la pàgina Ctrl+U mpleta F11 | <html> <html> <html> <title> La meva primera pàgina web!!! </title> </html> <html></html></html></html> | |
| Fet | | | |

Figura 2. Visualització del codi font en el navegador Mozilla Firefox

Atenció

En enregistrar des d'un editor de textos un arxiu, cal vigilar que l'arxiu no quedi enregistrat amb extensió *.txt* darrera de l'extensió que hagueu indicat, doncs en intentar obrir-lo des de l'explorador d'arxius, el sistema operatiu decidirà que és un arxiu de text (en base a la darrera extensió) i l'obrirà amb un editor de textos enlloc d'obrir-lo amb un navegador. La visualització en el navegador és exactament la mateixa que abans (figura 1), però la lectura (i modificació, si s'escau) d'aquest codi és més dificultós que l'anterior. Per tant, és aconsellable estructurar el codi per a fer-lo llegible i fàcilment modificable.

Els navegadors també proporcionen alguna opció per a visualitzar el codi HTML que estan visualitzant. La figura 2 mostra la manera de fer-ho en el navegador Mozilla Firefox.

Proveu a veure el codi font de qualsevol de les pàgines web que accediu normalment.

1.5. Marques per al format bàsic del text

Es pot establir diferents categories dins les marques utilitzades per a formatar el text, de manera que distingirem marques per al format de paràgrafs, marques per a definir apartats (capçaleres i separadors), marques per a gestionar la font i l'estil del text i marques per a gestionar caràcters especials.

1.5.1. Marques per al format de paràgrafs

La taula 2 mostra les marques bàsiques que proporciona el llenguatge HTML per a gestionar els paràgrafs.

| Marques Significat | | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|--------------------|---|-------------------------------|-----------------------------|
| | Serveixen per a delimitar un paràgraf. | Obligatòria | Opcional |
| <center> </center> | Permet centrar tot el text del paràgraf. | Obligatòria | Obligatòria |
| | Està desaprovada. Es pot aconseguir amb <div align="center"></div> | 0 | |
| | Permet definir un bloc de continguts sobre el que es poden definir diferents propietats amb la utilització de diversos atributs. | | |
| | L'atribut align permet indicar l'alineació del text: | | Obligatòria |
| | align = "left", alinea per l'esquerra align = "right", alinea per la dreta align = "center", alinea pel centre align = "justify", alinea en ambdós marges | | |
| <div> </div> | L'atribut title proporciona informació consultiva sobre el bloc. A diferència de l'element title (zona head de la pàgina) que dona informació sobre tot el document i que només pot aparèixer una vegada, l'atribut title pot aparèixer en qualsevol número d'elements. Per saber si un element suporta aquest atribut, cal consultar la definició de l'element. Els valors de l'atribut title poten ser representats pels agents d'usuari de diferents maneres. Per exemple, els navegadors visuals acostumen a representar el títol com un tooltip (missatge curt que apareix quan el ratolí s'atura sobre la zona) | Obligatòria | |
| | Insereix un salt de línia sense canviar de paràgraf. | Obligatòria | Prohibida |

Taula 2. Marques per al format de paràgrafs

En ocasions es confon la utilització de les marques
br> i doncs pot donar la sensació que ambdues serveixen pel mateix, i això és del tot fals. Per a un navegador, un paràgraf indica un bloc independent d'altres i que pot tenir definides unes característiques com, per exemple, l'alineació. En canvi, un salt de línia és simplement una interrupció utilitzada per a canviar de línia seguint dins el mateix paràgraf. La visualització d'un paràgraf va acompanyada d'un espai buit abans i després del paràgraf, mentre que la visualització del salt de línia no porta aquest efecte.

Considereu el següent codi HTML com a exemplificació de les marques presentades. La figura 3 en mostra el resultat. Observeu el *tooltip* que apareix com a resultat de tenir el ratolí damunt el tercer paràgraf.

<html> <head> <title>Exemples de format de paràgrafs</title> </head> <bodv> <div title="Paràgraf 1" align="center">Aquesta pàgina serveix per a exemplificar la utilització de les diferents marques que el llenguatge HTML proporciona per a gestionar paràgrafs. Aquest primer paràgraf està centrat.</div> <div title="Paràgraf 2" align="left">Com s'observa, ara estic en un segon paràgraf, justificat a l'esquerra. Encara que en l'arxiu HTML hi hagi un salt de línia, en la visualització no s'observa donat que continuo en el mateix paràgraf.</div> <div title="Paràgraf 3" align="right">Ara ja estic en el tercer paràgraf, justificat a la dreta. Vaig a inserir un "salt de línia".
br>Ja l'he inserit. Observem, però, que continuem en el mateix paràgraf. Vaig a inserir un altre salt de línia.
Observem que els salts de línia no provoquen separació entre línia anterior i línia posterior, cosa que sí provoquen els canvis de paràgraf. Vaig a canviar de paràgraf.</div> <div title="Paràgraf 4" align="justify">Acabo de canviar de paràgraf. Ja estic en un nou paràgraf, en el qual hem indicat justificació en ambdós marges. Per a comprovar-ho em cal escriure un bon rotllo, doncs del contrari...</div> </body>

</html>

Figura 3. Exemple de format de paràgrafs. 😉 Exemples de format de paràgrafs - Mozilla Firefox Eitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda - 🔹 🕨 💽 - Google Q 📄 Hotmail gratuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windows Aquesta pàgina serveix per a exemplificar la utilització de les diferents marques que el llenguatge HTML proporciona per a gestionar paràgrafs. Aquest primer paràgraf està centrat. Com s'observa, ara estic en un segon paràgraf, justificat a l'esquerra. Encara que en l'arxiu HTML hi hagi un salt de línia, en la visualització no s'observa donat que continuo en el mateix paràgraf. Ara ja estic en el tercer paràgraf, justificat a la dreta. Vaig a inserir un "salt de línia". Ja l'he inserit. Observem, però, que continuem en el mateix paràgraf. Vaig a inserir un altre salt de línia. Observem que els salts de línia no provoquen separació entre línia anterior i línia posterior, cosa que sí provoquen els canvis de paràgraf. Vaig a canviar de paràgraf. Paràgraf 3 Acabo de canviar de paràgraf. Ja estic en un nou paràgraf, en el qual hem indicat justificació en ambdós marges. Per a comprovar-ho em cal escriure un bon rotllo, doncs del contrari. Fet

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina02.html* amb el codi de l'exemple.

1.5.2. Marques per a definir apartats: capçaleres i separador

El llenguatge HTML facilita sis etiquetes per a capçaleres (títols) dins una pàgina web i una etiqueta per a una barra horitzontal separadora.

| Marques | Significat | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|------------|--|-------------------------------|-----------------------------|
| <h1> </h1> | Capçalera de nivell 1 (el més important) | Obligatòria | Obligatòria |
| <h2> </h2> | Capçalera de nivell 2 | Obligatòria | Obligatòria |
| <h3> </h3> | Capçalera de nivell 3 | Obligatòria | Obligatòria |
| <h4> </h4> | Capçalera de nivell 4 | Obligatòria | Obligatòria |
| <h5> </h5> | Capçalera de nivell 5 | Obligatòria | Obligatòria |
| <h6> </h6> | Capçalera de nivell 6 (el menys important) | Obligatòria | Obligatòria |
| <hr/> | Barra horitzontal | Obligatòria | Prohibida |

Taula 3. Marques per a definir apartats: capçaleres i separador

Considereu el següent codi HTML com a exemplificació de les marques presentades. La figura 4 en mostra el resultat.

```
<html>
<head><title>Exemples de capçaleres</title></head>
<body><div align="center">
<hl>Capçalera 1</hl><h2>Capçalera 2</h2><h3>Capçalera 3</h3><hr>
<h4>Capçalera 4</h4><h5>Capçalera 5</h5><h6>Capçalera 6</h6>
</div></body>
</html>
```

Aquestes marques es podrien considerar dins la categoria de marques per al format de paràgrafs, però per la seva importància s'ha preferit tractar-les a banda. No s'han d'utilitzar per augmentar o disminuir la grandària del tipus de lletra, ja que cada navegador les mostra de manera diferent. S'utilitzen per dividir correctament en seccions la pàgina web de la mateixa manera que es fa en un document normal.

```
Figura 4. Exemple de marques de capçalera.
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina03.html* amb el codi de l'exemple.

1.5.3. Marques per a gestionar la font i l'estil del text

El tipus i mida de la font estan definits en la configuració del navegador i alguns navegadors (no tots) accepten que un document pugui modificar aquestes característiques. En els navegadors que no accepten la modificació d'aquestes característiques, aquestes marques no tindran cap efecte.

La taula 4 mostra unes marques bàsiques per gestionar el tipus i mida de la font, així com el seu color i estil.

| Taula 4. Marques | per a gestionar la | font i l'estil del text |
|------------------|--------------------|-------------------------|
| | | |

| Marques | Significat | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|------------------|---|-------------------------------|-----------------------------|
| | Permet definir el tipus de font, la grandària i el color per a tot el document. | | |
| | Està desaprovada en favor de les fulles d'estil. | | |
| | L'atribut face defineix una llista de noms de fonts separats per comes que l'agent d'usuari hauria de cercar en ordre de preferència. | | |
| <basefont/> | L'atribut size especifica la grandària de font. Els seus valors possibles són de l'1 al 7. Això estableix la font en alguna grandària fixa, la representació de la qual depèn de l'agent d'usuari. No tots els agents d'usuari poden representar les 7 grandàries. | Obligatòria | Prohibida |
| | L'atribut color especifica el color del text. Valors possibles: | | |
| | Nom del color. Hi ha 16 noms possibles: black, silver, gray, white, maroon, red, purple, fuchsia, green, lime, olive, yellow, navy, blue, teal, aqua. Valor hexadecimal (iniciat amb #) corresponent al codi RGB del color. | | |
| | Si aquest element no s'indica o si no conté l'atribut size, es pren com a grandària base el valor 3. | | |
| | Permet definir el tipus de font, la grandària i el color per al text del seu contingut (és a dir, el text inclòs entre les marques d'inici i de final). | | |
| | Està desaprovada en favor de les fulles d'estil. | Obligatòria | Obligatòria |
| | L'atribut face defineix una llista de noms de fonts com en basefont. | | |
| | L'atribut size especifica la grandària de font. Valors possibles: | Congatoria | obligatoria |
| | De l'1 al 7 per establir una grandària fixa la representació de la qual depèn de l'agent d'usuari. De +1 a +7 i de -1 a -7 per establir una grandària relativa a l'establerta per l'element basefont (o 3 si no està definit) | | |
| | L'atribut color especifica el color del text, com en basefont. | | |
| | Text en negreta | Obligatòria | Obligatòria |
| <i> </i> | Text en cursiva | Obligatòria | Obligatòria |
| <u> </u> | Text subratllat | Obligatòria | Obligatòria |
| <s> </s> | Text tatxat | Obligatòria | Obligatòria |
| | Text superíndex | Obligatòria | Obligatòria |
| | Text subíndex | Obligatòria | Obligatòria |
| <big> </big> | Incrementa la grandària de font | Obligatòria | Obligatòria |
| <small> </small> | Decrementa la grandària de font | Obligatòria | Obligatòria |

Recurs HTML que preten separar el contingut d'una

pàgina de la visualització. La seva aparició va provocar la desaprovació de moltes

Fulles d'estil

| Marques | Significat | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|------------------|--|-------------------------------|-----------------------------|
| <blink> </blink> | Text parpellejant (que acostuma a ser molt molest) | Obligatòria | Obligatòria |

El següent codi HTML exemplifica algunes de les marques presentades.

La figura 5 en mostra el resultat, tot i que l'efecte parpellejant de la darrera paraula no es pot observar en material imprès.

```
<html>
                                                                       marques.
<head><title>Exemples de format de text</title></head>
<bodv>
En aquest punt no hi ha cap etiqueta "font" i, per tant, actua el navegador per defecte.
<font size="7" color="yellow">Mida 7. Color "yellow".</font>
<font size="6" color="red">Mida 6. Color "red".</font>
<font size="5" color="blue">Mida 5. Color "blue"</font>
<font size="4" color="#FF9900">Mida 4. Color "#FF9900"</font>
<font size="3" color="#33FFCC"><b>Mida 3. Color "#33FFCC". Negreta</b></font>
<font size="+2">Mida +2. Per tant és com si fós 5 doncs no hi ha "basefont"</font>
<u>Subratllat</u><s>Ratllat</s><blink>Parpellejant</blink>
</bodv>
</html>
```

Figura 5. Exemples de marques per gestionar la font i l'estil del text.



A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer pagina04.html amb el codi de l'exemple.

1.5.4. Marques per a gestionar caràcters especials

Les pàgines web visualitzades en els exemples anteriors així com la majoria de pàgines web en llengua catalana i castellana que visitem, contenen un seguit de caràcters especials com les vocals accentuades, la "ç", la "ñ",...

Fixeu-vos que en el paràgraf anterior hem classificat aquests caràcters com a "especials". Per què? En la llengua catalana, què té d'especial la "ç" o la "à" respecte la "n" o la "u"? Des del punt de vista de la llengua, no hi ha un ordre d'importància entre els diferents símbols emprats.

En informàtica, però, si que hi ha una diferència important. Els ordinadors gestionen la informació en format binari (és a dir, en zeros i uns). Les combinacions de zeros i uns formen números que es tradueixen en lletres. Com? Mitjançant taules. Així, podem assignar el valor 64 a la lletra "a", el valor 65 a la lletra "b", etc.

La traducció àmpliament acceptada és la taula ASCII que conté 127 caràcters. Està clar que amb 127 caràcters no n'hi ha prou per representar tots els símbols que incorporen les diferents llengües, i, com que l'evolució de la informàtica sempre ha estat lligat a la llengua anglesa, en els 127 caràcters de la taula ASCII hi ha principalment els símbols emprats en aquesta llengua.

Si existeix aquest problema, com és que visualitzem correctament les pàgines que contenen caràcters especials? La visualització és correcta segons sigui la configuració del navegador. La majoria de navegadors faciliten, en alguna de les seves opcions, la possibilitat de definir la codificació de caràcters a utilitzar. Així, per exemple, en *Mozilla Firefox* ho trobareu en el menú *Visualitza* |*Codificació de caràcters*. Si provem a visualitzar l'exemple de la figura 3 havent habilitat la codificació *Unicode* (*UTF-8*) obtindrem el resultat de la figura 6.

Figura 6. Visualització d'una pàgina amb codificació Unicode (UTF-8)

| 🕲 Exemples de format de par¤grafs - Mozilla Firefox |
|---|
| Eltxer Edita Visualitza Historial Adreces d'interès Eines Ajuda |
| |
| 🕒 Hotmail gratuito 🕒 Personalizar vínculos 🗋 Windows Media 🗋 Windows |
| Aquesta p�gina serveix per a exemplificar la utilitzaci� de les diferents marques que el llenguatge HTML proporciona per a gestionar par�grafs. Aquest primer par�graf est� centrat. |
| Com s'observa, ara estic en un segon par∲graf, alineat per l'esquerra. Encara que en l'arxiu HTML hi hagi un salt de 1∲nia, en la visualitzaci∲ no s'observa donat que continuo en el mateix par∲graf. |
| Ara ja estic en el tercer par 🕈 graf, alineat per la dreta. Vaig a inserir un "salt de l¢nia". Ja l'he inserit. Observem, per �, que continuem en el mateix par � graf. Vaig a inserir un altre salt de l¢nia. Observem que els salts de l¢nia no provoquen separaci � entre l¢nia anterior i l¢nia posterior, cosa que s � Par � graf 3 Acabo de canviar de par � graf. Ja estic en un nou par � graf, en el qual hem indicat alineaci � en ambd � s marges. Per a comprovar-ho em cal escriure un bon rotllo, doncs del contrari |
| Fet |

Això vol dir que la nostra pàgina pot no ser visualitzada correctament des de qualsevol navegador. La correcta visualització dependrà de la configuració del navegador, la qual acostuma a estar d'acord a la regió del planeta on es trobi (configuració regional).

Com que això suposaria un greu problema, HTML facilita mecanismes per a incloure qualsevol dels caràcters especials en les nostres pàgines web de manera que siguin correctament visualitzats independentment de la configuració del navegador. Vegem-ho! Els caràcters especials a incloure en una pàgina web es poden classificar en dues tipologies: estesos i de control.

1) Caràcters estesos

Els caràcters estesos són aquells pels que el corresponent valor numèric està més enllà del 127. La seva codificació es farà mitjançant la norma anomenada *Unicode* que consisteix, en el cas dels caràcters especials de la nostra llengua, en la inserció de noves marques, dins de cada paraula que els continguin, per tal d'obtenir una correcta interpretació en els navegadors. Tenim dues formes de fer-ho:

a) Utilitzant el codi numèric corresponent entre els caràcters &# i ;

Així, per exemple, per representar la "ç" que té el codi ASCII estès 135, hauríem d'escriure ‡, és a dir, la paraula *caçador* dins una pàgina HTML s'escriu com ca‡ ador.

Un altre exemple pot ser la representació del caràcter "½" que té el codi ASCII estès 189. Caldria escriure ½.

b) Hi ha una segona forma per aquells símbols que es construeixen com a combinació de dos caràcters, com són les vocals accentuades ("à" és el caràcter "a" amb accent obert, "í" és el caràcter "i" amb accent tancat,...), la "ç" (és el caràcter "c" amb símbol de trencat), la "ñ" (és el caràcter "n" amb titlla),...

En aquests casos es pot utilitzar una sintaxi sense necessitat de conèixer els codis ASCII corresponents. La sintaxi consisteix en utilitzar el símbol & seguit del primer caràcter i d'un mnemotècnic corresponent al segon caràcter finalitzant amb el ; com es veu en els següents exemples.

| ç: | ç | ñ : | ñ |
|-----|---|-----|---|
| à : | à | í : | í |

Probablement ja haureu intuït els mnemotècnics utilitzats en els anteriors exemples (cedil, tilde, grave, acute,...). Hi ha força mnemotècnics dels quals en presentem els més útils per a nosaltres a la taula 5.

| Taula 5. Mnemotècnics | per a caràcters especials | 3 |
|-----------------------|---------------------------|---|
| | | |

| Caràcter a traduir | Mnemotècnic |
|--------------------|-------------|
| Accent greu | grave |
| Accent agut | acute |
| Dièresi | uml |
| Accent circumflex | circ |
| | |

Manteniment dels portals d'informació

24

| Caràcter a traduir | Mnemotècnic |
|--------------------|-------------|
| Titlla | tilde |
| Trencat | cedil |
| Punt volat | middot |

Refem el contingut de la pàgina web visualitzada a les figures 3 i 6:

<html> <head> <title>Exemples de format de paràgrafs</title> </head> <body> <div title="Paràgraf 1" align="center">Aquesta pàgina serveix per a exemplificar la utilització de les diferents marques que el llenguatge HTML proporciona per a gestionar paràgrafs. Aquest primer paràgraf està centrat.</div> <div title="Paràgraf 2" align="left">Com s'observa, ara estic en un segon paràgraf, alineat per l'esquerra. Encara que en l'arxiu HTML hi hagi un salt de lí nia, en la visualització no s'observa donat que continuo en el mateix paràgraf.</div> <q> -div title="Paràgraf 3" align="right">Ara ja estic en el tercer paràgraf, alineat per la dreta. Vaig a inserir un "salt de línia".
br> Ja l'he inserit. Observem, però, que continuem en el mateix paràgraf. Vaig a inserir un altre salt de lí nia.
 Observem que els salts de lí nia no provoquen separació entre lí nia anterior i lí nia posterior, cosa que sí provoquen els canvis de parà graf. Vaig a canviar de paràgraf.</div> . div title="Paràgraf 4" align="justify">Acabo de canviar de paràgraf. Ja estic en un nou paràgraf, en el qual hem indicat alineació en ambdós marges. Per a comprovar-ho em cal escriure un bon rotllo, doncs del contrari...</div> </bodv> </html>

La visualització d'aquest codi en un navegador configurat amb codificació *Unicode (UTF-8)* és correcta, com a la figura 3.

Ara bé, vaja feina si en escriure hem d'anar introduint aquestes marques... No us espanteu, doncs els editors especialitzats en pàgines web inseriran automàticament aquestes marques mentre escrivim.

2) Caràcters de control

En HTML existeixen quatre caràcters de control, que s'utilitzen per formar etiquetes i establir paràmetres i que és possible que haguem d'utilitzar en el nostre text. Així, per exemple, ja sabem que el caràcter "<" s'utilitza per a indicar l'inici de les marques HTML. Què hem de fer per utilitzar-lo dins el nostre text sense que HTML interpreti que és l'inici d'una marca? Tenim també uns codis per aconseguir-ho:

> <: < &: & >: > ": "

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina05.html* amb el codi de l'exemple.

1.6. Marques per a gestionar enllaços

Els enllaços són dels elements més importants d'una pàgina web.

Un enllaç és una zona textual o gràfica que en ser seleccionada ens trasllada a un altre document o a una altra posició dins el document actual.

Aquests elements són els que possibiliten la hipertextualitat en el llenguatge HTML i donat que HTML és el llenguatge d'Internet, la diferència de l'hipertext d'HTML amb altres tipus d'hipertext és que els documents on enllaça poden residir físicament a l'altra punta del planeta.

La taula 6 mostra l'etiqueta <a> responsable dels enllaços.

Taula 6. Marques per a gestionar enllaços

| Marques Significat | | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|--------------------|--|-------------------------------|-----------------------------|
| | Permet definir enllaços via diversos atributs. | | |
| | L'atribut href permet definir la ruta on ens portarà l'enllaç. | | |
| | L'atribut name permet marcar un destí, dins un document, per a vincles definits per href cap el document | | Obligatòria |
| <a> | L'atribut target permet indicar la finestra on s'ha d'obrir el nou document. Els valors possibles són: | Obligatòria | |
| | <pre>_blank, per a obrir-se en una nova finestra i sense nom _self, per obrir el document en la mateixa finestra _parent, per obrir el document en la finestra mare de la finestra que conté l'enllaç (és equivalent a _self si la finestra actual no té finestra mare) _top, per obrir el document en la primera original (és equivalent a _self si la finestra actual és la principal)</pre> | | |

Així, per introduir en una pàgina web un vincle per anar a una altra pàgina web, escriurem quelcom similar a:

text_i/o_imatge

Per a definir sotsdirectoris en els camins d'accés en HTML s'utilitza sempre el símbol "/" (com en els sistemes Linux).

En visualitzar un enllaç en un navegador, el "text" apareixerà subratllat i en un color diferent a l'habitual i si és "imatge", apareixerà envoltada per una vora del mateix color amb que apareixeria el text.

L'atribut name serveix per a posar marques dins les pàgines web, de manera que qui vulgui enllaçar amb la pàgina pugui indicar, si li interessa, un punt en concret dins la pàgina. És a dir, l'atribut name no

serveix per indicar un enllaç, sinó per posar una marca dins una pàgina web. Un exemple de marca podria ser:

Per a indicar aquest punt marcat en un enllaç, cal utilitzar la sintaxis:

text_i/o_imatge

Les marques d'una pàgina web també poden estar referenciades des de la pròpia pàgina. Llavors només s'indica la marca emprant la sintaxis: text_i/o_imatge

En tot aquest tema cal posar especial atenció en com indicar les direccions on cal anar a cercar el document enllaçat. Hi ha diferents possibilitats:

1) El document a accedir resideixi en un lloc web diferent d'on resideix la pàgina que té l'enllaç.

En aquest cas caldrà indicar la direcció URL absoluta d'on es troba el document.

2) El document a accedir resideixi en el mateix lloc web on resideix la pàgina que té l'enllaç.

En aquest cas la millor opció és utilitzar una ruta relativa a la ubicació actual. Per exemple, suposem l'esquema d'ubicacions de la figura 7.



Figura 7. Exemple d'esquema d'ubicacions de documents en un lloc web

Observem, a la taula 7, exemples de com cal definir les rutes relatives per anar d'uns documents a uns altres en la situació plantejada a la figura 7.

| Taula 7. Rutes relatives per anar d'uns documents a uns altres en la situació de la figu |
|--|
|--|

| Document actual | Document a accedir | Ruta relativa |
|-----------------|--------------------|------------------------------|
| docA02 | docAB02 | carpetaAB/docAB02 |
| docA01 | docB02 | /carpetaB/docB02 |
| docAA01 | docBB02 | //carpetaB/carpetaBB/docBB02 |

Per a exemplificar la gestió d'enllaços podem considerar les dues pàgines web següents i llur funcionament en la figura 8. Per a fer el codi llegible, no s'ha incorporat les marques per als caràcters especials.

Figura 8. Exemple de visualització i gestió d'enllaços entre pàgines web.



A la secció "Recursos" del web d'aquest crèdit podeu trobar els fitxers pagina06.html i pagina07.html amb el codi de l'exemple.

<html>
<html>
<html>
<html>
<html>
<html>
<title>Exemples d'enllaços - Arxiu <pagina06.html></title>
</head>
<body>
Aquesta pàgina està enregistrada amb el nom pagina06.html

En la mateixa ubicació hi ha una altra pàgina de nom pagina07.html
A continuació facilitem un enllaç que ens porti a la pàgina 07

Observem que el text "pàgina 07" té una visualització diferent que indica un enllaç


```
Si cliquem damunt "pàgina 07" anirem al document pagina07.html
En aquest punt facilitem un enllaç que ens porti a la <a href="pagina07.html#anc01">pàgina 07 -
anc01</a>
</body>
</html>
<html>
<head>
<title>Exemples d'enllaços - Arxiu &lt;pagina07.html&gt;</title>
</head>
<body>
Aquesta pàgina està enregistrada amb el nom pagina07.html<br>
En la mateixa ubicació hi ha una altra pàgina de nom pagina06.html
<a name="anc01"></a>A l'inici d'aquest paràgraf hi ha una àncora de nom "anc01".
A continuació facilitem un enllaç que ens porti a l'ancoratge <a href="#anc01">anc01</a> de la
mateixa pàgina.
A continuació facilitem un enllaç que ens porti a la <a href="pagina06.html">pàgina 06</a>
</body>
</html>
```

Per últim, tenir present que el document destí pot ser una altra pàgina web o qualsevol tipus de document textual i/o gràfic; l'únic requeriment per a que es pugui accedir a un document, és que en l'ordinador des d'on es pretén accedir, hi ha d'haver el programari adequat per a poder-lo visualitzar i/o gestionar. Així, per exemple, si es vol accedir a un document d'OpenOffice caldrà que l'ordinador des d'on s'accedeix tingui instal·lat aquest programari o algun de compatible.

1.7. Marques per a gestionar imatges

Oi que moltes vegades hem sentit dir que una imatge val més que mil paraules? Ben segur que hi estem d'acord! Tota publicació, des de la més senzilla fins a la més seriosa, conté imatges, si més no a la portada. Una pàgina web, com a tipus de publicació que és, s'enriqueix enormement si conté gràfics i imatges ben situades.

Les imatges donen vida a una pàgina web, aclareixen el contingut i abreugen molt les coses. No obstant, utilitzar imatges dins una pàgina web té un petit inconvenient, i és que les imatges retarden la descàrrega de la pàgina. Quantes més imatges conté una pàgina, més pesa i més tarda a carregar-se, per això, cal estudiar i veure quin és el format gràfic més adequat per a cada imatge en cada cas.

Hi ha programes específics de tractament d'imatges que en redueixen el pes i que permeten editar i tractar les imatges segons les necessitats de cadascú. Això és així perquè, per poder veure les imatges dins els navegadors convencionals, cal utilitzar un format fàcilment llegible i interpretable pel navegador. Les imatges poden ser de diferents tipus, però les extensions estàndard més comunes millor suportades pels navegadors són les de tipus *.gif* i *.jpg.*

La taula 8 mostra l'etiqueta per la gestió d'imatges.

Taula 8. Marques per a gestionar imatges

| Marques | Significat | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|---------|---|-------------------------------|-----------------------------|
| | Permet incloure gràfics e imatges via la utilització de diversos atributs. | | |
| | L'atribut src especifica el nom i camí de l'arxiu que conté la imatge. | | |
| | L'atribut alt especifica el text que es mostrarà enlloc de la imatge en aquells navegadors que no siguin capaços de mostrar la imatge o si l'usuari ha desactivat la visualització de imatges. Alguns navegadors utilitzen aquest text com a <i>tooltip</i> quan el ratolí passa pel damunt de la imatge. | | |
| | Els atributs height i width permeten indicar al navegador l'alçada i l'amplada de la imatge en pixels. No són imprescindibles, però sí aconsellables doncs el navegador pot mostrar un requadre de la grandària de la imatge mentre aquesta està baixant i, mentrestant, pot mostrar la resta de la pàgina. La grandària original de la imatge queda invalidada i la imatge es mostra proporcional als valors indicats. | Obligatòria | Prohibida |
| | L'atribut border permet indicar el gruix de la vora que el navegador visualitza al voltant d'una imatge si aquesta té un enllaç associat. Si es vol eliminar aquest efecte només cal utilitzar border amb valor zero. | | |
| | L'atribut align permet alinear una imatge respecte el text que l'acompanya. Aquest atribut està desaprovat . Possibles valors són: | | |
| | top, que situa el punt més alt de la imatge coincidint amb el punt més alt de la línia de text actual middle, que situa el punt mig (en alçada) de la imatge amb la base del text bottom, que és el valor per defecte, alinea el punt més baix de la imatge amb la base del text | | |
| | left, que col·loca la imatge a l'esquerra del text right, que col·loca la imatge a la dreta del text | | |

Figura 9. Exemple d'ubicació de imatges 🥹 Exemple d'ubicació d'imatges - Mozilla Firefox Eitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda ▼ ► Google < 🔹 🔹 😪 🚱 🥼 📄 file:///E:/pagina08.html Q 🗋 Hotmail gratuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windows Generalitat de Catalunya Departament d'Educació FP Oberta A l'esquerra veiem el logo de FP Oberta. No hem establert cap alineació. Per tant, actua align = "bottom" III Generalitat de Catalunya Departament d'Educació Iccui A l'esquerra veiem el logo de FP Oberta. **FP Oberta** Hem establert align = "top" Ceneralitat de Catalunya 😡 A l'esquerra veiem el logo de FP Oberta. **FP Oberta** Hem establert align = "middle" III Generalitat de Catalunya Departament d'Educació Recu: A l'esquerra veiem el logo de FP Oberta. FP Oberta Hem establert align = "left" A la dreta veiem el logo de FP Oberta. Generalitat de Catalunya Departament d'Educació Hem establert align = "right" FP Oberta Fet

Exemplifiquem-ho en la següent pàgina web amb la corresponent visualització a la figura 9. Suposem, per simplificar, que la imatge *FPOberta.gif* resideix en la mateixa ubicació que la pàgina web.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina08.html* amb el codi de l'exemple.

```
<html>
<head><title>Exemple d'ubicaci&oacute; d'imatges</title></head>
<body>
<img src="FPOberta.gif" alt="Logo de FP Oberta" heigth="50" width="151">
A l'esquerra veiem el logo de FP Oberta.<br>
No hem establert cap alineació. Per tant, actua align = "bottom"
<img src="FPOberta.gif" alt="Logo de FP Oberta" heigth="50" width="151" align="top">
A l'esquerra veiem el logo de FP Oberta.<br>
Hem establert align = "top"
<img src="FPOberta.gif" alt="Logo de FP Oberta" heigth="50" width="151" align="middle">
A l'esquerra veiem el logo de FP Oberta.<br>
Hem establert align = "middle"
<img src="FPOberta.gif" alt="Logo de FP Oberta" heigth="50" width="151" align="left">
A l'esquerra veiem el logo de FP Oberta. <br>
Hem establert align = "left"
<img src="FPOberta.gif" alt="Logo de FP Oberta" heigth="50" width="151" align="right">
 la dreta veiem el logo de FP Oberta.<br>
Hem establert align = "right"
</body>
</html>
```

1.8. Marques per a gestionar taules o graelles

Les taules o graelles són possiblement la manera més clara d'organitzar la informació, doncs ens permet distribuir el contingut de la pàgina web allà on ens interessi.

No sé si us heu adonat que en totes les pàgines desenvolupades fins el moment, la informació és seqüencial, és a dir, línia sota línia. I si volguéssim incloure text en diferents columnes? La única possibilitat és utilitzar les taules. Fins i tot, per a situar text al costat de imatges és molt més controlat fer-ho amb taules que amb els diversos valors del paràmetre align a l'etiqueta .

La taula 9 presenta les marques per a la gestió de les taules.

| Taula 3. Marques per a la gestio de taule | Taula 9. | Margues | per a la | a gestió d | de taules |
|---|----------|---------|----------|------------|-----------|
|---|----------|---------|----------|------------|-----------|

| Marques Significat | | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|--------------------|--|-------------------------------|-----------------------------|
| | Indica l'inici i el final d'una taula. | | |
| | L'aparença de la taula és configurable per diversos atributs. | | |
| | L'atribut border especifica el gruix de la vora del voltant de les cel·les. Per defecte és zero, és a dir, no es dibuixarà cap vora. | | |
| | L'atribut cellspacing especifica el número de pixels que separen les cel·les. | | |
| | L'atribut cellpadding especifica el número de pixels que hi ha d'haver entre la vora d'una cel·la i el seu contingut. | Obligatòria | Obligatòria |
| | L'atribut width especifica l'amplada de la taula. Pot estar en pixels o en un % que indica quina part de l'amplada total de la finestra es vol utilitzar. L'aparició del símbol % indica que és en percentatge. | | |
| | L'atribut align determina la posició de la taula respecte el document. I pot ser per l'esquerra (left), centre (center) o dreta (right). No té cap efecte si l'amplada de la taula és del 100%. Aquest atribut està desaprovat . | | |
| | Són les marques per indicar l'inici i el final de cada fila de la taula. | | Opcional |
| > | Hi ha d'haver tants elements com files tingui la taula. | Obligatòria | |
| | L'aparença de les files és configurable per diversos atributs. | Congatoria | |
| | L'atribut align alinea el contingut de les cel·les d'una fila | | |

| Marques Significat | | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 |
|----------------------|--|-------------------------------|-----------------------------|
| | horitzontalment per l'esquerra (left), centre (center) o dreta (right). Aquest atribut està desaprovat . | | |
| | L'atribut valign alinea el contingut de les cel·les de la fila verticalment per dalt (top), per baix (bottom) o pel centre (middle). | | |
| | Són les marques per a crear les diferents cel·les. | | |
| | Hi ha d'haver tants elements o com cel·les hagi de tenir la fila. | | |
| | Les etiquetes i són equivalents. La segona s'utilitza per a capçaleres, de manera que el seu contingut s'escriurà per defecte en negreta i centrat. | | |
| | L'aparença de cada cel·la és configurable per diversos atributs. | | |
| | L'atribut align alinea el contingut de la cel·la horitzontalment per l'esquerra (left), centre (center) o dreta (right). Si no existeix actua l'atribut align de la fila (si està definit). Aquest atribut està desaprovat . | Obligatòria | Opcional |
| | L'atribut valign alinea el contingut de la cel·la verticalment per dalt (top), per baix (bottom) o pel centre (middle). | Congatoria | |
| | L'atribut width permet indicar l'amplada de la cel·la. Es pot especificar en pixels o en %. En cas de percentatge aquest es refereix a l'amplada total de la taula. | | |
| | L'atribut colspan especifica el número de cel·les de la fila situades a la dreta de l'actual que s'uneixen a aquesta (incloent la cel·la en la que es declara el paràmetre). | | |
| | L'atribut rowspan especifica el número de cel·les de la columna situades sota de l'actual que s'uneixen a aquesta (incloent la cel·la en la que es declara el paràmetre). | | |
| | Permet assignar un títol a la taula. Només admet un atribut. | | |
| <caption> </caption> | L'atribut align marca la posició del títol dins la taula. Els valors possibles són: dalt (top)valor per defecte-, baix (bottom), esquerra (left) o dreta (right). Aquest atribut està desaprovat . | Obligatòria | Obligatòria |

Així doncs, amb les marques , i es confecciona l'estructura de la taula i el contingut de cada cel·la es situa entre les corresponents marques i

El contingut d'una cel·la pot ser textual i/o gràfic i pot estar organitzat utilitzant els diversos recursos utilitzables en la pàgina: paràgrafs, enllaços, taules, llistes,... És a dir, una cel·la pot contenir taules.

| -igura 10. Exemple de disseny de taula | | | | | | |
|---|--|-------------------|-------------------|--|--|--|
| 🕲 Exemple d'utilització de taules - Mozilla Firefox | | | | | | |
| Eitxer Edita Yisualitza Historial | Adreces d'interès Ei <u>n</u> es Ajuda | | 0 | | | |
| 🧼 • 🔿 • 💽 😣 🟠 | File:///E:/pagina09.html | - | | | | |
| 🗋 Hotmail gratuito 📄 Personalizar v | vínculos 📄 Windows Media 📄 Windo | ows | | | | |
| La meva primera taula | | | | | | |
| Fila 1 - Cel·la 1 Fila 1 - Cel·la 2 Fila 1 - Cel·la 3 | | | | | | |
| Fila 2 - Cel·la 1 Fila 2 - Cel·la 2 | | | | | | |
| Fila 3 - Cel·la 1 | Fila 3 - Cel·la 2 | Fila 3 - Cel·la 3 | Fila 3 - Cel·la 4 | | | |
| | | | | | | |
| Fet | | | .: | | | |

La figura 10 ens mostra la visualització d'una taula. A continuació teniu el corresponent codi.

```
<html>
<head><title>Exemple d'utilitzaci&oacute; de taules</title></head>
<bodv>
<caption><h3>La meva primera taula</h3></caption>
 Fila 1 - Cel· la 1
   Fila 1 - Cel· la 2
   Fila 1 - Cel· la 3
 Fila 2 - Cel· la 1
   Fila 2 - Cel· la 2
 Fila 3 - Cel· la 1
   Fila 3 - Cel· la 2
   Fila 3 - Cel·la 3
   Fila 3 - Cel· la 4
 </bodv>
</html>
```

En el codi anterior podem observar que dins de la taula hem utilitzat un sagnat per a definir-hi les files i dins les files hem utilitzat un altre sagnat per a definir les cel·les. D'aquesta manera és més senzill identificar l'estructura de la taula en el codi font de la pàgina. Aquests sagnats no tenen cap repercussió en la visualització de la pàgina.

Fixem-nos que hem definit una taula amb tres files cadascuna de les quals te un número de cel·les diferent. El navegador crea una taula que te per número de columnes el major número de cel·les de totes les files. I, com en el cas de la figura 10, no és problema pel navegador que el número de cel·les sigui diferent en les diverses files de la taula.

El resultat visual del navegador potser no és el desitjat, doncs igual preteníem que la cel·la 3 de la fila 1 i la cel·la 2 de la fila 2 abracin l'espai de les cel·les 3 i 4 de la fila 3. Per aconseguir això cal utilitzar l'atribut colspan obtenint el codi següent, visualitzat a la figura 11.

```
A la secció "Recursos" del
web d'aquest crèdit podeu
trobar el fitxer pagina10.html
amb el codi de l'exemple.
```

```
<html>
<head><title>Exemple d'utilitzaci&oacute; de taules</title></head>
<bodv>
<caption><h3>Taula utilitzant &quot;colspan&quot;</h3></caption>
 Fila 1 - Cel· la 1Fila 1 - Cel· la 2
   Fila 1 - Cel· la 3
 Fila 2 - Cel· la 1Fila 2 - Cel· la 2
 Fila 3 - Cel· la 1Fila 3 - Cel· la 2
   Fila 3 - Cel·la 3
   Fila 3 - Cel· la 4
 </body>
</html>
```

| Figura 11. Exemple de disseny de lauía utilitzant l'atilibut conspan. | | | | | | |
|---|---------------------------------------|-------------------|-------------------|--|--|--|
| 🕲 Exemple d'utilització de tau | les - Mozilla Firefox | | | | | |
| Eitxer Edita Visualitza Historial | Adreces d'interès Ei <u>n</u> es Ajud | 3 | 0 | | | |
| 👍 • 🔿 • 🥑 🛞 🏠 🗈 file:///E:/pagina10.html | | | | | | |
| 🕒 Hotmail gratuito 🕒 Personalizar vínculos 🗋 Windows Media 🗋 Windows | | | | | | |
| Taula utilitzant ''colspan'' | | | | | | |
| Fila 1 - Cel·la 1 | Fila 1 - Cel·la 2 | Fila 1 - Cel·la 3 | | | | |
| Fila 2 - Cel·la 1 | | | Fila 2 - Cel·la 2 | | | |
| Fila 3 - Cel·la 1 | Fila 3 - Cel·la 2 | Fila 3 - Cel·la 3 | Fila 3 - Cel·la 4 | | | |
| | | | | | | |
| Fet | | | | | | |

Figura 11 Exemple de dissenv de taula utilitzant l'atribut

1.9. Marques per a gestionar llistes

Les llistes són un recurs d'HTML que permet classificar la informació en una estructura jeràrquica, és a dir, podem definir llistes dins altres llistes formant arbres.

La taula 10 presenta les marques a utilitzar en la gestió de llistes.

| Taula 10. Marques per a la gestió de llistes | | | | |
|--|---|-------------------------------|-----------------------------|--|
| Marques | Significat | Marca inicial en HTML 4.01 | Marca final en HTML 4.01 | |
| | Indiquen l'inici i el final d'una llista no ordenada. | | | |
| | Els elements de la llista s'indiquen amb la marca | | | |
| | L'atribut desaprovat type especifica l'estil dels objectes de la llista per als agents d'usuari visuals (navegadors). Els valors possibles són: disc, square i circle. Els navegadors haurien d'intentar presentar un "disc" com un petit cercle ple, un "square" com un petit quadrat buit i un "circle" com un petit cercle buit. | Obligatòria | Obligatòria | |
| | Indiquen l'inici i el final d'una llista ordenada. | | | |
| | Els elements de la llista s'indiquen amb la marca | | | |
| | L'atribut desaprovat type especifica l'estil de numeració dels objectes de la llista per als agents d'usuari visuals (navegadors). Els valors possibles són: | Obligatòria | Obligatòria | |
| | per a utilitzar numeració aràbiga (1, 2, 3,) a, per a utilitzar numeració alfabètica en minúscules (a, b, c,) A, per a utilitzar numeració alfabètica en majúscules (A, B, C,) per a utilitzar números romans en minúscules (i, ii, iii,) per a utilitzar número romans en majúscules (I, II, III,) | | | |
| | L'atribut desaprovat start especifica el valor pel que cal començar a numerar. | | | |
| <1i> 1i | Són les marques per als diferents objectes de les llistes ordenades i no ordenades. | Obligatòria | Opcional | |
| <11> 11 | Pot portar l'atribut desaprovat value que actua com l'atribut start però a partir de l'element sobre el que s'aplica. | | | |
| <dl> </dl> | Indiquen l'inici i el final d'una llista de definicions. | Obligatòria | Obligatòria | |
| <dt> </dt> | Són les marques per als objectes a definir en les llistes de definicions. | Obligatòria | Opcional | |
| <dd> </dd> | Són les marques per a les definicions en les llistes de definicions. | Obligatòria | Opcional | |

El següent codi exemplifica la utilització de les marques per a la gestió de llistes. La figura 12 en mostra la seva visualització.

```
<html>
<head><title>Exemple d'utilitzaci&oacute; de llistes</title></head>
<body>
> Llista 1 - Primer element 
  Llista 1 - Segon element 
  Llista 1 - Tercer element que inclou una llista no ordenada
     Subllista de C - Primer element
       Subllista de C - Segon element
       Subllista de C - Tercer element
     Llista 1 - Quart element
  Llista 1 - Cinquè element que inclou una llista ordenada
     Sublista d'E - Primer elementSublista d'E - Segon element</or>
       Sublista d'E - Tercer element
     Llista 1 - Sisè element
  Llista 1 - Setè element que inclou una llista de definicions
     <d1>
       <dt>+</dt><dl>Signe matem&agrave;tic per a la suma</dl>
       <dt>-</dt><dl>Signe matem&agrave;tic per a la resta</dl>
       <dt>x</dt><dl>Signe matem&agrave;tic per a la multiplicaci&oacute;</dl>
       <dt>/</dt><dl>Signe matem&agrave;tic per a la divisi&oacute;</dl>
     </dl>
  </body></html>
```

A l'igual que en la definició de taules, podem observar que dins de la llista hem utilitzat un sagnat per a definir-hi els seus objectes i, si dins d'aquests hi ha una altra llista, hem utilitzat un nou sagnat. D'aquesta manera és més senzill identificar l'estructura de la llista en el codi font de la pàgina. Aquests sagnats no tenen cap repercussió en la visualització de la pàgina. A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina12.html* amb el codi de l'exemple.

Figura 12. Exemple d'utilització de llistes

| 🕹 Ex | emple | d'utilització de Ilistes - Mozilla Firefox | |
|--------|-----------|---|-------------------|
| Eitxer | Edita | Visualitza Historial Adreces d'interès Eines Ajuda | $\langle \rangle$ |
| | • | - 🕑 🛞 🏠 🗋 file:///E:/pagina11.html - 🕨 💽 - Google | |
| 🗋 На | tmail gra | tuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windows | |
| A | Llista | 1 - Primer element | |
| B. | Llista | 1 - Segon element | |
| C. | Llista | 1 - Tercer element que inclou una llista no ordenada | |
| | 0 | Sublista de C - Primer element | |
| | 0 | Subllista de C - Segon element | |
| | 0 | Sublista de C - Tercer element | |
| D. | Llista | 1 - Quart element | |
| E. | Llista | 1 - Cinquè element que inclou una llista ordenada | |
| | I. | Sublista d'E - Primer element | |
| | Π. | Sublista d'E - Segon element | |
| 20000 | Ш. | Sublista d'E - Tercer element | |
| F | Llista | 1 - Sisè element | |
| G. | Llista | 1 - Setè element que inclou una llista de definicions | |
| | + | | |
| | | Signe matemàtic per a la suma | |
| | - | | |
| | | Signe matemàtic per a la resta | |
| | X | | |
| | | Signe matemàtic per a la multiplicació | |
| | 1 | | |
| | | Signe matemàtic per a la divisi | |
| Fet | | | |

1.10. Una eina d'edició de pàgines web: Nvu

Les pàgines web que visualitzen els navegadors estan desenvolupades amb el llenguatge de marques HTML i, per tant, els dissenyadors de pàgines web l'han de conèixer i emprar. La seva utilització, però, pot esdevenir feixuga i obliga al dissenyador a memoritzar una gran quantitat d'ordres i de marques diferents.

Per a facilitar la feina dels dissenyadors de pàgines web ha anat apareixent editors especialitzats que generen el codi HTML a partir d'una tecnologia *WYSIWYG*.

La tecnologia *WYSIWYG*, acrònim de la frase anglesa "*What you see is what you get*" (el que veus és el que obtens), en l'àmbit del disseny permet que el que es veu durant l'edició es correspongui més o menys acuradament amb el resultat final (imprès o en pantalla).

La majoria dels processadors de textos utilitzen, actualment, aquesta tecnologia. Els editors HTML que la incorporen, faciliten que el dissenyador desenvolupi la pàgina web de manera similar a l'escriptura en un processador de textos: va dissenyant en pantalla allò que es visualitzarà en el resultat final. L'editor s'ocupa de generar el corresponent codi HTML.

Així doncs, la tecnologia *WYSIWYG* permet dissenyar pàgines web sense utilitzar directament codi HTML. No obstant això, cal conèixer el codi HTML doncs en ocasions haurem de poder efectuar petits retocs en el codi HTML generat per l'editor.

En el mercat hi ha diferents editors HTML amb la tecnologia WYSIWYG: Dreamweaver, Nvu, Microsoft Frontpage, IucundeWeb, Quanta Plus, Amaya, Writer d'OpenOffice,... Anem a treballar amb una d'elles: Nvu.

Nvu (pronunciat *N-view* provinent de *new view*) és un editor HTML gratuït que facilita el disseny de pàgines web gràcies a les diferents visualitzacions que possibilita (codi font, finestra *WYSIWYG*, visió amb marques HTML realçades).

Nvu inclou també altres característiques com la gestió de treballs via projectes, client FTP integrat per a pujar la pàgina web directament des de *Nvu* fins el corresponent servidor i suport per a tots els elements típics de les pàgines web: marcs, formularis, taules, plantilles de disseny, fulles d'estil CSS, etcètera.

Nvu està disponible per als sistemes operatius *Linux*, *Mac* OS X i *Microsoft Windows* i es pot obtenir una versió per a qualsevol plataforma a partir del seu codi font. Fins i tot existeix una versió portàtil que pot ser transportada i utilitzada directament des d'una memòria USB sense necessitat d'instal·lar-se a l'ordinador.

1.10.1. Interfície

El primer pas a efectuar en treballar amb una eina informàtica és familiaritzar-se amb la seva interfície. No ens ha de ser difícil la utilització de Nvu si estem acostumats a utilitzar algun processador de textos.

Com en totes les eines d'edició, *Nvu* facilita una zona d'edició (distingible a la figura 13) on s'escriu el text, es col·loquen les imatges,... És com una pàgina virtual de paper.

Al voltant de la zona d'edició, un seguit d'elements proporcionen accés al conjunt de comandaments que permeten dissenyar la pàgina web. Tots aquests comandaments es poden ocultar o mostrar per mitjà del menú *Visualitza*|*Mostra*/*Amaga*. Vegem-ne la seva funcionalitat tot identificantlos a la figura 13.

| 😨 sense títol - Nvu | - 6 🛛 |
|--|-------|
| Etxer Edita Visualitza Insereix Format Taula Eines Ajuda Barra de menús | |
| Nou Obra Desa Publica Nurreja Àncera Enllar, Imaige Taula Formulari Vierfica Imprimeix Barra d'eines de composició | N |
| Text del cos ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ B / U 注目目 臣 至 通 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ | |
| (cap classe) 🔽 Amplada variable 👱 🔍 🕲 🗓 1 !!! 三 三 🎼 (岡) 🦏 🍸 蓋 | |
| × (sense titol) Barra d'exploració amb fitxes | * |
| Gestor de llocs web | |
| Vista: Tots els fitxers V | |
| Edua lices 📦 🖆 🗭 😵 😧 📔 🦳 Regles | |
| Nom P3 | |
| | |
| | |
| | |
| | |
| Gestor de llocs web Zona d'edició | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| Normal TE Marcadors HTML O Codi font HTML Revisualització Barra de mode d'edició | |
| | |

Figura 13. Zones de la interfície de l'editor de pàgines web Nvu

1) Regles

Es troben situades al voltant de la zona d'edició i mostren l'amplada i l'alçada en pixels de la pàgina o d'un objecte seleccionat.
2) Gestor de llocs web

Situat a l'esquerra de la zona d'edició, permet administrar el lloc web on residiran les nostres pàgines. La creueta que té a l'extrem superior esquerre permet tancar-lo. La tecla de funció F9 facilita la funcionalitat d'obertura i tancament.

La grandària d'aquesta zona és governable col·locant el ratolí damunt la barra de separació vertical amb la zona d'edició, fent-hi clic i sense deixar-lo, arrossegar la barra cap a l'esquerra o cap a la dreta.

3) Barra de menús

Situada dalt de la finestra, permet accedir a tots els comandaments de Nvu.

4) Barra d'eines de composició

Situada sota la barra de menús, conté una botonera per accedir de forma ràpida als comandaments més emprats. La seva composició es pot personalitzar pel menú *Visualitza*|*Mostra*/*Amaga*|*Personalitza* la barra d'eines |Barra d'eines principal. Cada botó té associat un tooltip que es visualitza en passar el ratolí pel damunt.

5) Barra d'eines de format

Situada sota la barra d'eines de composició, està estructurada en dues zones (té l'aparença de dues barres) i conté les eines per donar format al contingut de la zona d'edició. La seva composició es pot personalitzar pel menú *Visualitza* |*Mostra*/*Amaga* |*Personalitza la barra d'eines* |*Barra d'eines de formal*. Cada botó té associat un indicador de funció (en anglès, *tooltip*) que es visualitza en passar el ratolí pel damunt.

6) Barra d'exploració amb fitxes

Situada sota la barra d'eines de format i damunt de la regla horitzontal. *Nvu* permet treballar simultàniament amb varis documents els quals es visualitzen en forma de fitxes. Aquesta barra conté una pestanya per a cada document obert i permet canviar d'una pàgina a una altra.

7) Barra de mode d'edició

Situada immediatament sota de la zona d'edició permet indicar el tipus de visualització del document actiu a la zona d'edició. Hi ha quatre possibilitats: normal, marcadors HTML, codi font HTML i previsualització. Per a observar l'efecte dels diferents tipus de visualització, obrim amb *Nvu* el document *pagina07.html*. Podem fer-ho des de *Nvu* amb el menú *Fitxer* | *Obre un fitxer* i seleccionant l'arxiu en la finestra exploradora de fitxers que apareix o des del propi sistema d'arxius del sistema operatiu, indicant que volem obrir l'arxiu amb l'eina *Nvu*. Una vegada obert, vegem l'efecte dels diferents tipus de visualització sobre el document.

Figura 14. Tipus de visualització "normal" d'un document HTML en Nvu

| Exe | emples d'enllaços - Arxiu <pagina th="" 🙀<=""></pagina> | | | |
|-------|--|--|--|--|
| | 605px | | | |
| 202px | Aquesta pàgina està enregistrada amb el nom pagina07.html En la mateixa ubicació hi ha una altra pàgina de nom pagina06.html A l'inici d'aquest paràgraf hi ha una àncora de nom "anc01". A continuació facilitem un enllac que ens porti a l'ancoratge anc01 de la mateixa pàgina. | | | |
| | A continuació facilitem un enllaç que ens porti a la <u>pàgina 06</u> | | | |
| | 🗅 Normal 🔟 Marcadors HTML 🖸 Codi font HTML 🔍 Previsualització | | | |

a) Normal

Mostra la pàgina tal i com es veurà en un navegador juntament amb les marques d'àncora i els contorns de les taules que hi pugui haver en el document. La figura 14 mostra la visualització "normal" del document *pagina07.html*. Hi podem observar una àncora a l'inici del segon paràgraf.

b) Marcadors HTML

Mostra totes les marques HTML existents en el document. La figura 15 mostra la visualització "marcadors HTML" del document *pagina07.html*. Podem observar-hi les marques <body>, , <a> i l'àncora.

Figura 15. Tipus de visualització "marcadors HTML" d'un document HTML en Nvu



c) Codi font HTML

Mostra el codi HTML, com es pot veure en la figura 16, amb dues característiques importants:

- El text del codi apareix amb diferents colors. Això és una ajuda visual que permet distingir els diferents continguts entre sí.
- La numeració automàtica de les línies en el marge esquerra, fet que pot servir en documents *.html* grans per prendre anotacions sobre la ubicació exacta de certs continguts.

| Figura 1 | 6. Tipus de visualització "codi font HTML" d'un document HTML en Nvu |
|----------|--|
| 1. | |
| 2. | <html></html> |
| з. | <head></head> |
| 4. | <title>Exemples d'enllaços - Arxiu</title> |
| 5. | <pre>\$lt;pagina07.html></pre> |
| 6. | |
| 7. | <body></body> |
| 8. | Aquesta pàgina està enregistrada amb el |
| 9. | nom pagina07.html |
| 10. | En la mateixa ubicaci¢oacute; hi ha una altra pàgina de |
| 11. | nom pagina06.html |
| 12. | A l'inici d'aquest |
| 13. | par s agrave;graf hi ha una s agrave;ncora de nom "anc $01".$ |
| 14. | A continuació facilitem un enllaç que |
| 15. | ens porti a l'ancoratge anc01 |
| 16. | de la mateixa pàgina. |
| 17. | A continuaciíoacute; facilitem un enllaíccedil; que |
| 18. | ens porti a la pàgina |
| 19. | 06 |
| 20. | |
| 21. | |
| | |
| 📄 Normal | TT Marcadors HTML 🖸 Codi font HTML 🔍 Previsualització |

d) Previsualització

Mostra la pàgina web en el mode *WYSIWYG*, és a dir, tal i com es visualitzaria en qualsevol navegador. La figura 17 n'és un exemple.

Figura 17. Tipus de visualització "previsualització" d'un document HTML en Nvu

| Exe | emples d'enllaços - Arxiu <pagina< th=""><th><u> </u></th></pagina<> | <u> </u> | | |
|---|---|----------|--|--|
| | 635px | | | |
| | Aquesta pàgina està enregistrada amb el nom pagina07.html En la mateixa ubicació hi ha una altra pàgina de nom pagina06.html | | | |
| 48px | A l'inici d'aquest paràgraf hi ha una àncora de nom "anc01". | | | |
| | A continuació facilitem un enllaç que ens porti a l'ancoratge <u>anc01</u> de la mateixa pàgina. | | | |
| | A continuació facilitem un enllaç que ens porti a la <u>pàgina 06</u> | | | |
| | | | | |
| 🕒 Normal 🔟 Marcadors HTML 🖸 Codi font HTML 🔍 Previsualització | | | | |

8) Barra d'estat

Aquesta barra està situada en la part inferior i mostra les marques HTML obertes en la posició on es troba el cursor dins la zona d'edició.

Així, en l'exemple precedent, si situem el cursor enmig de la paraula *anc01*, veurem que la barra d'estat mostra <body> <a> doncs hi ha obert un enllaç (<a>) que es troba dins un paràgraf () el qual es troba dins el cos de la pàgina (<body>).

1.10.2. Disseny de pàgines web

Nvu és un editor que permet modificar pàgines web existents (menú *Fitxer*|*Obrir un fitxer* o botó *Obre* de la barra d'eines de composició) i crear-ne de noves (menú *Fitxer*|*Nou* o botó *Nou* de la barra d'eines de composició). Així mateix permet definir les propietats que incideixen en el comportament global de la pàgina.

Creació d'una pàgina web

El botó Nou va acompanyat a la seva dreta d'una petita fletxa que permet visualitzar les opcions d'obertura:

- Pàgina en una pestanya nova, que obre una nova fitxa en la zona d'edició
- Pàgina en una finestra nova, que provoca una nova execució de l'editor Nvu mantenint oberta la instància en execució actual
- Més opcions..., que provoca l'aparició de la finestra modal de la figura 18.

El menú *Fitxer*|*Nou* provoca directament l'obertura de la finestra modal de la figura 18.



La finestra modal que facilita *Nvu* permet crear un document en blanc i es pot indicar si el document seguirà la sintaxi XHTML (per defecte no) i/o si serà un document "DTD estricte" (per defecte no). També permet crear documents basats en plantilles i crear aquestes plantilles, però la gestió de plantilles s'escapa de l'àmbit d'aquest apartat.

No podem deixar de comentar, però, el tema de si el document seguirà la sintaxi XHTML o si serà un document "DTD estricte". Per millor entendre què implica cada opció, procedim a crear diversos documents combinant les diferents opcions i, per a cada document creat, activem la

visualització "codi font HTML" i observem les dues primeres línies de codi:

• Opció XHTML desmarcada i opció DTD estricte desmarcada

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
```

• Opció XHTML marcada i opció DTD estricte desmarcada

• Opció XHTML desmarcada i opció DTD estricte marcada

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
```

• Opció XHTML marcada i opció DTD estricte marcada

En els quatre casos, tenim una primera línia amb la marca <!DOCTYPE> que informa sobre la normativa que segueix el contingut del document. És a dir, hi ha diversos tipus de documents HTML (els que verifiquen l'especificació *HTML 4.01*, els que verifiquen l'especificació *HTML 4.0*, els que verifiquen l'especificació *XHTML Basic*, etcètera) i la marca <!DOCTYPE> permet definir l'especificació que es suposa seguirà el contingut del document.

Nvu sap generar la marca <!DOCTYPE> per a quatre possibilitats segons haguem marcat les opcions *XHTML* i/o *DTD estricte* però n'hi ha d'altres i, si ens cal, podem modificar els valors que acompanyen a la marca <!DOCTYPE> modificant el codi font HTML del document.

És molt convenient incloure en els documents HTML la marca <!DOCTYPE> amb la informació que correspongui per a informar als agents d'usuari que hagin d'interpretar el document. Davant documents que no incorporen la definició del tipus de document, els agents d'usuari prenen decisions per defecte, les quals podrien estar equivocades.

DTD - Definició de Tipus de Document

Una definició de tipus de document és una descripció sobre l'estructura d'un document i la sintaxis que seguiran els seus elements.

En les recomanacions del W3C per a la darrera versió d'HTML (4.01 – any 1999) s'incloïa l'especificació per a la definició de tres tipus de documents: el "DTD Estricte", el "DTD Transicional" i el "DTD amb marcs".

El DTD Estricte de HTML 4.01, exclou els elments i atributs de presentació que el W3C espera que deixin d'utilitzar-se a mida que maduri el suport a les fulles d'estil i comenta que els autors hauríen d'utilitzar el DTD estricte sempre que sigui possible, però deixa oberta la possibilitat d'utilitzar el DTD Transicional si és necessari el suport d'elements i atributs de presentació.

El DTD amb marcs de HTML 4.01 és el que caldria utilitzar per a documents que utilitzen marcs (frames). Aquest DTD és idèntic al DTD Transicional excepte, però, canviant la marca <body> per la marca <frameset>.

La segona línia sempre és l'etiqueta <html>, solitària si es tracta d'un document HTML o acompanyada de l'opció xmlns si es tracta d'XHTML.

Nosaltres utilitzarem *Nvu* per a dissenyar pàgines amb codi HTML i amb *DTD Transicional* en el que es permet utilitzar elements desaprovats. Per tant no marcarem cap de les dues opcions *XHTML* i *DTD Estricte*.

Una vegada creada, ens apareix en una nova pestanya o una nova finestra (segons haguem indicat), la nova pàgina amb la zona d'edició buida. El document HTML corresponent a la pàgina encara no té nom ni està enregistrat en el sistema d'arxius de l'ordinador.

Per guardar la pàgina en el sistema d'arxius del nostre ordinador, podem utilitzar el botó *Desa* de la barra d'eines de composició o el menú *Fitxer*|*Desa*.

Podem comprovar la seva visualització en el navegador preestablert del sistema per medi del botó *Navega* de la barra d'eines de composició o pel menú *Fitxer*|*Navega pàgina*. Si volem utilitzar un altre navegador l'haurem de cridar des de fora de *Nvu*. En qualsevol cas, el que hem de veure és una pàgina buida amb el títol a la barra blava superior.

Títol i propietats de la pàgina

Nvu permet definir algunes propietats bàsiques de la pàgina pel menú *Format*|*Títol de pàgina i propietats...* que provoca l'aparició de la finestra modal de la figura 19. En ella hi trobem diferents apartats:

- Títol, per omplir-lo amb el títol que apareixerà a la barra blava superior del navegador.
- Autor, per omplir-lo amb el nom de l'autor el qual no apareix per enlloc en visualitzar la pàgina.
- Descripció, per omplir-la amb una breu explicació del contingut de la pàgina, la qual tampoc apareix per enlloc en visualitzar la pàgina.
- Idioma, per omplir-lo amb l'idioma (seguint una codificació preestablerta) en que estarà redactada la pàgina. Aquesta informació

la utilitzen els cercadors per a poder mostrar les pàgines en un determinat idioma.

• Codificació de caràcters, per omplir-la amb la que correspongui, que acostuma a ser la que *Nvu* proposa (segons la configuració idiomàtica del sistema)

La figura 19 mostra la finestra de les propietats d'una hipotètica pàgina inicial substitutòria del material paper d'aquesta unitat didàctica.

Figura 19. Pantalla de Nvu per a introduir les propietats de la pàgina en disseny.

| Informació general ——— | | | | |
|---|--|--------------------------------------|--|--|
| Ubicació: | [Nova pàgina, no desada enci | ara] | | |
| Última modificació: | Desconegut | | | |
| <u>T</u> ítol: | Elaboració de pàgines web am | nb llenguatges de marques | | |
| <u>A</u> utor: | Isidre Guixà i Miranda | Isidre Guixà i Miranda | | |
| Descripció: | Pàgina inicial de la tercera uni | tat didàctica del crèdit "Mantenime | | |
| Plantilles | | | | |
| 🔲 Aquesta pàgina és un | a plantilla | | | |
| Internacionalització —— | | | | |
| Idioma: | ca 📃 | Escull un idioma | | |
| Direcció d'escriptura: | Cap direcció especifi 🔽 | | | |
| Codificació de caràcters: | ISO-8859-1 E | scull una codificació de caràcters | | |
| - Per a usuaris avancats | | | | |
| Si desitaeu editar d'altres i | roptinguts en el <bead>, podeu ul</bead> | tilitzar l'onció "Codi font HTMI" en | | |
| el Menú de Visualitza o en la barra d'eines d'edició. | | | | |
| | | | | |

Color de la pàgina i del fons

Nvu permet definir els colors de fonts i dels text de la pàgina pel menú *Format*|*Colors de la pàgina i del fons...* que provoca l'aparició de la finestra modal de la figura 20. En ella hi trobem diferents possibilitats:

Figura 20. Pantalla de Nvu per a definir els colors de la pàgina en disseny.

| olors i fons de la pàgina | | | |
|---|---|--|--|
| Colors de la pàgina | definativia da calava da la adminati | | |
| Colors per derecte del lector (no derineixis els colors de la pagir Utilitza els colors personalitzats: | | | |
| Text gormal: | Text normal Text de l'enllaç Text de l'enllaç actiu Text de l'enllaç visitat | | |
| Imatge de fons: | la pàgina Escull un fitxer Edició avançada D'acord Cancel·la | | |

• No definir cap color específic i deixar que actuïn els colors per defecte del navegador

- Definir colors personalitzats per al text normal, el text dels enllaços, el text de l'enllaç actiu, el text dels enllaços ja visitats i pel fons.
- Definir una imatge de fons

Afegir i suprimir text

Una vegada creada la pàgina podem, en la visualització "normal", començar a escriure el text que volem incorporar-hi, com si es tractés d'un processador de text.

Figura 21. Text escrit a la zona d'edició en visualització "normal".

| Elaboracio de pagines web amb lien | | · · · · · · · · · · · · · · · · · · · | | | |
|------------------------------------|--|--|--|--|--|
| | 635px | | | | |
| Introducció | | | | | |
| 260px | Internet, la xarxa de comunicacions que posa en contacte milions de sistemes informàtics repartits per tot el planeta, facilita diversos serveis als seus usuaris: correu electrònic (e-mail), xerrades (xat), pujar i baixar fitxers (FTP), taulers d'anuncis (news), vídeo conferència i, el seu servei més utilitzat, la teranyina de documents (pàgines web), formada per un immens conjunt de documents que poden tenir contingut textual, gràfic, àudio, animacions, etcètera. | | | | |
| | En les unitats didàctiques pr web així com els procedime aquesta unitat didàctica apri siguin atractives i entenedor usuaris de la xarxa. | ecedents heu anat treballant els conceptes vinculats amb l'explotació d'un lloc nts a emprar per actualitzar-ne els continguts i la normativa legal vigent. En endreu com mantenir i dissenyar, d'una forma correcta, pàgines web que es, de forma que hi hagi un alt grau de probabilitat de ser visitades per molts | | | |
| | Normal TO Marcadors HTML | Codi font HTML | | | |

La figura 21 mostra l'edició dels dos primers paràgrafs de la introducció.

Hem anat copiant el text introductori de la present unitat didàctica i no ens hem preocupat d'introduir cap marca HTML, ni per als caràcters especials. A continuació s'exposa el codi HTML que ha generat *Nvu*:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ca">
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Elaboraci&oacute; de p&agrave;gines web amb llenguatges de marques</title>
  <meta content="Isidre Guix&agrave; i Miranda" name="author">
 <meta content="P&agrave;gina inicial de la tercera unitat did&agrave;ctica del
 crèdit "Manteniment de portals d'informació"
 name="description">
</head>
<body>
Introducció<br>
<br>
Internet, la xarxa de comunicacions que posa en contacte milions de
sistemes informà tics repartits per tot el planeta, facilita
diversos serveis als seus usuaris: correu electrònic
(e-mail), xerrades (xat), pujar i baixar fitxers (FTP), taulers
d'anuncis (news), vídeo conferència i,
el seu servei més utilitzat, la teranyina de documents
(pà gines web), formada per un immens conjunt de documents
que poden tenir contingut textual, gràfic, àudio,
animacions, etcètera.<br>
<br>
En les unitats didà ctiques precedents heu anat treballant
els conceptes vinculats amb l'explotació
d'un lloc web així com els procediments a emprar
per actualitzar-ne els continguts i la normativa legal vigent. En
aquesta unitat didà ctica aprendreu com mantenir i dissenyar,
d'una forma  correcta, pàgines web que siguin
atractives i entenedores, de forma que hi hagi un alt grau de
```

```
probabilitat de ser visitades per molts usuaris de la xarxa.<br>
</body>
</html>
```

Observacions:

- L'etiqueta <html> inicial conté la informació sobre l'idioma (opció lang).
- Dins la capçalera (zona entre les marques <head> i </head>) hi ha informació sobre l'autor, la descripció i el joc de caràcters emprat, utilitzant l'element <meta> per a informacions genèriques. L'element <meta> té prohibida la marca final (és a dir, no existeix </meta>).
- Els caràcters especials han quedat correctament codificats per a que siguin correctament tractats sota qualsevol codificació idiomàtica.
- Cada salt de línia introduït en la zona d'edició mitjançant la tecla *Enter* ha generat la corresponent marca
>.

1.10.3. Format bàsic del text

Nvu proporciona diferents mecanismes per donar format al text de la pàgina de manera que aquesta sigui còmodament llegible pels visitants.

Paràgrafs

En el text introduït a la pàgina mostrada en la figura 21, no hi hem introduït cap paràgraf. L'efecte de canvi de paràgraf l'hem aconseguit inserint salts de línia que han provocat l'aparició de marques
> en el codi HTML. Observem, a la figura 21, que el cursor està situat enmig de la quarta línia i que la regla vertical mostra un únic bloc que avarca tot el text amb una alçada de 260 pixels.

Anem a fer que cada paràgraf visual sigui veritablement un paràgraf HTML. Per aconseguir-ho ens hem de situar en qualsevol línia del primer paràgraf (deixem de moment de banda el títol "Introducció") en visualització "normal" i despleguem la llista *Text del cos* de la barra d'eines de format i seleccionem l'opció *Paràgraf*, com a la figura 22. També hi podríem arribar pel menú *Format* |*Paràgraf*.

Observem, a la figura 23, que el bloc de text on hi ha situat el cursor s'ha separat més del seus predecessor i successor. Ben segur que ara trobem excessiva la separació que hi ha, però és deguda a l'espai de separació que proporciona el paràgraf més els salts de línia que ja hi havia. A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina12.html* amb el codi de l'exemple. Observem també que la regla vertical ara no avarca tot el text sinó només el paràgraf en el que estem situats.

Figura 22. Opcions per a donar format a un paràgraf



Figura 23. Efecte de la definició d'un paràgraf en el bloc on hi ha situat el cursor

| ∎ 8 | Elaboració de pàgines web amb lle 🙀 | | |
|---|--|--|--|
| | 635px | | |
| | Introducció | | |
| Internet, la xarxa de comunicacions que posa en contacte milions de sistemes informàtics repartits per el planeta, facilita diversos serveis als seus usuaris: correu electrònic (e-mail), xerrades (xat), pujar i b fitxers (FTP), taulers d'anuncis (news), vídeo conferència i, el seu servei més utilitzat, la teranyina de documents (pàgines web), formada per un immens conjunt de documents que poden tenir contingut textual, gràfic, àudio, animacions, etcètera. | | | |
| | En les unitats didàctiques precedents heu anat treballant els conceptes vinculats amb l'explotació d'un lloc web així com els procediments a emprar per actualitzar-ne els continguts i la normativa legal vigent. En aquesta unitat didàctica aprendreu com mantenir i dissenyar, d'una forma correcta, pàgines web que siguin atractives i entenedores, de forma que hi hagi un alt grau de probabilitat de ser visitades per molts usuaris de la xarxa. | | |
| | Normal 🔟 Marcadors HTML 🖸 Codi font HTML 🔍 Previsualització | | |

Definim també un paràgraf HTML en el darrer paràgraf visual i eliminem els salts de línia que havíem inserit inicialment. Per eliminar els salts de línia podem col·locar-nos-hi amb el cursor i procedir a la seva eliminació amb la tecla *Supr* del teclat de la mateixa manera que faríem en un processador de textos. La figura 24 en mostra els resultats en visualització "marcadors HTML".

Figura 24. Visualització "marcadors HTML" de pàgina amb paràgrafs definits.

| | Elaboració de pàgines web amb lle | | | | |
|-------|--|--|--|--|--|
| | 635px | | | | |
| 100px | Introducció Introducció Internet, la xarxa de comunicacions que posa en contacte milions de sistemes informàtics repartits per tot el planeta, facilita diversos serveis als seus usuaris: correu electrònic (e-mail), xerrades (xat), pujar i baixar fitxers (FTP), taulers d'anuncis (news), vídeo conferència i, el seu servei més utilitzat, la teranyina de documents (pàgines web), formada per un immens conjunt de documents que poden tenir contingut textual, gràfic, àudio, animacions, etcètera. En les unitats didàctiques precedents heu anat treballant els conceptes vinculats amb l'explotació d'un lloc web així com els procediments a emprar per actualitzar-ne els continguts i la normativa legal vigent. En aquesta unitat didàctica aprendreu com mantenir i dissenyar, d'una forma correcta, pàgines web que siguin atractives i entenedores, de forma que hi hagi un alt grau de probabilitat de ser visitades per molts usuaris de la xarxa. | | | | |
| | Normal 🔟 Marcadors HTML 🖸 Codi font HTML 🔍 Previsualització | | | | |

Alineació i sagnat

Nvu facilita quatre botons d'alineació a la barra d'eines de format que permeten alinear el text als marges que interessi i dos botons per a posar/treure sagnat al text. Per aconseguir-ho simplement hem de tenir el cursor situat en la zona de text a tractar i prémer el corresponent botó.

Així, si en la pàgina que estem dissenyant volguéssim tenir la primera línia centrada i els altres dos paràgrafs justificats, caldria situar el cursor en la primera línia i prémer el botó *Alinea al centre* i situar el cursor en cada paràgraf i prémer el botó *Alinea de forma justificada*. La figura 25 en mostra els resultats.

Figura 25. Visualització "marcadors HTML" de pàgina amb paràgrafs definits i alineats

| Elaboració de pàgines web amb lle | ¥ . | | | | | | |
|--|--|--|--|--|--|--|--|
| | 607px | | | | | | |
| 4 px BODY DIV | Introduccid | | | | | | |
| Internet, la xarxa de c per tot el planeta, facili pujar i baixar fitxers (F teranyina de document tenir contingut textual, g | Internet, la xarxa de comunicacions que posa en contacte milions de sistemes informàtics repartits per tot el planeta, facilita diversos serveis als seus usuaris: correu electrònic (e-mail), xerrades (xat), pujar i baixar fitxers (FTP), taulers d'anuncis (news), vídeo conferència i, el seu servei més utilitzat, la teranyina de documents (pàgines web), formada per un immens conjunt de documents que poden tenir contingut textual, gràfic, àudio, animacions, etcètera. | | | | | | |
| En les unitats didàctiques precedents heu anat treballant els conceptes vinculats amb l'explotació d'un lloc web així com els procediments a emprar per actualitzar-ne els continguts i la normativa legal vigent. En aquesta unitat didàctica aprendreu com mantenir i dissenyar, d'una forma correcta, pàgines web que siguin atractives i entenedores, de forma que hi hagi un alt grau de probabilitat de ser visitades per molts usuaris de la xarxa. | | | | | | | |
| Normal Marcadors HTML | Codi font HTML | | | | | | |

A la figura 25 s'observa que per haver indicat un tipus d'alineació a la primera línia, la qual no formava part de cap paràgraf, *Nvu* ha definit un bloc de continguts amb la marca <div> per a poder donar valor a l'opció d'alineació. Això no ha estat necessari en la resta de paràgrafs.

Activem la visualització "codi font HTML" i fixem-nos en les següents zones del codi generat per *Nvu*:

```
<body>
<div style="text-align: center;">Introducci&oacute;</div>
Internet, ...
En les unitats ...
</body>
```

Observem que per definir l'alineació no ha utilitzat l'atribut align dins les marques $\langle p \rangle$ i $\langle div \rangle$ sinó que ha emprat l'atribut style. Això és degut a que *Nvu* segueix les indicacions de *W3C* referent a la utilització dels estils. En els exemples que segueixin és interessant anar visualitzant sempre el codi HTML que genera *Nvu* per a comparar-lo amb els nostres coneixements d'HTML.

Capçaleres

En *Nvu*, per poder assignar a una línia un dels sis tipus de capçalera que permet HTML, cal utilitzar la llista desplegable *Text del cos* de la barra d'eines de format (figura 22) i seleccionar l'opció *Encapçalament* que correspongui. La figura 26 mostra l'efecte d'assignar l'*Encapçalament 1* a la primera línia de la pàgina que estem dissenyant.

Figura 26. Visualització "marcadors HTML" de pàgina amb paràgrafs alineats i capçalera



Aparença del text

Nvu proporciona un conjunt d'eines a la barra d'eines de format per a definir l'aparença del text, que cal aplicar havent seleccionat prèviament el text a modificar. *Nvu* genera el codi HTML implementant, sempre que pot, l'atribut style amb els valors que corresponguin.

Les eines que proporciona la barra d'eines de format són:

- Botó bifuncional Escull color pel text i Escull color pel fons
- Botó Escull un color pel text ressaltat
- Botó Mida de la lletra més petita
- Botó Mida de la lletra més gran
- Botó Negreta
- Botó Cursiva
- Botó Subratllat
- Llista desplegable Escull font
- Botó Èmfasi
- Botó Èmfasi més fort

Respecte els fonts comentar que la llista de fonts disponibles dependrà de les que estiguin instal·lades a l'ordinador i, per tant, totes són utilitzables, però és aconsellable utilitzar només *Helvètica, Arial, Times* o *Courier* que són les que la llista desplegable *Escull font* proposa de forma separada a les restants, com es veu a la figura 27.

El motiu de la recomanació radica en que no tots els ordinadors tenen instal·lats els mateixos tipus de fonts i, per tant, un navegador que hagi de visualitzar una pàgina que contingui un font no instal·lat en el seu ordinador, no podrà utilitzar aquell tipus de font i el substituirà pel primer que tingui disponible de la "família genèrica" del font emprat.

Figura 27. Llista desplegable per escollir font

| | Amplada variable | ~ | 0 |
|---|----------------------------|----|---|
| I | Amplada variable | | ~ |
| l | Amplada fixa | | |
| b | Helvetica, Arial Escull fo | nt | |
| | Times | | = |
| | Courier | | |
| - | Agency FB | | |
| ł | Algerian | | |
| | Arial | | |
| | Arial Black | | |
| | Arial Narrow | | |

Per entendre millor el concepte de "família genèrica" del font emprat, assignaré al text de la primera línia el tipus de lletra *Agency FB*. El codi HTML generat per *Nvu* és:

<div style="text-align: center; font-family: Agency FB;">

Observem l'apartat font-family (que es pot traduir per "família de font") amb el valor *Agency FB* corresponent al font emprat. En aquest cas no hi ha cap altra informació de font i ves a saber quina font utilitzarà un navegador que no tingui instal·lada la font *Agency FB*. Per a minimitzar el possible impacte negatiu de la no existència de la font, podem acompanyar el valor *Agency FB* d'altres tipus de font, cosa que hem de fer en la visualització "codi font HTML". Així, podríem posar:

<div style="text-align: center; font-family: Agency FB,Helvetica,Arial,sans-serif;">

D'aquesta manera els navegadors cerquen en el seu ordinador la font a utilitzar seguint l'ordre indicat en font-family: en primer lloc l'Agency FB i si no està instal·lada cercarà l'Helvetica i si tampoc hi és, cercarà l'Arial i si tampoc hi és provarà amb qualsevol de la família de fonts genèrica sans-serif.

Hi ha diverses famílies de fonts genèriques:

- sans-serif (pal sec en llengua catalana) indica la família de fonts sense ornamentacions en els seus extrems. Les lletres Helvetica i Arial són de pal sec.
- serif (*serifa* en llengua catalana) indica la família de fonts amb ornamentacions en els seus extrems. La lletra *Times* és serifa.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina13.html* amb el codi de l'exemple. • monospace indica la família de fonts en les que tots els símbols tenen la mateixa amplada. La lletra *Courier* és d'amplada fixa.

Així, si tot i les recomanacions d'utilitzar només les fonts *Helvetica*, *Arial*, *Times* o *Courier* ens aventurem a indicar un altre tipus de font, seria convenient completar en el codi font HTML l'apartat font-family amb la família de fonts genèrica corresponent al tipus de lletra emprat.

1.10.4. Gestió d'enllaços

Nvu dona tot tipus de facilitats per a definir enllaços cap a altres documents i altres serveis.

Recordem que en referència a enllaços cap a altres documents, els documents a enllaçar poden residir:

- En el mateix lloc web i per tant és aconsellable indicar la ruta relativa a la pàgina actual
- En un lloc web extern al lloc web on es troba la pàgina i en aquest cas cal indicar la ruta absoluta

Recordem també que si el document a accedir té àncores internes podem accedir-hi directament.

Per a definir un nou enllaç des de *Nvu*, podem utilitzar el botó *Enllaç* de la barra d'eines de composició o el menú *Insereix*|*Hiperenllaç* que provoca l'aparició de la finestra modal de la figura 28. En ella hi trobem diferents possibilitats:

| Figura 28. Pantalla de Nvu per a definir enllaços | | | | |
|--|--|--|--|--|
| Propietats de l'enllaç 🛛 🔀 | | | | |
| Text d'enilaç Entreu el text per a mostrar en l'enilaç: | | | | |
| Ubicació de l'enllaç Entreu la localització d'una pàgina web, un fitxer local, una adreça de correu o seleccioneu una àncora o la capçalera des de la llista emergent: | | | | |
| L'anterior és una adreça de correu | | | | |
| Més gropietats Edició avançada D'acord Cancel·la Ajuda | | | | |

• Text d'enllaç, que només es pot introduir si en demanar l'aparició de la finestra no havíem seleccionat cap text. En tal cas, el text que introduïm en aquest apartat, apareixerà a la zona d'edició en el lloc on estes situat el cursor.

- Ubicació de l'enllaç, que permet indicar la ruta que correspongui. Les possibilitats d'aquest espai són variades:
 - Ruta absoluta si es tracta d'un document extern al nostre lloc web:

http://www.xtec.cat

Si del document destí volem accedir a un lloc marcat amb una àncora podem indicar-ho afegint el nom de l'àncora:

http://www.xtec.cat#nom_marca

 Ruta relativa a la pàgina actual si es tracta d'un document que resideix en el mateix lloc web que la nostra pàgina. També cal afegir el nom de l'àncora si es vol anar a algun punt concret.

En aquest cas cal activar la casella *La URL és relativa a la localització de la pàgina* que hi ha una mica més avall.

 Directament el nom de la marca si es tracta d'un enllaç a una àncora de la pròpia pàgina:

#nom_marca

- Direcció de correu electrònic si l'enllaç ha de servir per enviar un correu electrònic a alguna adreça concreta. En tal cas, cal activar la casella *L'anterior és una adreça de correu* que hi ha a continuació.
- Més propietats, que permet ampliar la finestra amb nous camps, dels quals ens interessa fer esment a l'apartat objectiu |l'enllaç és per a ser obert que permet decidir en quin document s'obrirà l'enllaç i que correspon als valors d'HTML: _blank, _self, _parent i _top.

Per finalitzar la gestió d'enllaços, comentar que *Nvu* també permet definir àncores allà on sigui menester, utilitzant el botó *Àncora* de la barra d'eines de composició o el menú *Insereix* |*Àncora*. En qualsevol cas, apareix una finestra modal on cal introduir el nom de l'àncora.

1.10.5. Gestió d'imatges

Nvu proporciona funcionalitats per inserir imatges dins les pàgines web i dotar-les de les propietats de visualització que interessi. Per aconseguirho disposem del botó *Imatge* de la barra d'eines de composició o del menú *Insereix*|*Imatge* que provoca l'aparició de la finestra modal de la figura 29. En ella hi trobem vàries pestanyes:



a) Pestanya *Ubicació*, amb totes les propietats relatives a la ubicació de la imatge:

- Ubicació, per indicar l'arxiu imatge, el qual pot residir en el mateix espai web que la nostra pàgina (utilitzarem ruta relativa i activarem la casella corresponent següent) o en un espai web estern (utilitzarem ruta absoluta)
- Rètol indicatiu (*tooltip*), per indicar el text que el navegador visualitzarà quan el ratolí passi per damunt la imatge.
- Text alternatiu, per especificar el text que es mostrarà enlloc de la imatge en aquells navegadors que no siguin capaços de mostrar la imatge o si l'usuari ha desactivat la visualització de imatges. Si no es vol indicar, cal marcar l'opció *No utilitzis un text alternatiu*.

La paraula *inserció* utilitzada per inserir una imatge en el document pot donar peu a pensar que la imatge queda incrustada en el document com passa en molts processadors de textos i això no és així. En el document només hi ha la ruta per accedir a la imatge i, per tant, quan el document es pugi al servidor web també hi haurà d'haver la imatge en el lloc adequat segons la ruta indicada.

b) Pestanya *Dimensions*, per indicar si es vol mantenir la grandària original de la imatge o si es vol alterar en la visualització.

No té gaire sentit ser els propietaris d'una imatge a visualitzar en una única pàgina web i indicar-ne una visualització més petita que la grandària de la imatge, doncs la grandària és el factor que penalitza en el temps d'accés a la imatge i, per tant, sembla més lògic tenir la imatge emmagatzemada amb la grandària de visualització.

Sí pot tenir sentit, però, disminuir la grandària quan hem d'accedir a una imatge de la que no en som els propietaris i no tenim possibilitat de disminuir-ne el seu pes o malgrat en siguem els propietaris, aquesta imatge es visualitza amb unes dimensions més grans en altres pàgines web.

Per últim, comentar que el fet d'augmentar la grandària de visualització d'una imatge respecte la seva grandària original provoca pèrdua de resolució en la visualització.

c) Pestanya *Aparença* per governar el posicionament de la imatge i del text al seu voltant. Facilita els següents apartats:

- Espaiat, per indicar l'espai buit que es vol en certes zones al voltant de la imatge:
 - D'esquerra a dreta
 - Per sobre i per sota
 - En un contorn sòlid
- Alinea el text a la imatge, per indicar com s'ha de situar la imatge respecte el text que pugui acompanyar-la.

d) Pestanya *Enllaç* per permetre associar un enllaç a la imatge de la mateixa manera que s'insereixen enllaços en el text.

Com exemple d'utilització, afegim a la pàgina web que hem anat dissenyant, el logotip de FP Oberta (imatge *FPOberta.gif* que suposem es troba en la mateixa ubicació que la nostra pàgina web) acompanyat d'un text que expliqui que prement damunt ella s'accedeix a la pàgina principal del campus de FP Oberta i dotant a la foto del corresponent enllaç. La figura 30 en mostra la visualització "marcadors HTML".

| • | sense titol) | Elaboracio de pagines web amb Ile | ` | <u> </u> | | |
|--|---|-----------------------------------|--|---------------|--|--|
| | 706рх | | | | | |
| | E E | oducció | | | | |
| Internet, la xarxa de comunicacions que posa en contacte milions de sistemes informàtics repartits per tot el plat facilita diversos serveis als seus usuaris: correu electrònic (e-mail), xerrades (xat), pujar i baixar fitxers (FTP), tat d'anuncis (news), vídeo conferència i, el seu servei més utilitzat, la teranyina de documents (pàgines web), form per un immens conjunt de documents que poden tenir contingut textual, gràfic, àudio, animacions, etcètera. | | | | | | |
| En les unitats didàctiques precedents heu anat treballant els conceptes vinculats amb l'explotació d'un ll com els procediments a emprar per actualitzar-ne els continguts i la normativa legal vigent. En aquesta unit aprendreu com mantenir i dissenyar, d'una forma correcta, pàgines web que siguin atractives i entenedore que hi hagi un alt grau de probabilitat de ser visitades per molts usuaris de la xarxa. | | | t els conceptes vinculats amb l'explotació d'un lloc web ai ntinguts i la normativa legal vigent. En aquesta unitat didàctic :ta, pàgines web que siguin atractives i entenedores, de form r molts usuaris de la xarxa. | кí a la | | |
| 52px | ■ MG ∰ Generalitat de C. A FP Ot | Clicant damunt la im | atge accedireu a la pàgina principal del campus de FP Obert | a | | |
| | Normal 🔟 Marcadors HTML 🔽 Codi font HTML 🔍 Previsualització | | | | | |

Nvu no permet, en definir un enllaç en una imatge, indicar la ubicació on s'obrirà l'enllaç, mentre si que ho permetia en definir un enllaç en text.

Figura 30. Exemple d'inserció d'imatge en una pàgina web

Això no ha de ser problema, doncs podem anar a modificar el codi HTML, el qual quedaria com segueix:

```
    <a href="http://www.fpoberta.net" target="_blank" >
    <img style="border: 0px solid ; width: 151px; height: 50px;"
        alt="Logotip FP Oberta" src="FPOberta.gif" align="middle"
        hspace="0" vspace="0">
        </a> &nbsp;Clicant damunt la imatge accedireu a la p&agrave;gina principal del campus de FP Oberta
```

En aquest codi, l'atribut target="_blank" ha estat afegit manualment per a poder aconseguir que la pàgina enllaçada s'obri en una nova pestanya o finestra. A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina14.html* amb el codi de l'exemple.

1.10.6. Gestió de taules o graelles

Nvu proporciona funcionalitats per definir i gestionar taules dins les pàgines web. Per aconseguir-ho disposem del botó *Taul.a* de la barra d'eines de composició o del menú *Insereix* | *Taula* que provoca l'aparició de la finestra modal de la figura 31. En ella hi trobem vàries pestanyes:

a) Pestanya *Ràpidament*, que permet crear una taula de forma molt intuïtiva doncs només cal seleccionar amb el ratolí el número de files i de columnes desitjades i la taula es crea automàticament.



b) Pestanya *Amb precisió*, que permet indicar el número de files i de columnes i l'amplada total de la taula.

c) Pestanya *Cel·la*, que permet especificar les propietats de les cel·les que es crearan utilitzant qualsevol dels medis anteriors.

La creació, per tant, és simple. Ens interessa poder disposar de mecanismes per a modificar-ne les propietats i també per omplir-la de continguts.

Com omplir les cel·les?

©Isidre Guixà i Miranda

Omplir les cel·les d'una taula des de *Nvu* és ben simple, doncs només cal situar-se amb el ratolí dins una cel·la i començar a escriure-hi o inserir-hi imatges o, fins i tot, crear-hi noves taules.

Les cel·les s'adapten automàticament al contingut que s'hi insereix i si el text és molt llarg automàticament es situarà en una nova línia.

Com modifiquem les dimensions?

En HTML no és possible modificar l'amplada d'una única cel·la, sinó que s'ha de modificar l'amplada de tota la columna.

L'amplada de les columnes i també l'alçada de les files es pot governar visualment amb les regles que envolten la zona d'edició. Observar (figura 32) que quan estem situats en una taula, les regles horitzontal i vertical marquen, respectivament, les amplades de les diferents columnes i les alçades de les diferents files.

Prement el ratolí damunt de les separacions entre els blocs de les regles (figura 32) i arrossegant-los cap esquerra o dreta (regla horitzontal) o cap amunt o avall (regla vertical) aconseguim modificar l'amplada de les columnes i l'alçada de les files.

Així mateix al voltant de la taula (figura 32) apareixen uns botons que en permeten modificar la seva grandària.



Com eliminar o afegir una fila o una columna?

A les parts superiors i esquerra de la cel·la en la que està situat el cursor hi apareixen uns botons amb una creuta (figura 33) que permeten eliminar tota la columna o tota la fila. Al costat del botó hi ha unes fletxes que permeten afegir files i columnes.

Figura 33. Elements per modificar visualment la taula

| • | sense tîtol) | | * | | | | |
|--------------------------|---|-------|--|--|--|--|--|
| | 200px | 200px | | | | | |
| 43px 43px 43px 43px 43px | | | Fletxes per afegir una nova fila (per damunt o per sota) o una nova columna (a l'esquerra o a la dreta | | | | |
| | Botons p'er'eliminar tota la fila o tota la columna | | | | | | |
| | 🗅 Normal 🔟 Marcadors HTML 🔯 Codi font HTML 🔍 Previsualització | | | | | | |

Per eliminar una columna o una fila o, fins i tot, la taula sencera, també es pot prémer el botó secundari del ratolí tenint el cursor situat a la zona a eliminar i seleccionar en el menú contextual que apareix (figura 34) l'opció *Suprimeix Taula* que a la seva vegada ens permet eliminar la fila o la columna o la taula o la cel·la o el contingut de la cel·la.

Figura 34. Menú contextual en prémer el botó secundari del ratolí damunt una taula



Propietats de la taula i de les cel·les

Si premem el botó *Taula* de la barra d'eines de composició tenint el cursor situat en alguna part d'una taula, *Nvu* ens presenta la finestra modal de la figura 35. Observem que té dues pestanyes: la primera (figura 35) on podem modificar les propietats que afecten al conjunt de la taula i la segona (figura 36) on podem modificar les propietats de les diferents cel·les de la taula.

Les dues pestanyes incorporen el botó *Edició avançada* que permet donar valors els atributs HTML que governen la funcionalitat de les taules. Els apartats de les pestanyes permeten definir aquests valors sense necessitat de saber-ne el nom dels atributs.

| Figura 35. Pantalla Nvu per definir les propietats d'una taula | | | | |
|--|--|--|--|--|
| Propietats de la taula 🛛 🛛 🔀 | | | | |
| Taula Cel·les | | | | |
| _ Mida | | | | |
| Eiles: 4 Alçada: % de la finestra 💙 | | | | |
| | | | | |
| Contorn i espaiat | | | | |
| Contorn: 2 píxels | | | | |
| Espaiat: 2 píxels entre cel·les | | | | |
| Separació: 2 píxels entre el contorn de les cel·les i el contingut | | | | |
| Alineació de la taula: Esquerra 🔻 Subtitol: Cap 💌 | | | | |
| Color de fons: (permet al color de la pàgina mostrar-se a través) | | | | |
| Edició avançada | | | | |
| D'acord Aplica Cancel·la Ajuda | | | | |

- 1) Pestanya *Taula* (visualitzada a la figura 35)
- L'apartat *Mida* a la part superior de la finestra permet definir el número de files i de columnes. A la seva dreta es pot fixar l'alçada i l'amplada de la taula. L'altura d'una taula no s'acostuma a definir.

Dins les llistes desplegables que acompanyen l'alçada i l'amplada hi ha dues unitats de mesura possibles:

- Píxels, que permet definir un valor fix
- % *de la finestra*, que permet definir un valor percentual

L'avantatja dels valors percentuals és que els navegadors intenten fer la millor adaptació possible segons l'amplada de la pàgina.

L'avantatja dels valors fixes és que es pot controlar amb molta precisió la presentació de la taula.

- L'apartat Contorn i espaiat en mig de la finestra permet definir:
 - L'espai, en píxels, que ocupa el contorn de la taula (valor de l'atribut border de la marca en HTML)
 - L'espaiat, en píxels, que hi ha entre les cel·les (valor de l'atribut cellspacing de la marca en HTML)
 - La separació que hi ha entre el contorn de les cel·les i el seu contingut (valor de l'atribut cellpadding de la marca en HTML) per aconseguir que el contingut de les cel·les no quedi enganxat a la seva vora.
- L'apartat *Alineació* que permet indicar la posició de la taula respecte la pàgina (atribut align de la marca en HTML)

- L'apartat *Subtítol* que permet indicar si la taula tindrà un títol (marca <caption> en HTML) i la seva ubicació.
- L'apartat *Color de fons* que permet indicar el color de fons de la taula i també deixar-lo com a transparent, fet que provocarà que el color de fons de la pàgina sigui el color de fons de la taula.
- 2) Pestanya Cel·les (visualitzada ala figura 36)
- L'apartat *Selecció* permet indicar si les propietats a definir afecten a una cel·la o a tota una columna o a tota una fila.
- L'apartat *Mida*, de forma similar a la pestanya *Taula*, permet definir la mida de la cel·la/fila/columna. Les mesures es poden indicar en valors absoluts (pixels) o relatius (percentatge).

| Propietats de la taula | | | | | |
|---|--|--|--|--|--|
| Taula Cel·les | | | | | |
| Selecció Cel·la ✓ Anterior ▶ Següent S'aplicaran els canvis actuals abans de canviar la selecció. | | | | | |
| Mida Alçada: píxels Alineació del contingut Vertical: Mig Horizontal: Esquerra | | | | | |
| Estil de cel·la: Normal Ajusta el text: Ajusta Color de fons: (permet al color de la taula mostrarse a través) | | | | | |
| | | | | | Utilitza quadres de verificació per determinar quines propietats Edició avançada |
| D'acord Aplica Cancel·la Ajuda | | | | | |

Figura 36. Pantalla Nvu per definir les propietats de les cel·les d'una taula

- L'apartat Alineació del contingut permet indicar l'alineació del contingut de la cel·la/fila/columna. L'alineació horitzontal es correspon amb l'atribut align i l'alineació vertical amb l'atribut valign de les marques (fila) o (columna).
- L'apartat *Estil de cel·la* que permet definir si la cel·la és normal (definida per la marca) o si és capçalera (definida per la marca >).
- L'apartat *Ajusta el text*, que permet definir l'alineació del text de la cel·la.
- L'apartat *Color de fons* que permet indicar el color de fons de la cel·la/fila/columna i també deixar-lo com a transparent, fet que provocarà que el color de fons de la taula el color de fons de la cel·la/fila/columna.

1.10.7. Gestió de llistes

Nvu proporciona a la barra d'eines de format diferents botons per a la gestió de llistes: *Llista numerada* (ordenada), *Llista de pics* (no ordenada), *Terme de definició* i *Descripció de definició*. També podem utilitzar el menú *Format*|*Llista*. La funcionalitat d'aquests botons es pot ampliar amb la utilització dels botons per posar/treure sagnat.

Per modificar les propietats d'una llista també disposem del menú *Format*|*Propietats llista* en estar situats damunt de la llista. Ens apareix la finestra modal de la figura 37 on podem definir el tipus de llista (ordenada, no ordenada, de definició) així com l'estil dels elements per a les llistes ordenades (numeració aràbiga, lletres, numeració romana,...) i per a les llistes no ordenades (cercle sòlid, quadrat,...)

| igura or i a mana por a dom | |
|------------------------------------|--|
| Propietats de la llista 🛛 🛛 🔀 | |
| Tipus de llista | |
| Llista de pics (sense numerar) 🛛 🗸 | |
| Estil de pic | |
| Automàtic 💌 | |
| Comença a: | |
| Edició avançada | |
| D'acord Cancel·la | |

Figura 37. Pantalla per a definir les propietats d'una llista

1.10.8. Ortografia

És d'agrair la possibilitat de correcció ortogràfica que proporciona *Nvu* per medi del botó *Verifica* de la barra d'eines de composició o del menú *Edita*|*Comprova l'ortografia*, que mostra la finestra modal de la figura 38.

Les llengües que sàpiga verificar dependrà de la instal·lació que se n'hagi fet i de les extensions que tingui instal·lades. La figura 38 mostra per on es pot seleccionar la llengua a verificar i els diccionaris incorporats en la instal·lació en llengua catalana.



2. Aprofundiment en el disseny de pàgines web

La finalitat d'aquest nucli és aprofundir en el disseny de pàgines web amb la utilització de recursos avançats que facilita HTML. Utilitzarem, sempre que sigui possible, un editor específic de pàgines web (Nvu) per aplicar d'una manera ràpida aquests recursos, sense deixar de banda el coneixement del corresponent codi HTML, doncs en ocasions haurem de retocar pàgines sense disposar de cap editor específic. Fins i tot, certs recursos els haurem d'aplicar directament amb codi HTML si l'editor específic no incorpora la seva utilització.

2.1. Estils i fulls d'estil

En els orígens de la web i en les seves primeres versions, HTML era un llenguatge fàcil d'aprendre i molt reduït en quant a les seves marques i estructura. Tot va canviar quan van aparèixer els primers navegadors que eren capaços de representar recursos gràfics com afegit a la informació textual.

D'aquesta manera el número de llocs web va començar a créixer i amb ell el número de marques que l'especificació HTML contemplava. L'objectiu era construir llocs web cada vegada més atractius i per aconseguir-ho HTML havia d'incloure noves marques destinades a aconseguir efectes visuals.

Amb tots aquests canvis que la web havia sofert ens trobem amb un llenguatge que havent estat en els seus inicis "orientat a l'estructura" ara està totalment "orientat a la presentació".

Un exemple d'aquest fet el trobem en marques com , <u> o <i> que defineixen estils de visualització sense aportar res a l'estructura del document representat. Un fet similar el podem trobar en la utilització errònia de la marca per a construir un títol enlloc d'utilitzar la corresponent marca <h#>, situació que provoca el deteriorament de l'estructura del document.

Així doncs, amb la proliferació de la utilització de marques de presentació, estem aconseguint que a la majoria de pàgines web hi hagi més informació de visualització que no pas informació verdaderament rellevant i això és preocupant.

Hi ha motius pels que no podem permetre que els documents publicats a la web perdin la seva estructura:

- La indexació pels cercadors és molt més complicada. Així, per exemple, Google assigna un pes més alt a una pàgina que utilitza marques <h#> per a definir les seves seccions que als camps <meta> del document
- Es redueix l'accessibilitat. Actualment existeixen aplicacions que faciliten la lectura de pàgines web a persones discapacitades i si no hi ha una mínima estructura, el resultat pot ser lamentable.
- L'estructura de la pàgina i la informació continguda en ella és molt més senzilla de mantenir. Actualment certs aspectes del codi HTML poden fer que una mateixa pàgina tingui visualitzacions diferents en diferents navegadors. Aquests errors són de difícil depuració quan la pàgina conté una estructura de marques complicada i sense cap estructura. Per altra banda, un canvi en un tipus de font suposa tornar a dissenyar totes les pàgines d'un lloc web en haver de substituir totes les marques afectades.

Tots aquests problemes han provocat que apareguessin els estils i els fulls d'estil que separen la informació de presentació de la informació d'estructura del document.

La utilització d'estils i fulls d'estil aporta, en contraposició a la utilització dels elements de presentació estàndards d'HTML:

- Creació de documents visualment més rics que HTML doncs la utilització d'estils està pensada únicament i exclusiva per assistir al dissenyador a l'hora de donar estil a un document estructurat.
- Reducció de la càrrega de treball en dissenyar tot un lloc web doncs les fulles d'estil permeten centralitzar en una ubicació certs efectes visuals que s'han de fer visibles en diverses seccions del lloc web enlloc d'haver-los de tenir disseminats per pàgines i pàgines del lloc.
- Reutilització en diverses pàgines del lloc web, de manera que és senzill generar un estil general del lloc web i mantenir-lo consistent per a totes les pàgines, de manera que si hem de modificar un estil que és comú a totes les pàgines només haurem de modificar-lo en el lloc on està definit.
- Reducció de la grandària de codi HTML enviat al navegador
- Preparació per al futur doncs ajuda a deixar d'utilitzar les marques HTML que *W3C* ha marcat com a obsoletes i que deixaran de ser utilitzades en un futur.

I per garantir la uniformitat d'utilització d'estils i fulls d'estil, *W3C* ha definit l'estàndard *CSS*, acrònim de *Cascading Style Sheets* (Fulls d'Estil en Cascada).

2.1.1. Sintaxi bàsica dels fulls d'estil

CSS funciona en base a regles, és a dir, declaracions sobre l'estil d'un o més elements. Els fulls d'estil són un conjunt d'aquestes regles que s'apliquen a un document HTML (i també a documents XML).

Una regla té dues parts: un selector i la declaració. A la seva vegada, la declaració està composta per una propietat i el valor que se li assigna.

Exemple:

```
a {color: white; background-color: black;}
```

- a és el selector i correspon a la marca <a> de l'HTML
- {color: white; background-color: black;} és la declaració que en aquest cas conté dues propietats (separades per punt i coma):
 - color, amb el valor white
 - background-color, amb el valor black

Aquesta regla provoca, allà on s'apliqui, que el text dels enllaços (gestionats per la marca $\langle a \rangle$) es vegin en color blanc sobre fons negre.

Per a comprovar-ho enregistreu aquesta regla en un document pla i en el navegador *Internet Explorer* indiqueu aquest document en el menú *Eines* |*Opcions d'internet* |*General* |*Accessibilitat*, opció *Formateu els documents amb el meu full d'estils.* Vegeu, a la dreta de la figura 39, com es mostren els enllaços quan el navegador té assignat aquest full d'estils.

El selector funciona com enllaç entre el document i l'estil, especificant els elements que es veuran afectats per aquesta declaració.

La declaració és la part de la regla que estableix quin serà l'efecte.

Extensió dels fitxers que contenen Fulls d'Estil

És un bon costum utilitzar l'extensió .*css* per als fitxers que contenen fulls d'estil.



Figura 39. Exemple d'aplicació d'un full d'estils personal en un navegador.

Les tres formes més conegudes de donar estil a un document són les següents:

- Utilitzant un full d'estil extern que estarà vinculat a un document a través d'una marca <link> que ha d'estar situada en la secció <head>.
- Utilitzant la marca <style> a l'interior del document al que es vol donar estil. Generalment es situa a la secció <head> i d'aquesta manera l'estil és reconegut abans que la pàgina es carregui del tot.
- Indicant estils directament sobre aquells elements que ho permeten a través de l'atribut style dins de la secció <body>. Aquest tipus d'estil perd les avantatges que donen els fulls d'estil en barrejar-se el contingut amb la presentació. No segueix la sintaxi "selectordeclaració" sinó que directament s'assigna a l'atribut style la seqüència de propietats amb els valors corresponguin:

<marca style="propietat: valor; propietat:valor;...">...</marca>

Els editors específics de pàgines web, com el *Nvu*, utilitzen aquest darrer mètode en generar el codi HTML a partir de les característiques de presentació que el dissenyador ha indicat. Així, per exemple, si en *Nvu* creem un document nou, hi escrivim un títol amb les característiques d'*Encapçalament 1*, centrant-lo i assignant-li el color vermel, *Nvu* ens genera el codi:

 $Observem \ que \ la \ marca < \texttt{h1}> \ que \ determina \ el \ títol \ porta \ l'atribut \ \texttt{style} \\ amb \ la \ declaració \ de \ dues \ propietats: \ \texttt{text-align} \ i \ \texttt{color}.$

Algunes normes bàsiques a l'hora de crear un full d'estil són:

Els selectors poden aparèixer individualment o agrupats, separats per comes.

Així, per exemple, si a tots els títols d'un document volem assignar el color vermell, podem posar:

| h1 | { | color: | red; | } |
|----|---|--------|------|---|
| h2 | { | color: | red; | } |
| h3 | { | color: | red; | } |
| h4 | { | color: | red; | } |
| h5 | { | color: | red; | } |
| h6 | { | color: | red; | } |

o agrupant-ho tot en:

h1,h2,h3,h4,h5,h6 { color: red; }

• En cas que per a un selector (o grup de selectors) hi hagi vàries propiestats a assignar, es poden agrupar separant cadascuna d'elles amb un punt i coma, com es veu en el següent codi:

a {color: white; background-color: black;}

La darrera propietat no té per què anar seguida de punt i coma, però és usual posar-li.

• Per a fer més llegible el full d'estil, s'acostuma a descriure una propietat per línia, com es veu en el següent codi:

```
a {
    color: white;
    background-color: black;
}
```

• En cas que una propietat tingui més d'un valor, els diferents valors es separen amb una coma, com es veu en el següent codi:

p {font-family: Agency FB, Helvetica, Arial, sans-serif;>

• En cas que un valor estigui format per vàries paraules, caldrà escriure'l entre cometes:

p {font-family: Agency FB,Helvetica,Arial,"sans serif";>

2.1.2. Assignació de fulls d'estil externs a documents

Per aconseguir l'adequada càrrega d'un full d'estils hem de posicionar la marca <link> a la secció <head> de la pàgina, seguint la sintaxi:

<link rel="stylesheet" type="text/css" href="estils.css" title="default">

Els atributs de la marca <link> són:

- rel, que defineix el tipus del document amb el que establim la relació. En general el seu valor és "stylesheet".
- type, que sempre serà "text/css".
- href, que és el valor de la URL que ens porta al fitxer que conté el full d'estils.
- title, que actualment quasi no s'utilitza però que serà molt important en el futur per a permetre que el visitant del document pugui decidir entre diversos fulls d'estil alternatius vinculats al document

Un document HTML pot tenir vàries marques <link> amb el valor "stylesheet" en l'atribut rel. En tal cas, el navegador carregarà els diversos fulls d'estil combinant les regles que defineixen i aplicarà el resultat al document.

Un document HTML pot tenir assignats un o més fulls d'estil alternatius per a ser utilitzats per l'usuari. Per aconseguir-ho cal utilitzar la marca <link> assignant el valor "alternate stylesheet" a l'atribut rel.

En la càrrega d'un document, els fulls d'estil persistents indicats amb el valor "stylesheet" en l'atribut rel de les marques <link> són aplicats al document. Els fulls d'estil alternatius han de poder ser seleccionats pel visitant del document. *W3C* recomana que el navegador hauria de proporcionar un medi per a que l'usuari pugui escollir el full d'estils alternatiu que desitgi. En aquest cas l'atribut title té interès.

Mozilla Firefox proporciona un menú per seleccionar els fulls d'estils alternatius via el menú *Visualitza*|*Estil de pàgina*. En canvia, *Internet Explorer* (fins a versió 7.0) no proporciona cap menú i, per tant, l'usuari no pot escollir entre els fulls d'estil alternatius proporcionats per la pàgina.

Com a exemple de tot això, observem la següent pàgina web:

| | |
|--|---|
| En ella es fa referència a dos fitxers de fulls d'estil: | A la secció "Recursos" del web d'aquest crèdit hi podeu trober ele fituere |
| • Fitxer <i>estalt01.css</i> : | <i>pagina15.html, estalt01.css</i> i <i>estalt02.css</i> corresponents a l'exemple. |
| h1,h2,h3,h4,h5,h6 {color: red;} | |
| • Fitxer <i>estalt02.css</i> : | |

La figura 40 ens mostra com el navegador *Mozilla Firefox* permet, una vegada la pàgina està carregada, que l'usuari pugui escollir entre els diversos fulls d'estil alternatius identificats pel seu títol.



h1, h2, h3, h4, h5, h6 {color: green;}



2.1.3. Definició de fulls d'estil interns en un document

La definició completa d'un full d'estil es pot efectuar dins el propi document, a la zona <head> amb la utilització de la marca <style> i finalitzant el full d'estils amb la marca </style>.

En aquest cas, la marca <style> ha de contenir obligatòriament l'atribut type amb el valor "text/css".

La marca <style> pot contenir enllaços a fulls d'estils esterns definits amb la directiva @import enlloc de la marca <link>. No es pot, però, indicar fulls d'estil alternatius.

En cas de tenir definits enllaços externs amb la directiva @import, aquests han d'estar situats a l'inici de la zona definida per la marca <style>.

©Isidre Guixà i Miranda

67

Com a exemple serveixi el següent codi:

En aquest codi s'aplica en primer lloc l'estil del fitxer estil16.css:

```
h1,h2,h3,h4,h5,h6 {text-align: center;}
```

que provoca que els títols estiguin centrats i, posteriorment, s'aplica l'estil intern que provoca que els títols siguin en color vermell.

2.1.4. Definició de fulls d'estil en Nvu

L'editor *Nvu* facilita la possibilitat de definir fulls d'estil en el menú *Eines* |*Editor CSS*. En la pantalla que ens apareix (com la de la figura 41) podem crear totes les definicions d'estils en *Nvu*.

Editem amb *Nvu* el fitxer *.html* del darrer exemple i activem l'eina *Editor CSS*. Obtindrem la pantalla de la figura 41 en la que podem observar, a la part esquerra, la totalitat d'estils aplicats al document així com les regles contingudes en cada estil. Situant-nos damunt una regla qualsevol, ens apareix informació rellevant a les pestanyes de la part dreta de la pantalla.

| Figura 41. Pantalla <i>Nvu</i> | per a edició de fulls d'estil |
|--------------------------------|-------------------------------|
|--------------------------------|-------------------------------|

| Fulls d'estil CSS | | \mathbf{X} |
|---|---|--------------|
| Estils i regles ■ fulla d'estils interna ■ @mport estil 16.css ↓ <i>h1, h2, h3, h4, h5, h6</i> ↓_h1, h2, h3, h4, h5, h6 | Amunt General Text Fons Contorns Caixa Llistes Aural Avail Regla d'estil Selector: h1, h2, h3, h4, h5, h6 Dedaracions: Import color: red Import | ància |
| | Refresca | |
| ©import @media Enilac elk. Esti ek. Regla | Suprimeix | |
| ✓ Mode expert | Tanc | a |

A la secció "Recursos" del web d'aquest crèdit podeu trobar els fitxers pagina16.html i estil16.css amb el codi de l'exemple. Vegem com utilitzar aquesta eina que ens facilita *Nvu* i per a fer-ho partirem d'un document que encara no tingui cap estil definit. Concretament ho farem amb el document visualitzat a la figura 42.

En primer lloc, si la casella *Mode expert* de l'extrem inferior esquerra està activat, el desactivem per a indicar-li a *Nvu* que organitzi les declaracions d'estil i nosaltres ens limitarem a definir les regles. Fixeuvos que en desmarcar aquesta casella, només queda actiu el botó *Regla*.

Figura 42. Document sense cap estil sobre el que començarem a aplicar-ne des de Nvu. 😻 Elaboració de pàgines web amb llenguatges de marques - Mozilla Firefox Eitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 -
 🔹 🕨 🚺 Google Q 🗋 Hotmail gratuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windows Introducció Internet, la xarxa de comunicacions que posa en contacte milions de sistemes informàtics repartits per tot el planeta, facilita diversos serveis als seus usuaris: correu electrònic (e-mail), xerrades (xat), pujar i baixar fitxers (FTP), taulers d'anuncis (news), vídeo conferència i, el seu servei més utilitzat, la teranyina de documents (pàgines web), formada per un immens conjunt de documents que poden tenir contingut textual, gràfic, àudio, animacions, etcètera En les unitats didàctiques precedents heu anat treballant els conceptes vinculats amb l'explotació d'un lloc web així com els procediments a emprar per actualitzar-ne els continguts i la normativa legal vigent. En aquesta unitat didàctica aprendreu com mantenir i dissenyar, d'una forma correcta, pàgines web que siguin atractives i entenedores, de forma que hi hagi un alt grau de probabilitat de ser visitades per molts usuaris de la xarxa **FP Oberta**

En prémer el botó Regla s'activa la pestanya General a la dreta (figura 43)

Figura 43. Pestanya de l'Editor CSS de *Nvu* per assignar una nova regla General Text Fons Contorns Caixa Llistes Aural New Style rule

 Crea un nou:

 • named style (enter class name below)
 • style applied to all elements of type (enter type below)
 • create Style rule
 • O • Create Style rule
 • Create Style rule
 • O • Create Style rule
 • Create Style rule
 • Create Style rule
 • O • Create Style rule

El document visualitzat a la figura 42 el podeu trobar a la secció "Recursos" del web d'aquest crèdit com arxiu pagina 14. html.

Versió catalana de *Nvu*

En aquest material s'està treballant amb la versió catalana de *Nvu* però, com es pot veure a la figura 42, alguna pantalla encara està pendent de traducció!

En aquest punt *Nvu* ens deixa de crear dos tipus de regla: regles per a les marques estàndards d'HTML o regles per a classes creades per nosaltres. Nosaltres ens limitarem a definir regles per a les marques estàndards d'HTML i això correspon a la segona opció que ens facilita *Nvu: estil aplicat a tots els elements del tipus...*

Una vegada escollida la segona opció, hem d'indicar en el camp següent, el selector, és a dir, les marques sobre les que s'aplicarà aquesta regla que estem definint i, en aquest punt, és necessari tenir coneixements d'HTML doncs *Nvu* no ens proporciona facilitats per escollir les marques (igual en versions futures hi haurà ajudes...). Una vegada definida la

Classes en HTML

En HTML és possible definir classes d'objectes i aplicar estils a les classes, però això s'escapa de l'abast d'aquest material. marca (selector) premem el botó *Crear regla d'estil* i ja ens apareix la regla a la llista de l'esquerra. Però encara no hem definit la declaració (no hem donat valors a les propietats sobre les que ens interessa incidir).

Per assignar les diferents propietats amb els seus valors, l'*Editor CSS* facilita diverses pestanyes a la seva part dreta. Coneguem-les a partir de diverses situacions.

• Com assignar marges a la pàgina?

Com que es tracta d'una propietat que afecta a tota la pàgina, caldrà definir una nova regla d'estil sobre la marca <body>. Les propietats corresponents a marges resideixen a la pestanya *Caixa* de la part dreta de l'*Editor CSS* (figura 44). Aquesta pestanya inclou moltes propietats, de les quals ens interessen ara les de la part inferior relatives als marges.

General Text Fons Contorns Caixa Llistes Aural Visualització: Visibilitat: sense especificar sense especificar Y × Coma flotant: ¥ sense especificar Neteja: sense especificar V Overflow: sense especificar Posició: ¥ sense especificar Z-index: Amplada: Alçada: Amplada mín.: Alçada mín.: Amplada max. Alçada max. Offsets Separacions: Superior Esquerra × Y ~ Dreta: Y ¥ Inferior: Tanca

En anar definint diversos valors pels marges, *Nvu* va modificant la visualització de la pàgina per a que el dissenyador sigui conscient de l'efecte de la propietat que està modificant. Així, si posem uns marges com: superior a 40 px, esquerra a 15% i dreta a 15% obtenim la visualització de la figura 45 que podem comparar amb la inicial figura 42.

És interessant veure el codi que Nvu ha generat i que ja hem d'entendre sense cap problema:

```
<head>
...
<style type="text/css">
body { margin-top: 40px;
margin-right: 15%;
margin-left: 15%;
}
</style>
...
</head>
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina17.html* amb el codi de l'exemple.

```
Figura 44. Pestanya de l'Editor CSS de Nvu per a definir dimensions i marges
```



Com assignar un color de fons o una imatge de fons?

Si volem modificar el color blanc de fons d'una pàgina, cosa que ja sabem fer pel menú *Format*|*Colors de la pàgina i del fons* (opció que crea un atribut style dins la marca <body> del document), amb la utilització d'una regla d'estil, sembla que correspondrà ampliar les propietats definides sobre la regla que ja tenim definida per la marca <body>. Així doncs, en l'*Editor CSS*, seleccionem la regla definida sobre la marca <body> i activem la pestanya *Fons* de la part dreta de l'*Editor CSS* (figura 46) des d'on podem definir un color de fons per a la pàgina o assignar-hi una imatge de fons.



Figura 46. Pestanya de l'Editor CSS de *Nvu* per a definir un fons per a la pàgina

El resultat en codi HTML d'assignar un color de fons es tradueix en l'ampliació de la marca <style> ja existent a la secció <head> amb:

```
<head>
...
<style type="text/css">
body { margin-top: 40px;
margin-right: 15%;
margin-left: 15%;
background-color: rgb(255, 255, 102);
}
</style>
...
</head>
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina18.html* amb el codi de l'exemple.

• Com modificar el color per defecte del text normal d'una pàgina?

Sembla que correspondrà ampliar encara més les propietats existents sobre la regla que ja tenim definida per la marca <body>. Així doncs, en l'*Editor CSS*, seleccionem la regla definida sobre la marca <body> i activem la pestanya *Text* de la part dreta de l'*Editor CSS* (figura 47) des d'on podem definir un color de fons per a la pàgina o assignar-hi una imatge de fons.

Figura 47. Pestanya de l'Editor CSS de *Nvu* per a definir propietats de text General Text Fons Contorns Caixa Llistes Aural



El resultat en codi HTML d'assignar un color al text normal de la pàgina es tradueix en l'ampliació de la marca <style> ja existent a la secció <head> amb:

```
<head>
...
<style type="text/css">
body { margin-top: 40px;
margin-right: 15%;
margin-left: 15%;
background-color: rgb(255, 255, 102);
color: rgb(0, 0, 153);
}
</style>
...
</head>
```

• De la mateixa manera que hem creat una regla per la marca <body> a la que hem anat incorporant diverses propietats, podem definir A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina19.html* amb el codi de l'exemple. regles per a altres marques. Així, per exemple, anem a definir una regla per als títols de tipus 1 de manera que el seu color sigui vermell.

Utilitzant l'*Editor CSS*, premerem el botó *Regla* a la part esquerra i demanarem de crear una nova regla sobre l'element h1. Posteriorment anirem a la pestanya *Text* i assignarem el color vermell. El resultat en codi HTML es tradueix en l'ampliació de la marca <style> ja existent a la secció <head> amb:

```
<head>
...
<style type="text/css">
    body { margin-top: 40px;
    margin-right: 15%;
    margin-left: 15%;
    background-color: rgb(255, 255, 102);
    color: rgb(0, 0, 153);
    }
    h1 { color: rgb(255, 0, 0);
    }
</style>
...
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina20.html* amb el codi de l'exemple.

2.1.5. L'efecte "cascada" dels fulls d'estil

Coneixem els estils i els fulls d'estil, però les sigles *CSS* es tradueixen per "fulls d'estil en cascada". On és la "cascada"?

El concepte de "cascada" és fonamental en la utilització dels estils i fulls d'estil. El principi és simple: un estil s'aplica a un objecte i a tots els objectes que aquest conté, també anomenats "fills". És a dir, els "fills" hereten les regles definides en els "pares". Aquesta herència o cascada només té sentit en un únic sentit: de pare s a fills. Es verifica que:

- Si modifiquem una propietat en un objecte fill, l'objecte pare no es veurà afectat per la modificació.
- Si una propietat està definida en un objecte pare i també es defineix en un objecte fill, la propietat del pare és ignorada.

Això ho podem observar en els darrers exemples:

- En haver assignat el color blau a tot el text (marca <body>), tot el text (títols inclosos) es visualitza de color blau.
- En assignar el color vermell als títols (marca <h1> que es troba dins <body>) el text que no correspon a títols gestionats per la marca <h1> continua apareguent blau (com mana el pare <body>), mentre que el títol gestionat per la marca <h1> apareix vermell.
2.2. Recursos multimèdia tradicionals

La web és un medi de comunicació ideal per a presentar informació multimèdia, és a dir, informació que utilitza diferents medis de comunicació: text, imatge, animació, vídeo i so.

A hores d'ara sabem incloure text i imatges en els nostres documents web i ens manca saber gestionar la resta de possibilitats.

2.2.1. Mapes sensibles

Els mapes sensibles són imatges que presenten múltiples enllaços segons la zona on es situï el cursor. Per aconseguir això es divideix la imatge en zones mitjançant les coordenades de les mateixes.

Hi ha editors específics de pàgines web que incorporen mecanismes per a poder trossejar la imatge en zones capturant les coordenades que les defineixen. Però *Nvu* no ho permet, així que ens veiem obligats a fer-ho d'una forma més manual i generar directament el codi HTML.

Per utilitzar mapes sensibles necessitem dues actuacions: declarar el mapa i assignar-lo a una imatge.

1) La declaració d'un mapa s'efectua amb la utilització de la marca <map> seguint la següent sintaxis:

```
<map name="nom_mapa">
<area shape=... coords=... href=... alt=...>
...
</map>
```

A la marca <map> només podem definir-hi el nom del mapa. És dins de cada marca <area> on podem definir més dades com:

- shape, que defineix la forma de la zona, que pot ser rectangular (rect), circular (circ) o poligonal (poly)
- coords, que defineix les coordenades de la zona, seguint la següent normativa:
 - Si es tracta d'un quadrat, cal indicar "x1,y1,x2,y2" on (x1,y1) són les coordenades de l'extrem superior esquerra i (x2,y2) les de l'extrem inferior dret.
 - Si es tracta d'un cercle, cal indicar "x,y,radi" on (x,y) són les coordenades del centre

Detecció de coordenades en una imatge?

La manera més simple d'aconseguir les coordenades d'un punt en una imatge és utilitzant qualsevol editor d'imatges (Gimp, Paint,...)

- Si es tracta d'un polígon, cal indicar "x1,y1,x2,y2,x3,y3" definint cada parella (x,y) un vèrtex del polígon. La darrera parella s'unirà amb la primera per a tancar el polígon.
- href, que defineix l'enllaç cap el destí on anirà l'usuari si pulsa sobre la zona
- nohref, per indicar que la zona no té definit cap enllaç
- alt, per incorporar el text que es presentarà en lloc de la imatge en navegadors no visuals per accedir a l'enllaç

2) L'assignació d'un mapa a una imatge s'efectua amb l'atribut usemap dins la marca que declara la imatge seguint la següent sintaxis:

El nom del mapa sempre ha d'anar precedit del símbol #.

Exemplifiquem-ho en la següent pàgina web amb la corresponent visualització a la figura 48. Suposem, per simplificar, que la imatge *navegadors.jpg* resideix en la mateixa ubicació que la pàgina web. Les seves mides són 80x42.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de mapes sensibles</title>
  <style type="text/css">
    body { text-align: center;
  </style>
</head>
<body>
<map name="navegadors">
  <area shape="rect" coords="0,0,40,42"</pre>
   href="http://www.mozilla-europe.org/es/products/firefox/"
    alt="Pà gina principal de Mozilla Firefox">
  <area shape="rect" coords="41,0,80,42"
    href="http://www.microsoft.com/spain/windows/products/winfamily/ie/default.mspx"
    alt="Pàgina principal d'Internet Explorer">
</map>
Premi damunt la imatge per accedir a les webs oficials dels corresponents navegadors<br/>
<img style="width: 80px; height: 42px;" usemap="#navegadors' alt="" src="navegadors.jpg" border="0">
</bodv>
</html>
```

Tinguis en compte que els enllaços aquí indicats a les pàgines principals dels navegadors són vàlids en el moment en que s'ha redactat aquest material i que pot haver quedat obsolet en el moment en que ho esteu llegint.



A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina21.html* amb el codi de l'exemple i la imatge *navegadors.jpg*.

2.2.2. Incorporació de so via enllaços

La majoria de navegadors actuals incorporen la capacitat de reproduir so (sempre i quan l'ordinador tingui el maquinari imprescindible). L'únic problema és que els arxius acostumen a ser grans i, com que no és imprescindible, pocs dissenyadors inclouen so a les seves pàgines.

Per altra banda també pot succeir que un navegador necessiti incorporar algun component per a poder reproduir l'arxiu de so.

W3C recomana que els navegadors incorporin mecanismes d'alerta per quan un recurs no està disponible o no és identificable. Les recomanacions són:

- Si el navegador no pot localitzar un recurs vinculat, hauria d'alertar a l'usuari.
- Si el navegador no pot identificar el tipus de recurs vinculat, hauria d'intentar processar-lo de totes maneres i hauria d'alerta a l'usuari permetent-li una intervenció per identificar el tipus de document.

Podem trobar-nos en dues situacions per a incorporar so dins un document:

a) Visualització en alguna ubicació de la pàgina d'alguna marca que provoqui la reproducció d'un arxiu de so a sol·licitud de l'usuari.

Aconseguir això és molt fàcil, doncs només cal utilitzar les possibilitats de definició d'enllaços cap a altres documents que faciliten la marca <a> i la definició de mapes sensibles. Haurem d'assignar el document de so com a destí de l'enllaç i la seva reproducció es produirà quan l'usuari activi l'enllaç.

La figura 49 mostra l'efecte d'un enllaç d'aquest tipus cap a un arxiu de so de tipus *MP3* en el navegador *Mozilla Firefox* sota *Windows*. En aquest

cas, com que *Mozilla Firefox* no té enregistrada la preferència de programa per a obrir aquest arxiu, ens presenta diferents possibilitats: *obre amb* que provocarà la reproducció de l'arxiu de so o *desa al disc*.



El codi HTML corresponent a aquesta pàgina és:

Cal observar, però, que el reproductor emprat en activar l'enllaç depèn de la configuració de l'acció *obertura* assignada a cada tipus d'arxiu en el corresponent navegador. El navegador *Internet Explorer* utilitza les assignacions definides en l'explorador de Windows (menú *Eines*|*Opcions de carpeta*|*Tipus de fitxer*), mentre que el navegador *Mozilla Firefox* utilitza les assignacions definides en la seva pròpia configuració (menú *Opcions*|*Contingut*|*Tipus de fitxer*). La taula 11 mostra diverses combinacions comprovades i algun efecte col·lateral no esperat segons la configuració. A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina22.html* amb el codi de l'exemple i l'arxiu de so *cn00147.mp3*.

Taula 11. Possibilitats de reproducció en activar un enllaç cap un arxiu de so.

| Sistema operatiu i navegadors | Reproductor assignat als fitxers .mp3 (1) | Com es produeix la reproducció |
|-------------------------------------|---|--|
| Windows XP | Windows Media | Es posa en marxa el reproductor, en una aplicació |
| Internet Explorer | Player 11 | Independent, reproduint i arxiu de so. La pagina web |
| 7 Real Player 10.5 | | |
| Mozilla Firefox 2 | QuickTime Player | En <i>Mozilla Firetox</i> , però, si l'assignació de tipus de fitxer correspon al connector (no l'aplicació!!!) de |

| Sistema operatiu i navegadors | Reproductor assignat als fitxers .mp3 (1) | Com es produeix la reproducció |
|-------------------------------------|---|---|
| | 7.1.5 | <i>QuickTime Player</i> (fet que succeeix amb la instal·lació normal de <i>QuickTime Player</i>), l'activació de l'enllaç provoca l'aparició del connector en una nova pàgina web substituint la pàgina actual, provocant un resultat no desitjat. |
| Linux Red Hat Mozilla Firefox 2 | Real Player 10.0.8 (gold) XMMS 1.2.10 | Es posa en marxa el reproductor, en una aplicació independent, reproduint l'arxiu de so. La pàgina web continua sent visible en el navegador. |

(1) Pe navegador *Internet Explorer* es tracta de l'assignació efectuada en l'explorador de Windows, mentre que pel navegador *Mozilla Firefox* es tracta de l'assignació efectuada en la seva pròpia configuració.

b) Reproducció automàtica en la càrrega de la pàgina, amb possibilitat de definir un reproducció cíclica de manera que no finalitzi mai.

La solució en aquest cas no és estàndard. Alguns navegadors han incorporat marques pròpies que faciliten aquesta possibilitat, però *W3C* no ho ha incorporat en l'especificació estàndard d'HTML.

Així, *Internet Explorer* incorpora des de la seva versió 2.0 la possibilitat d'incorporar fons sonors utilitzant la marca <bgsound> amb la següent sintaxis:

<bgsound src="nom_arxiu" loop="infinite">

L'atribut loop és optatiu i només serveix per a indicar que la reproducció sigui cíclica i no finalitzi mai. Vegem-ne un exemple:

Aquest document és reproduït de la forma esperada en el navegador *Internet* Explorer, mentre que el so és ignorat per altres navegadors com *Mozilla Firefox*, el qual no incorpora cap marca especial per aconseguir aquest efecte.

W3C va incorporar en la versió 4.0 d'HTML la marca <object> per a ser capaç d'incrustar en els navegadors qualsevol tipus de fitxer amb un seguit de paràmetres per a controlar-ne el seu comportament. Així doncs, la proposta seria utilitzar aquesta marca per a incloure també els arxius de so de qualsevol tipus en les pàgines web. A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina23.html* amb el codi de l'exemple i l'arxiu de so *cn00147.mp3*. Una altra solució possible és utilitzar codi *JavaScript*, però això ja és un altre tema...

2.3. Objectes genèrics

Les versions d'HTML anteriors a l'especificació 4.0, permetien als autores incloure imatges (per medi de la marca) i aplicacions (per medi de la marca <applet>). Aquests dos elements tenien limitacions:

- No serveixen per resoldre el problema més general de com incloure tipus de medis nous i futurs.
- La marca <applet> només funciona amb aplicacions basades en Java. De fet aquesta marca està desaprovada en favor de la nova marca <object>.
- Plantegen problemes d'accessibilitat.

Per solucionar aquests problemes, l'especificació 4.0 d'HTML introdueix la nova marca <object>.

2.3.1. Com utilitzar la marca <object>

La marca <object> neix per oferir una solució universal per a la inclusió de qualsevol tipus de fitxers en els documents HTML de manera que els navegadors rebin la informació necessària per a poder tractar l'objecte (fitxer incrustat en el document).

La nova marca assumeix algunes tasques realitzades per marques ja existents (per a imatges, <applet> per aplicacions, <iframe> per a altres documents HTML) i haurem d'acostumar-nos a utilitzar-la malgrat que puguem continuar utilitzant els mecanismes existents.

Com a regla general, la marca <object> servirà per a definir un objecte o component extern al navegador, el qual serà l'encarregat de reproduir (so, animació, vídeo,...) l'objecte. Per aconseguir una reproducció correcta, la marca <object> ha de permetre declarar l'objecte, la seva ubicació i el seu tipus i unes marques optatives especials <params> permetran acabar de definir els valors que pugui necessitar.

Així, podem introduir que la sintaxis a emprar serà similar a:

```
<object atribut1="valor1" atribut2="valor2" ... atributN="valorN">
   <param name="nom" value="valor">
   <param name="nom" value="valor">
   ...
</object>
```

Un atribut bàsic en la declaració d'un objecte és l'anomenat type que permet indicar al navegador el tipus de l'objecte que ha de carregar. El navegador hauria d'utilitzar aquesta informació per a esbrinar si disposa d'alguna aplicació adequada per a reproduir l'objecte, avortant-ne la càrrega si no en disposa.

De tot això se'n dedueix que hi ha dos punts importants per a que un objecte pugui ser reproduït correctament:

1) El tipus de l'objecte ha d'estar ben identificat en el document HTML.

Per aconseguir-ho cal utilitzar els tipus MIME, que sense saber-ho hem estat utilitzant fins ara en les capçaleres dels documents que *Nvu* ens generava (content="text/html").

2) El navegador ha de tenir instal·lada alguna aplicació adequada per a la reproducció de l'objecte.

Les aplicacions que possibiliten aquesta funcionalitat s'anomenen connectors (*plugins*).

La taula 12 mostra una llista dels atributs més utilitzats en la marca <object> i la taula 13 en fa el mateix pels paràmetres més usuals. En ambdues taules s'ha incorporat dues columnes per a comentar-ne el comportament en els navegadors *Mozilla Firefox* i *Internet Explorer*.

| Atribut | Tipus de valor i significat | Mozilla Firefox 2.0 | Internet Explorer 11 |
|-----------------|--|---|--|
| data | Nom de l'arxiu a reproduir amb la seva ubicació, que pot ser relativa a la pàgina o absoluta | Obligatori si no es posa el paràmetre scr | No en fa cas i en el seu lloc és obligatori utilitzar el paràmetre src |
| type | Cadena corresponent al mime-type adequat a l'objecte | Obligatori | Obligatori |
| width / height | Amplada i alçada, en pixels, del control | Obligatori, del contrari no es visualitza. Cal indicar la grandària adequada en funció de les parts de l'objecte que es desitgi visualitzar (controls, barra d'estat, pantalla,) | Optatiu. Si no s'indica, el navegador decideix la grandària en funció de les parts de l'objecte activades (controls, barra d'estat, pantalla,). |
| align | Idèntic al de l'etiqueta que permet alinear l'objecte respecte el text que l'envolta | Optatiu | Optatiu |
| hspace / vspace | Separació horitzontal i vertical, en pixels, entre l'objecte i el text que l'envolta | Optatiu | Optatiu |
| standby | Missatge que apareix en pantalla mentre l'objecte es carrega (depenent del seu pes i de la velocitat de connexió pot ser interessant emprar aquest atribut) | Optatiu | Optatiu |
| border | Valor numèric per indicar si l'objecte ha d'anar acompanyat d'una vora. Per defecte el seu valor és zero. | No en fa cas | Optatiu |

Taula 12. Llista dels atributs més utilitzats en la marca <object>

| Paràmetre Valor | | Significat | Valor per defecte | Mozilla Firefox 2.0 | Internet Explorer 11 |
|-----------------|-------------------------------------|--|-------------------|--|-------------------------|
| src | Nom de l'arxiu a pot ser relativa a | a reproduir amb la seva ubicació, que a la pàgina o absoluta | | Obligatori si no es posa l'atribut data | Obligatori |
| showcontrols | true/false | Indica si l'objecte ha de visualitzar els controls de reproducció | true | No en fa cas i sempre són visibles | Optatiu |
| autostart | true/false | Indica si l'arxiu ha de començar a reproduir-se per si sol en carregar-se la pàgina o si pel contrari serà necessària l'actuació de l'usuari per a que comenci la reproducció (fet que només podrà dur a terme si els controls estan visibles) | true | No en fa cas i sempre comença la reproducció | Optatiu |
| showdisplay | true/false | Indica si l'objecte ha de mostrar la pantalla de visualització (caldrà activar- ho si es tracta d'informació visual) | false | No en fa cas i mai el mostra | Optatiu |
| showstatusbar | true/false | Indica si l'objecte ha de mostrar la barra d'estat | false | No en fa cas i mai el mostra | Optatiu |

| Taula 1 | 13. Llista | dels | paràmetres | més | utilitzats | en l | la | marca | <object></object> |
|---------|------------|------|------------|-----|------------|------|----|-------|-------------------|
|---------|------------|------|------------|-----|------------|------|----|-------|-------------------|

2.3.2. Tipus MIME

MIME és l'acrònim anglès de *Multipurpose Internet Mail Extensions* (Extensions Multipropòsit de Correu d'Internet) i es tracta d'un estàndard que especifica com un programa (inicialment de correu o navegador web) ha de transferir arxius multimèdia (vídeo, so i, per extensió, qualsevol arxiu que no estigui codificat en *US-ASCII*. Abans de l'existència dels tipus MIME, qualsevol arxiu que no es limités a text ASCII havia de ser codificat a *US-ASCII*.

L'estàndard MIME adjunta un arxiu de capçalera a cada arxiu, especificant el tipus i el subtipus del contingut de l'arxiu principal. Gràcies a aquesta informació, tant el servidor com el navegador poden gestionar i presentar correctament les dades.

En la utilització diària d'internet, estem beneficiant-nos (i a vegades sofrint) els tipus MIME. Cada vegada que sol·licitem una pàgina d'internet, s'obre un diàleg entre el nostre navegador i el servidor que proporciona la pàgina. El nostre navegador demana la pàgina i el servidor, abans d'enviar-la, confirma que existeix i el tipus de dades que conté. Això darrer es fa via el tipus MIME que correspongui.

La gestió de tipus MIME a la web apareix en tres punts ben diferenciats:

1) En el servidor, que ha de ser capaç de gestionar diversos tipus MIME i tenir-los activats.

2) En la pàgina web, on l'autor constantment fa referència a tipus MIME. Així, els enllaços cap a arxius externs (fulls d'estil, scripts de *JavaScript*, objectes incrustats,...) haurien d'especificar el tipus de l'arxiu enllaçat, com es veu en els següents exemples.

- Enllaços a full d'estil

<link rel="stylesheet" type="text/css" href="estils.css">

- Crida a scripts JavaScript

<script language="JavaScript" type="text/javascript" scr=... >

– Definició d'objectes

<object type="audio/mp3" ...>

3) En el navegador del client, el qual ha d'estar capacitat per interpretar els tipus MIME que el servidor li envia i, fins i tot, ha de poder informar al servidor dels tipus MIME que pot acceptar.

Per a permetre aquesta funcionalitat, el navegador ha de tenir instal·lades les aplicacions adequades als diferents tipus MIME que interessi gestionar. Són els connectors (*pluggins*).

L'organisme que s'encarrega de registrar els tipus MIME és *IANA*, acrònim anglès d'*Internet Assigned Numbers Authority* (Agència d'Assignació de Números d'Internet), la qual depèn de l'ICANN, acrònim anglès de *Internet Corporation for Assigned Names and Numbers* (Corporació d'Internet per l'Assignació de Noms i Números).

Els tipus MIME s'agrupen, actualment, en vuit categories i cada tipus MIME s'identifica per el nom compost: categoria/tipus. En ocasions es parla de tipus/subtipus. Les vuit categories són: application, audio, image, message, model, multipart, text i video. Cadascuna d'elles està formada per un conjunt més o menys gran de tipus MIME i cadascun dels tipus acostuma a portar associat una o vàries extensions d'arxius.

La taula 14 recull alguns dels tipus existents amb algunes de les extensions d'arxius normalment associades.

| Tipus MIME | Extensions d'arxius | Tipus MIME | Extensions d'arxius |
|------------|---------------------|------------------------|---------------------|
| image/bmp | .bmp | video/x-msvideo | .avi |
| image/gif | .gif | application/pdf | .pdf |
| image/jpeg | .jpg, .jpeg, .jpe | application/postscript | .ai, .eps, .ps |
| image/tiff | .tif, .tiff | application/rtf | .rtf |

Taula 14. Recull de tipus MIME i extensions associades d'arxius

| Tipus MIME | Extensions d'arxius |
|-----------------|---------------------|
| audio/x-wav | .wav |
| audio/x-midi | .mid |
| text/plain | .txt |
| text/richtext | .rtf, .rtx |
| video/mpeg | .mpeg, .mpg, .mpe |
| video/quicktime | .qt, .mov |

| Tipus MIME | Extensions d'arxius |
|-------------------------|---------------------|
| application/gzip | .gz |
| application/x-tar | .tar |
| application/zip | .zip |
| application/x-java-vm | .class |
| Application/x-java-arch | .jar |
| Application/msword | .doc, .dot |

2.3.3. Connectors

Per a poder reproduir arxius definits dins les pàgines web com a objectes genèrics, els navegadors necessiten tenir instal·lat el connector adequat.

Els connectors (*plugins*) són aplicacions informàtiques que interactuen amb una altra aplicació per afegir-li una funció o utilitat específica.

És a dir, si en un document web s'hi ha incrustat un objecte (<object>) que fa referència a un document d'un cert tipus MIME, el navegador que l'hagi de reproduir ha de disposar d'un connector adequat per aquell tipus MIME.

El fet que el sistema operatiu disposi d'una aplicació que pugui reproduir un determinat arxiu (corresponent a un tipus MIME) no és garantia de que els navegadors instal·lats en aquell sistema tinguin instal·lat el connector corresponent.

Els navegadors, en rebre pàgines web que contenen tipus MIME no suportats, haurien de permetre a l'usuari (recomanació *W3C*) la possibilitat de procedir a la instal·lació d'una manera senzilla. En aquest punt encara hi ha molt camí a recórrer.

Ben segur que us ha aparegut el dubte: quins connectors tinc instal·lats?

En el navegador *Mozilla Firefox* és molt senzill obtenir aquesta informació: cal escriure about:plugins a la barra de navegació i prèmer <return>. El navegador ens mostra un document HTML amb la informació corresponent a tots els connectors instal·lats, com es veu a la figura 50. La informació que facilita és molt interessant: connectors instal·lats (versió i nom de l'arxiu corresponent) i, per a cada connector, la relació de tipus MIME pels que pot donar servei. Aquesta darrera informació ens pot ser molt útil a l'hora d'incrustar objectes a les pàgines web, per tal de saber el tipus MIME que hauríem d'assignar a un arxiu de manera que els habituals connectors instal·lats en els navegadors els puguin reproduir (i no obligar a l'usuari a instal·lar nous connectors... suposant que els localitzés...).

| als connectors - Mozilla | Firefox | | | | |
|---|---|---|-------------|--------------------|--|
| dita ⊻isualitza <u>H</u> istorial / | Adreces d'interès Eines Ajuda | | | | |
| 🕨 - 🧭 💿 🏠 🕻 | about:plugins | | | • 🕨 🖸 • | Google |
| l gratuito 📄 Personalizar vínc | culos 📋 Windows Media 📄 Windows | ; | | | |
| | Coni | nectors inst | al·lats | | |
| Podeu trobar més informació so L'ajuda per a instal·lar els conne | bbre els connectors del navegador a moz ectors està disponible a <u>plugindoc.mozde</u> | rila.org. ev.org. | | | |
| Real Jukehov NS | Plugin | | | | |
| RealJukebox 113 | Plagin | | | | |
| Nom del fitxer: npriplu; RealJukebox Netscape | g.dl Plugin | | | | |
| Tipus MIME | | Descripció | | Sufixos | Habilitat |
| ripus i ta ite | | | | | |
| | RealJukebox NS Plugin File | | none | | Sí |
| none RealPlayer(tm) G Nom del fitxer: nppl326 RealPlayer(tm) LiveCor | RealJukebox NS Plugin File | ed Plug-In (32-bit) | none | | Si |
| none RealPlayer(tm) G Nom del fitxer: nppl326 RealPlayer(tm) LiveCon Tipus MIME | RealJukebox NS Plugin File | ed Plug-In (32-bit) Descripció | none | Sufixos | Si Habilitat |
| none RealPlayer(tm) C RealPlayer(tm) c RealPlayer(tm) LiveCor Tipus MIME audio/x-pn-realaudio-plugin | RealJukebox NS Plugin File 52 LiveConnect-Enable 50.dll nect-Enabled Plug-In RealPlayer(tm) as Plug-in | ed Plug-In (32-bit) Descripció | rpm | Sufixos | Sí Habilitat Sí |
| none RealPlayer(tm) C Non del fixer rop[32 RealPlayer(m) LiveCor Tipus MIME audio/ripmrealaudio-plugn RealPlayer Versia | RealJukebox NS Plugin File 52 LiveConnect-Enable 50.dl med:Enabled Plug-In RealPlayer(tm) as Plug-in on Plugin | ed Plug-In (32-bit) Descripció | none rpm | Sufixos | Si Həbilitət Si |
| none RealPlayer(tm) C Nom del fikrer: npp132 RealPlayer(tm) LiveCor Tipus MiMi audio):con-realaudio-plugin RealPlayer Versio | RealDukebox NS Plugin File 52 LiveConnect-Enable 30.dll nect-Enabled Plug-In RealPlayer(tm) as Plug-in on Plugin | ed Plug-In (32-bit) Descripció | rpm | Sufixos | Si Habilitat Si |
| RealPlayer(tm) CC Nom del ficxer: nopl52c RealPlayer(tm) LvcCi Tipus MME audo/scpn-realaudo-plugin RealPlayer Versio Nom del ficxer: ropripil. 6.0.12.17 | RealDukebox NS Plugin File 52 LiveConnect-Enable 50.dl nect-Enabled Plug-In RealPleyer(tm) as Plug-in on Plugin ug.dl | ed Plug-In (32-bit) Descripció | rpm | Sufixos | Si Habilitat Si |
| RealPlayer(tm) C Nom del fitxer: npl526 RealPayer(tm) LvcCor Tipus MHE audio/-pn-realaudio-plugin RealPlayer Versio Nom del fitxer: npripia 6.0.12.1799 Tipus HIME | RealDukebox NS Plugin File 62 LiveConnect-Enable 50.dll nect-Enabled Plug-in RealPlayer(tm) as Plug-in on Plugin ug.dl | ed Plug-In (32-bit) Descripció Descripció | rpm | Sufixos | Sî Habilkat Sî Habilkat |
| RealPlayer(tm) C Nom del fitxer: nopl326 RealPlayer(tm) LiveCor Tipus MHE audio/:-pn-realaudio-plugin RealPlayer Versio Nom del fitxer: nopripilo 6.0.12.1799 Tipus MHE | RealDukebox NS Plugin File 62 LiveConnect-Enable 50.dl nect-Enabled Plug-In RealPlayer(tm) as Plug-In on Plugin ug.dl | ed Plug-In (32-bit) Descripció | rpm | Sufixos | Sî Habilitat Sî Habilitat |
| RealPlayer(tm) C Nom del fitxer: nopl52c RealPlayer(tm) LvcC Tipus MME audo(x-pn-realaudo-plugin RealPlayer Versic Nom del fitxer: ropripik 6.0.12.179 Tipus MIME application/ind.m-realplayer.j | RealDukebox NS Plugin File | ed Plug-In (32-bit) Descripció | rpm rp1 | Sufixos Sufixos | Si Habilitat Si Habilitat Si |
| RealPlayer(tm) C Nom del fitxer: npl526 RealPlayer(tm) LiveCor Tipus MHE audio/-pn-realaudio-plugin RealPlayer Versio Nom del fitxer: npripik 6.0.12.1759 Tipus MHE application/ind.m-realplayer-j | RealDukebox NS Plugin File 62 LiveConnect-Enable 60.dil nect-Enabled Plug-in RealPlayer(tm) as Plug-in on Plugin ug.dil avascript RealPlayer Version Plugin | ed Plug-In (32-bit) Descripció | rpm rpj | Sufixos Sufixos | Sî Həblikət Sî Həblikət Sî |
| RealPlayer(tm) C Nom del fitxer: ropt522 RealPlayer(tm) LVCC Tipus MIME audio/cm-realaudio-plugin RealPlayer Versie Nom del fitxer: roptpla 6.0.12.1799 Tipus MIME application/vnd.m-realplayer-j | RealDukebox NS Plugin File 52 LiveConnect-Enable 50.dll rect-Enabled Plug-In RealPlayer(tm) as Plug-in on Plugin ug.dl levascript RealPlayer Version Plugin Plug-in | ed Plug-In (32-bit) Descripció Descripció | rpm rp] | Sufixos Sufixos | Sî Habiltat Sî Habiltat Sî |
| RealPlayer(tm) C Nom del fiber: repl22 RealPayer(tm) Drec repus Mint audio/cpn-realaudio-plugn RealPlayer Versio Nom del fiber: republic application/ind.m-realplayer; Mozilla Default P Nom del fiber: republic | RealDukebox NS Plugin File | ed Plug-In (32-bit) Descripció | rpm rpm | Sufixos Sufixos | Si Habilitat Si Habilitat Si |

2.3.4. Exemples d'incorporació d'objectes genèrics

Anem a veure alguns exemples d'incrustació d'objectes en les pàgines web. Per a decidir quin tipus MIME assignar en cada cas, ens hem basat en la informació que *Mozilla Firefox* ens facilita sobre els connectors instal·lats. La taula 15 recull les configuracions on s'ha comprovat la correcta execució dels exemples.

| Sistema operatiu | Navegador | Programari reproductor | Connectors |
|-----------------------|----------------------|--|-------------------------------------|
| | Internet | Windows Media Player 11 | Windows Media Player |
| Windows XP M Fi | Mozilla Firefox 2 | Real Player 10.5 QuickTime Player 7.1.5 | QuickTime Plug-in |
| | Mozilla | Real Player 10.0.8 (gold) | Plugger 5.1.3 |
| Linux | Firefox 2 | XMMS 1.2.10 MPlayer 1.0rc1 | Helix DNA Plugin Shockwave Flash |

Taula 15. Programari instal·lat que possibilita la correcta execució dels exemples

• Exemple d'incorporació d'un arxiu de so .mp3 vinculant-lo al tipus MIME "application/x-mplayer2".

A la secció "Recursos" del web d'aquest crèdit podeu trobar els arxius d'on s'ha extret els exemples.

• Exemple d'incorporació d'un arxiu de so *.wav* vinculant-lo al tipus MIME "application/x-mplayer2".

• Exemple d'incorporació de diferents versions d'un mateix arxiu per facilitar la seva reproducció des de diverses configuracions.

L'exemple mostra la incrustació d'un arxiu de so *.ogg* per la que hi pot haver navegadors que no tinguin un connector instal·lat. Aquesta és la situació de la nostra configuració, doncs en el sistema Windows no tenim cap connector que pugui reproduir aquest tipus d'arxiu. En canvi, en el sistema Linux disposem del meravellós *Plugger 5* que s'ho empassa tot...

HTML ens permet incrustar un objecte dins un altre, de manera que el navegador que ha de reproduir l'objecte ho intenta amb el més extern i si no ho aconsegueix ho intenta amb el que aquest tingui incrustat i així successivament. En aconseguir una reproducció es desestimen els objectes que hi pogués haver incrustats.

• Exemple d'incorporació d'un arxiu *Flash .swf* vinculant-lo al tipus MIME "application/x-shockwave-flash".

```
<!-- Fragment de l'arxiu pagina27.html -->
<object data="calculator.swf" type="application/x-shockwave-flash" height="400" width="533">
<param name="src" value="calculator.swf">
</object>
```

Flash és una eina que permet donar a les pàgines web un aspecte lúdic, una presentació potent i sorprenent. Amb Flash es pot realitzar un treball animat que posteriorment es guarda en un arxiu d'extensió *.swf* i que es pot inserir en les pàgines web. A Internet hi ha molts tutorials sobre *Flash*.

L'editor especialitzat en pàgines web *Dreamweaver* proporciona funcionalitats per inserir arxius *Flash* en les pàgines web, generant botons que permeten posar en marxa i aturar l'animació. L'editor *Nvu* no proporciona aquesta funcionalitat.

• Exemple d'incorporació d'un arxiu de video .*avi* vinculant-lo al tipus MIME "application/x-msplayer2".

2.4. Formularis

Un formulari HTML és una secció d'un document HTML que conté codi HTML i elements especials anomenats controls: caselles de selecció (*check boxes*), botons d'opció (*radio buttons*), menús, etiquetes (labels),... i que s'utilitza per a recollir informació dels usuaris de la pàgina.

Els usuaris normalment completen un formulari modificant els seus controls (introduint text, seleccionant objectes d'un menú, etc.) abans d'enviar el formulari a un agent d'usuari per a que el processi (per exemple, a un servidor web, a un servidor de correu,...).

Com a exemple inicial força entenedor, vegem el següent codi HTML:

</body> </html>

En aquest formulari no s'ha tingut en compte cap element de disseny i únicament s'ha volgut presentar un primer formulari que serveixi de base per a explicar alguns dels components que hi podem utilitzar en el seu disseny. La figura 51 en mostra la visualització en un navegador. A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina29.html* amb el codi de l'exemple.

Figura 51. Visualització d'un formulari en un navegador



L'usuari del formulari de la figura 51 pot omplir els camps *nom*, *cognoms* i *email* així com marcar el *sexe* que correspongui i, amb els botons, pot decidir si envia la informació o si pel contrari neteja els valors del formulari i comença de nou. En aquest cas, en prémer el botó *Enviar* es produirà l'enviament de la informació introduïda en el formulari via un correu electrònic degut a que a la marca <form> hi ha establert l'atribut action com a mailto. Aquest funcionament és factible sempre i quan en la sessió on s'està executant el navegador, hi hagi configurat un compte de correu electrònic, el qual serà utilitzat per enviar el correu cap al destinatari indicat al costat de mailto.

El funcionament dels navegadors en prémer el botó *Enviar* no és estàndard. En el cas de *Mozilla Firefox*, aquest navegador obre el programa de correu electrònic configurat a la sessió amb els continguts corresponents al formulari. La figura 52 en mostra un exemple on es pot observar com es construeix el missatge a partir dels camps input del formulari que tenien l'atribut name definit. Observeu que l'usuari pot modificar el contingut del missatge i la direcció d'enviament i l'assumpte... Vaja, que aquest funcionament no és desitjable. En canvi, *Internet Explorer* construeix igualment el missatge i l'envia directament sense obrir el programa de correu electrònic.

En l'exemple precedent hem pogut apreciar la utilització d'alguns elements desconeguts fins el moment: marca form amb atributs action, method i enctype i marca input amb atributs name, value i type. El llenguatge HTML dona moltes més possibilitats en el disseny de formularis.

| Figura 52. Funcionament de I | Mozilla Firefox en prémer el botó Enviar. | | | | | |
|--|--|--|--|--|--|--|
| 🕲 Exemple de formulari de correu - | Mozilla Firefox 📃 🗖 🗙 | | | | | |
| <u>F</u> itxer <u>E</u> dita <u>V</u> isualitza <u>H</u> istorial Ad <u>r</u> e | xes d'interès Ei <u>n</u> es Ajuda | | | | | |
| 🦛 • 🔿 • 🥑 🛞 🏠 🕒 Fil | le:///E:/pagina29.html 🔻 🕨 💽 🗸 Google 🔍 | | | | | |
| 🕒 Hotmail gratuito 📄 Personalizar vínculos 👔 Enviament de formulari de Firefox 🛛 🗐 🔀 | | | | | | |
| Nam: Papa | j Archivo Edición Ver Insertar Formato He 꿭 雅 | | | | | |
| Cognom: Gotera e- <u>Mail</u> : pgotera@gmail.com | Enviar Cortar Copiar | | | | | |
| Home Home O Home Hom Home Hom | De: redactor_03_06@gmail.com (Redactor UD3 🗸 | | | | | |
| ⊙ Dona [Enviar] Netejar | Image: Para: usuari@xtec.cat Image: Para: usuari@xtec.cat Image: Para: Image: Para: Image: Para: usuari@xtec.cat Image: Para: Image: Para: Image: Para: Image: Para: | | | | | |
| | | | | | | |
| | nom=Pepe cognom=Gotera email=pgotera@gmail.com sexe=Home | | | | | |
| | | | | | | |
| Fet | | | | | | |

2.4.1. Propietats d'un formulari

L'editor Nvu facilita ajuda per la creació de formularis i, ja que tenim ajut, l'anem a utilitzar enlloc d'escriure tot el formulari directament amb l'HTML.

Per a definir un formulari des de Nvu, podem utilitzar el botó Formulari de la barra d'eines de composició o el menú Insereix Formulari. En qualsevol cas apareix la finestra modal de la figura 53 on definirem les propietats inicials del formulari:

• Nom del formulari, el qual genera l'atribut name dins la marca <form>. De moment no ens té cap utilitat.

| Figura 53. Pantalla | Nvu per definició de formulari | | | |
|--|--------------------------------|--|--|--|
| Propietats del formulari | | | | |
| Paràmetres Nom del formulari: URL d' <u>A</u> cció: <u>M</u> ètode: | | | | |
| Codifi <u>c</u> ació: Marc <u>d</u> e destinació: | · | | | |
| Edició avançada | | | | |

• URL d'Acció, que cal emplenar amb l'agent d'usuari que hagi de rebre les dades introduïdes en el formulari per a processar-les.

Si el que volem és simplement enviar les dades a algú via correu electrònic, utilitzem la sintaxi mailto:direcció_correu que genera el següent codi HTML.

<form action="mailto:direcció_correu" ...>

En cas d'enviar les dades per correu electrònic cal obligatòriament utilitzar l'atribut enctype="text/plain" per a que l'agent d'usuari receptor del correu pugui entendre el contingut.

En canvi, si volem que les dades introduïdes per l'usuari siguin processades, haurem d'indicar la direcció de l'arxiu que contingui el programa que hagi de processar les dades. Utilitzarem, en aquest cas, la sintaxi direcció_arxiu que genera el següent codi HTML:

<form action="direcció_arxiu" ...>

Respecte la utilització de programes per a processar les dades d'un formulari cal comentar que tenim diverses possibilitats:

- Podem utilitzar programes existents en el nostre servidor web que permetin emmagatzemar i processar les dades en forma d'arxius u altres formats (molts servidors webs proporcionen aquestes facilitats). L'administrador del servidor web ens informarà al respecte.
- Podem utilitzar programes similars al cas anterior facilitats per servidors de tercers (alguns de pagament i altres gratuits)
- Podem desenvolupar els nostres propis programes via llenguatges com ASP o PHP.
- *Mètode*, que pot prendre únicament dos valors: get i post. Aquest atribut s'encarrega d'especificar la forma en que les dades són enviades. El valor per defecte és get.
 - El valor get provoca que el conjunt de dades del formulari (parelles nom=valor concatenades amb símbols &) s'afegeixi al valor indicat en l'atribut action amb un símbol ? com a separador, de manera que el valor de l'atribut action queda modificat pel conjunt de valors introduïts per l'usuari.
 - El valor post provoca que el conjunt de dades del formulari s'inclogui en el cos del formulari i s'envia a l'agent que el processarà.

Com que el valor get provoca la modificació de l'atribut action, no té sentit utilitzar-lo quan es pretén enviar les dades del formulari via

correu electrònic. Per això, sempre que l'atribut action conté un mailto:, el valor de l'atribut method és post.

Utilitzem el darrer exemple canviant el valor dels atributs action (indicarem una direcció web fictícia) i l'atribut method pel valor get. És a dir, la capçalera del formulari ens queda com:

<form action="ficticia" method="get">

En prémer el botó *Enviar*, el navegador intenta contactar amb l'adreça:

ficticia?nom=Pepe&cognom=Gotera&email=pgotera%40gmail.com&sexe=Home

Observeu que totes les dades del formulari viatgen acompanyant la direcció on es troba el programa que ha de processar-les. Intuïu que si les dades no estan codificades això pot provocar un greu problema de seguretat? Per tant, es recomana utilitzar el valor post.

No sembla incoherent que el valor per defecte de method sigui get?

Segons *W3C*, el mètode get s'hauria d'utilitzar quan el formulari és idempotent (és a dir, quan no té efectes secundaris). Moltes recerques en bases de dades no tenen efectes secundaris visibles i constitueixen aplicacions ideals per al mètode get. Si el servei associat al processament d'un formulari causa efectes secundaris (per exemple, si el formulari modifica una base de dades o la subscripció a un servei), s'hauria d'utilitzar el mètode post.

Per últim comentar que el mètode get restringeix els valors del conjunt de dades del formulari a caràcters *ASCII*. Només el mètode post amb enctype="multipart/form-data" pot enviar dades no ASCII.

- *Codificació*, que permet diversos valors tipus MIME i que la majoria de les vegades escollirem text/plain. Aquest camp genera l'atribut enctype comentat abans.
- Marc de destinació, que permet indicar la ubicació on s'ha d'obrir l'agent d'usuari responsable del procés de les dades del formulari. Aquest camp genera l'atribut target (com en la definició d'enllaços) i els valors possibles són els ja coneguts _blank, _self, _parent i _top.

Una vegada emplenats tots els camps de la figura 53, premem el botó *Accepta* i *Nvu* ens presenta una zona puntejada que ens delimita l'espai del formulari. Aquesta zona anirà modificant la seva grandària en funció del contingut que anem donant al formulari. Dins d'aquesta zona és vàlid ficar-hi qualsevol contingut HTML ja conegut (taules, llistes, paràgrafs,...) però també els elements especials dels formularis (camps de text, botons d'opció, caselles de selecció, àrees de text, botons,...)

2.4.2. Camps de text

Per afegir un camp de text, podem utilitzar la fletxa avall de la dreta del botó *Formulari* de la barra d'eines de composició i escollim l'opció *Camps del formulari*. També hi podem arribar per *Insereix* | *Formulari*. En qualsevol cas apareix una finestra com la de la figura 54 que ens permet escollir el tipus de camp i haurà de ser *text*.

| Propietats de camp de formulari 🛛 🔀 |
|--|
| Ipus de camp Text |
| Paràmetres del camp |
| Nom del camp: Valor inicial: |
| Només lectura |
| ▼ Més propietats |
| Edició avançada |
| D'acord Cancel·la |

Figura 54. Pantalla Nvu que permet definir les propietats de camp text.

Observem que se'ns demana la introducció de tres possibles paràmetres:

- *Nom del camp*, que ha de ser únic en tot el formulari i que serà el que s'enviarà a l'agent d'usuari que hagi de processar el formulari, acompanyat del valor que l'usuari de la pàgina introdueixi.
- *Valor inicial*, que serveix per a que el navegador mostri d'entrada un valor en el camp, que l'usuari podrà, evidentment, modificar.
- *Només lectura*, que serveix per a marcar un camp de manera que l'usuari no en pugui modificar el seu contingut.

Una vegada premut el botó *Acceptar*, *Nvu* genera, en el lloc on estàvem ubicats, el codi HTML similar al següent (pot no ser igual si algun camp no s'emplena):

<input readonly="true/false" name="nom_camp" value="valor_inicial">

Per cert, no us pregunteu quin sentit té disposar d'un camp de només lectura? Per a nosaltres, de moment, no té cap sentit, però penseu que els formularis es poden gestionar dins programes els quals sí disposen de mecanismes per a modificar les propietats d'un camp (només lectura, per exemple) abans que el navegador visualitzi el formulari.

2.4.3. Botons d'opció

Els botons d'opció (*radio buttons*) són elements visuals que faciliten múltiples opcions a l'usuari permetent que només n'escolli una. Per a que funcioni cal definir un grup per a tot el conjunt de botons de les diferents opcions.

Des de *Nvu*, via l'opció ja coneguda *Camps de formulari* de la figura 54 escollim *Botó de Grup* com a tipus de camp. La pantalla de la figura 54 canvia per la de la figura 55.

| Propietats de ca | mp de formulari 🛛 🔀 |
|-----------------------|---------------------|
| | |
| Botó de grup | |
| Paràmetres del ca | amp |
| <u>N</u> om del grup: | |
| Valor del camp: | |
| 🔻 Més <u>p</u> ropie | etats |
| | Edició avançada |
| | D'acord Cancel·la |

Figura 55. Pantalla Nvu que permet definir les propietats de camp Botó de grup.

Observem que se'ns demana la introducció de tres possibles paràmetres:

- *Nom del grup*, que ha de ser únic en tot el formulari i que ha de ser el mateix per a tots els botons que formin el grup. Aquest nom serà el que s'enviarà a l'agent d'usuari que hagi de processar el formulari, acompanyat del valor assignat al botó activat.
- *Valor del camp*, que és el valor que s'enviarà a l'agent d'usuari (i no pas la possible etiqueta que acompanyi al botó). Els diferents botons d'un mateix grup han de tenir assignats valors diferents. Aquest valor no es visualitza en el navegador.
- *Inicialment seleccionat*, que serveix per a marcar un dels botons un dels botons del grup de manera camp de manera que l'usuari no en pugui modificar el seu contingut.

Una vegada premut el botó *Acceptar*, *Nvu* genera per a cada botó i en el lloc on s'ubiquin, el codi HTML similar al següent:

Els noms Etiqueta 1, Etiqueta 2 i Etiqueta 3 no s'han entrat per la pantalla de la figura 54, sinó que s'han escrit directament dins el formulari, a la dreta de cada botó d'opció. Per això, en el codi, apareixen a la dreta de la definició de cada botó. El valor que s'enviarà a l'agent d'usuari que hagi de processar les dades serà el contingut de l'atribut value corresponent al botó actiu.

2.4.4. Caselles de selecció

Les caselles de selecció (*checkbox*) són elements visuals que permeten a l'usuari donar el valor de cert/fals o si/no o activat/desactivat a un camp en concret.

Des de *Nvu*, via l'opció ja coneguda *Camps de formulari* de la figura 54 escollim *Quadre de verificació* com a tipus de camp. La pantalla de la figura 54 canvia per la de la figura 56.

Figura 56. Pantalla Nvu que permet definir les propietats de camp Quadre de verificació.

| Propietats de camp | de formulari 🛛 🔀 |
|------------------------------------|-----------------------|
| Tipus de camp | |
| Quadre de verificacio | |
| Paràmetres del camp | |
| Nom del camp: | |
| Valor del camp: | |
| | Establert inicialment |
| Més propietats | |
| | Edició avançada |
| D | acord Cancel·la |

No cal comentar els paràmetres doncs són similars als de les opicons *Camp de text* i *Botó de grup*.

Nvu genera per a cada casella de selecció un codi HTML similar a:

<input name="nom_camp" value="valor" type="checkbox">

En el conjunt de dades que s'enviarà, la parella nom_camp=valor hi apareixerà si la casella de selecció està activada. En cas de no estar activada, la parella no forma part del conjunt de dades del formulari.

2.4.5. Àrea de text

Les àrees de text estan destinades a recollir una gran quantitat de text que pugui introduir l'usuari. Les seves propietats permeten definir una amplada (columnes) i una alçada (files) que provocarà l'aparició d'una finestra amb aquestes dimensions per a que l'usuari hi pugui escriure. El contingut, però, del text que introdueixi l'usuari no està limitat a aquesta finestra, ja que si en necessita més, el navegador li ho permetrà i mostrarà una barra vertical de desplaçament a la part dreta de l'àrea.

Des de *Nvu*, cal emprar l'opció *Àrea de Text* que acompanya al botó *Formulari* de la barra d'eines de composició. Ens apareix una pantalla com la de la figura 57.

Figura 57. Pantalla Nvu que permet definir les propietats d'una Àrea de text.

| Propietats d'Àrea de text 🛛 🔛 |
|-------------------------------|
| Paràmetres Nom del camp: |
| _olumnes: ▼ Més propietats |
| Edició avançada |
| D'acord Cancel·la |

Els paràmetres a introduir són clars: nom del camp, files i columnes. El codi generat és similar a:

<textarea cols="20" rows="4" name="comentari"></textarea>

Si es desitja que l'àrea de text contingui un text inicial, es pot incloure en el codi HTML entre les marques <textarea> i </textarea>, com:

<textarea ...>Ompliu aquest espai</textarea>

Això també es pot aconseguir des de Nvu via el botó Mes propietats.

2.4.6. Botons

En tot formulari es necessita com a mínim un botó per a executar l'enviament de les dades. Així mateix és molt usual incloure un botó que permeti fer neteja de tots els camps per a tornar a la situació inicial.

Des de *Nvu*, cal emprar l'opció *Definir botó* que acompanya al botó *Formulari* de la barra d'eines de composició. Ens apareix una pantalla com la de la figura 58.

Figura 58. Pantalla Nvu que permet definir els botons.



En visualitzar les possibilitats del paràmetre *Tipus* hi trobem tres possibles valors: *Submit*, *Reset* i *Botó*. Els dos primers no estan traduïts a la llengua catalana i correspondrien als conceptes *Enviar* i *Netejar/Restablir*. El tercer valor permet crear botons que portin associades funcionalitats que cal programar amb un llenguatge de programació (*JavaScript*, per exemple).

Observem que no hi ha cap espai per introduir el text que ha de mostrar el botó. Per tant, si generem els botons i no hi posem text, quasi no es distingiran en el formulari. La forma lògica de procedir és, en primer lloc escriure en el formulari (visualització *normal*) el text del botó, seleccionar el text i escollir l'opció *Definir botó*.

Els paràmetres *nom* i *valor* pels botons *Submit* i *Reset* no tenen, en principi, cap importància, i els omplirem amb els valors *enviar* pel botó *Submit* i *restablir* pel botó *Reset*. El codi HTML generat és similar a:

<button value="enviar" name="enviar">Enviar</button><button value="restablir" name="restablir" type="reset">Netejar</button>

Observem que el botó *restablir* va acompanyat per l'atribut type amb valor reset mentre que el botó *enviar* no porta l'atribut type. HTML entén que si un botó no porta l'atribut type és un botó de tipus *submit*. Ara bé, és pot incorporar aquest atribut amb el corresponent valor submit.

En cas d'incloure en el formulari altres botons que no siguin *reset* ni *submit* (i als que caldrà assignar programes), s'utilitza el valor button per a l'atribut type. Així, doncs, podríem trobar-nos amb codi HTML similar a:

<button value="enviar" name="enviar" type="submit">Enviar</button><button value="restablir" name="restablir" type="reset">Netejar</button><button value="botoX" name="botoX" type="button">Botó X</button><button value="botoY" name="botoY" type="button">Botó Y</button><button value="botoY" name="botoY" type="button">Botó Y</button><button>

2.4.7. Llistes de selecció

En ocasions caldrà que l'usuari pugui escollir una opció d'un nombrós conjunt d'opcions i la utilització de botons d'opció no és una bona solució donat que necessitaríem un munt de botons. Això és el que succeeix, per exemple, si volem que l'usuari hagi d'escollir la seva comarca preferida i volem subministrar a l'usuari la totalitat de les comarques. Resoldre la situació amb botons d'opció provocaria utilitzar un munt d'espai per a la totalitat de les comarques (a botó per comarca). La possibilitat dels botons d'opció tampoc seria vàlida si l'usuari ha de poder efectuar una elecció múltiple, és a dir, més d'una de les opcions proposades.

Les llistes de selecció són un recurs visual que facilita múltiples opcions a l'usuari permetent que n'esculli una, vàries o cap.

Per a definir una llista de selecció des de *Nvu*, cal emprar l'opció *Llista de selecció* del botó *Formulari* de la barra d'eines de composició. Ens apareix la pantalla de la figura 59.

Figura 59. Pantalla Nvu que permet definir una llista de selecció.

| Selecció de propiet | ats de la llista | × |
|----------------------|--------------------|--------------------|
| Text | Valor | Seleccionat 🛤 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Llista de selecció — | | Afegeix opció |
| Nom de la llista: | | Afegeix grup |
| Alçada: | Colocató aválticla | Suprimeix |
| | Dechabilitat | A May amunh |
| Índex de tabulació: | | |
| | | Mou a <u>v</u> all |
| | | Edició avançada |
| | | |
| | | acord Cancel·la |

En primer lloc, en l'apartat *Llista de selecció* (figura 59), posarem un nom a la llista i en definirem l'alçada (número de files visibles). També hi definirem si la llista permet selecció múltiple.

| Figura 60. Exe | mple de creació d' | una Ilista de | selec | ció. |
|-----------------------------|--------------------------|-------------------------------|-------|------|
| Selecció de propie | etats de la Ilista | | × | |
| Text | Valor | Seleccionat | ₽. | |
| comarques Vegueria Barce | lona | | ^ | |
| Barcelonès | barcelones | | = | |
| | iental VallesOccidental | | | |
| Vegueria Camp | o de Ta | | | |
| Baix Camp | ; tarragones baixCamp | | ~ | |
| Opció | | Afegeix opció | | |
| Lext: | | Afegeix grup | | |
| | Seleccionat incialment | Suprimeix | | |
| | 📃 Deshabilitat | Mou amunt | | |
| | | Mou a <u>v</u> all | | |
| | | Edició avançada |] | |
| | D'a | cord Cancel | la | |

La figura 59 mostra també dues opcions *Afegeix opció* i *Afegeix grup* que ens informen que les llistes de selecció permeten tenir les seves opcions agrupades. Així, per exemple, podríem construir una llista de comarques

96

agrupades per vegueries. En tal situació, aniríem afegint grups (les vegueries) i per cada grup afegiríem les opcions (les comarques), tal i com s'aprecia a la figura 60.

Observem, a la figura 60, que els textos introduïts contenen espais i qualsevol tipus de símbol, però en canvi, en els valors no hi introduïm cap símbol estrany. D'aquesta manera podem assegurar a l'agent que hagi de processar les dades que rebrà textos sense caràcters estranys. L'efecte visual seria similar al que s'observa a la figura 61.

Figura 61. Efecte visual d'una llista de selecció amb grups d'opcions

| 🕲 Comarques - Mozilla Firefox 📃 | |
|--|------------|
| Eitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda | \Diamond |
| 🦛 • 🔷 • 💽 🐼 🏠 🗋 file:///E:/pagina30.html 🔹 🕨 💽 • Google | Q |
| 🗋 Hotmail gratuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windows | |
| Comarques distribuïdes en vegueries segons proposta de l'Informe Roca (2001) Vegueria Barcelona | |
| Barcelonès Vallès Occidental Baix Llobregat Vallès Oriental Maresme Baix Llobregat Nord Garraf Alt Penedès Vegueria Camp de Tarragona Tarragonès Baix Camp Baix Camp Baix Penedès Alt Camp Conca de Barberà ▼ | |
| Fet | |

El codi HTML generat és similar a:

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina30.html* amb el codi de l'exemple.

```
<select size="15" name="comarques">
   <optgroup label="Vegueria Barcelona">
    <optgroup label="Vegueria Barcelon&egrave;s</option>
    <option value="barcelona">Barcelon&egrave;s</option>
    <option value="vallesOccidental">Vall&egrave;s Occidental</option>
    <option value="baixLlobregat">Baix Llobregat</option>
    <option value="vallesOriental">Vall&egrave;s Oriental</option>
    <option value="tarragones">Tarragon&egrave;s</option>
    <option value="tarragons">Tarragon&egrave;s</option>
    <option value="baixCamp">Baix Camp</option>
    <option value="baixPenedes">Baix Pened&egrave;s</option>
    </optgroup>
    </optgroup>
<//select>
```

En cas que es permeti selecció múltiple, caldrà incloure l'atribut multiple="true" a la marca <select>. L'atribut size indica el número de files que visualitzarà el navegador. El valor de l'atribut name és el que s'enviarà a l'agent que hagi de processar les dades del formulari acompanyat del valors escollits. En cas de selecció múltiple, s'enviaran tantes parelles comarques=valor com opcions seleccionades.

2.4.8. Observacions finals

Per a tots els controls definits cal tenir en compte les següents observacions:

1) És aconsellable, sempre que sigui possible, que el seu nom sigui descriptiu del seu contingut i, com sempre, cal evitar els caràcters accentuats i els espais.

2) En els controls que hagin d'anar acompanyats de valors introduïts per l'autor del formulari (botons d'opció, caselles de selecció, llistes de selecció,...) pot ser convenient que els valors tampoc continguin caràcters accentuats si l'agent que ha de processar les dades ha d'executar actuacions diferenciades segons els valors rebuts. Si no utilitzem caràcters accentuats assegurem el text que rebrà sempre l'agent processador.

3) Poden incorporar l'atribut tabindex=valor que permet definir l'ordre de tabulació dels camps, és a dir, l'ordre en que l'usuari anirà passant pels camps en prémer el tabulador. Si no s'indica, es seguirà l'ordre en que els camps estan definits dins el codi HTML, però a vegades pot interessar utilitzar aquest atribut.

4) Es poden desactivar, impedint així que l'usuari els utilitzi. Es seguiran mostrant en pantalla, però amb un aspecte diferent per indicar el seu estat de desactivació. Per aconseguir-ho cal afegir l'atribut disabled al corresponent control.

La utilitat d'aquesta opció és similar a la utilitat de l'atribut readonly, i és que l'estat d'activació/desactivació d'un control és accessible des de *JavaScript* i això permetrà activar/desactivar una part del formulari segons els valors que l'usuari hagi introduït prèviament en altres controls.

5) El text que acompanyi als controls pot escriure's de manera totalment independent del control, tal i com s'ha mostrat en els exemples precedents. Hi ha, però, la possibilitat d'enganxar un text a un control, amb la utilització de la marca <label>. Amb la utilització d'aquesta marca, quan l'usuari premi damunt l'etiqueta, el corresponent control canviarà l'estat, fet que no succeeix si el text és independent del control.

Vegem la diferència entre el codi HTML corresponent a un no vinculat a un control i el corresponent a una etiqueta vinculada al control.

• Text *Nom* no vinculat al control nom:

Nom: <input name="nom" type="text">

• Text Nom vinculat al control nom per medi d'una etiqueta:

<label>Nom: <inpunt name="nom" type="text"></label>

6) Hi ha la possibilitat d'agrupar varis controls de manera que el navegador mostri un rectangle al voltant d'ells. Això s'assoleix amb la marca <fieldset> que, a més, pot anar acompanyada d'un títol amb la marca <legend> (que permet el paràmetre align).

Un exemple d'utilització podria ser d'aquesta funcionalitat el tenim en el següent codi HTML, visualitzat a la figura 62.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de formulari de correu</title>
</head>
<body>
<form action="mailto:usuari@xtec.cat" method="post" enctype="text/plain">
<fieldset>
  <legend>Dades personals</legend>
  <label>Nom: <input name="nom" type="text"></label><br>
 <label>Cognom: <input name="cognom" type="text"></label><br><label>Sexe: <input name="sexe" value="Home" type="radio">Home
                <input name="sexe" value="Dona" type="radio">Dona
  </label>
</fieldset><br>
<label>e-Mail: </label><input name="email" type="text"><br><br>
<input value="Enviar" type="submit"><input value="Netejar" type="reset">
</form></body></html>
```

(**!**)

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina31.html* amb el codi de l'exemple.

2.5. Plantilles

Una plantilla és un document que incorporen alguns editors especialitzats en pàgines web (com per exemple *Nvu* i *Dreamweaver*) que serveix per a la creació d'un nombre elevat de pàgines que comparteixen un mateix disseny. Les plantilles faciliten enormement el manteniment del lloc web, ja que permeten modificar-lo o bé redissenyar l'aspecte de totes les pàgines ràpidament.

Les plantilles tenen el seu origen en programes de maquetació adreçats al disseny imprès, on era imprescindible poder fer canvis a diferents elements de les pàgines màsters (navegació, tipografia, gamma de colors...), que després s'aplicaven a les pàgines resultants de la manera més àgil possible.

La incorporació d'aquesta eina permet modificar alguns dels elements d'una plantilla fent que canviïn totes i cadascuna de les pàgines creades a partir d'aquesta. Això resulta d'especial utilitat quan s'elaboren webs de centres, revistes electròniques escolars..., on el disseny general no canvia (capçalera, fons...), però sí els títols i els continguts (apartats, articles...). Són de gran utilitat quan treballem amb els alumnes, ja que el dissenyador controla l'aspecte general i la resta de components de l'equip de treball incorporen contingut, però no tenen la possibilitat de canviar el disseny.

2.5.1. Dissenyar plantilles en Nvu

Per crear una plantilla, podem utilitzar la fletxa avall de la dreta del botó *Nou* de la barra d'eines de composició i escollim l'opció *Més opcions*. També hi podem arribar per *Fitxer*|*Nou*. En qualsevol cas apareix una finestra com la de la figura 18 on hem d'escollir l'opció *Plantilla en blanc* i procedir a la creació.

En aquest moment, el codi HTML que ha generat aquest procés tan simple és:

A partir d'aquest moment cal afegir el contingut que es voldrà repetir en les pàgines basades en la plantilla utilitzant els mecanismes normals d'edició de pàgines web.

Nvu bloqueja automàticament la majoria de les regions del document plantilla. L'autor de la plantilla ha d'especificar quines regions del document basat en la plantilla seran editables, fet que s'aconsegueix inserint àrees editables.

En el document plantilla es poden efectuar canvis tant en les àrees editables com en les àrees bloquejades. No obstant, en un document basat en una plantilla, l'usuari de la plantilla només podrà realitzar canvis en les àrees editables. És clar que per a que una plantilla sigui efectiva com a plantilla cal que com a mínim contingui una àrea editable, doncs del contrari els documents basats en la plantilla serien calcats a la plantilla en no tenir cap zona editable.

Les plantilles de Nvu s'enregistren en fitxers amb extensió mzt.

L'opció *Insereix* |*Plantilles* |*Insereix àrea editable* permet la inserció d'àrees editables a partir de la finestra com la de la figura 63.

 Insercix una àrea editable

 Nom de l'àrea

 Si et plau, dona un nom a aquesta àrea editable:

 Tipus d'àrea

 Moviment del text

 Bloc

 Opcions

 L'àrea és opcional

 L'àrea és repeatable

 L'àrea és movible

 D'acord
 Cancel·la

Figura 63. Pantalla Nvu per a les opcions de les àrees editables de les plantilles

Cada àrea editable ha de tenir obligatòriament un nom. Aquest apareix dins el document com una etiqueta que identifica l'àrea i també és utilitzat com a contingut inicial de l'àrea el qual pot ser modificat.

La figura 63 ens mostra que una àrea pot ser de tipus *Moviment de text* (*flow* en anglès) o de tipus *Bloc*.

- Una àrea *flow* acompanya al text en un paràgraf o capçalera i pren el format del context en el que es troba. El codi HTML generat queda emmarcat entre i .
- Una àrea *bloc* és una àrea independent, similar a un paràgraf de text. El codi HML generat queda emmarcat entre <div> i </div>.

A la figura 63 també veiem que en definir una àrea editable, hi ha tres possibles opcions a activar que li donen diferent funcionalitat. Es pot activar qualsevol combinació de les tres opcions. Aquestes són:

1) *L'àrea és opcional*, permet definir una àrea per la que l'usuari de la plantilla en dissenyar un document basat en la plantilla pugui decidir si manté l'àrea o no la manté.

2) L'àrea és repetible, que permet definir una àrea per a que es repeteixi un número de vegades. L'usuari de la plantilla en dissenyar un document basat en la plantilla, podrà fer tantes còpies de l'àrea repetible com sigui necessari. Per exemple, es pot definir que una fila d'una taula sigui repetible.

3) L'àrea és movible, que permet definir l'àrea com una capa.

Capes

Les capes són bàsicament contenidors o divisions de codi HTML. Com a tals, poden contenir al seu interior una pàgina independent en HTML, text, imatges, formularis, objectes Flash, connectors (*plug-in*)... proporcionant la possibilitat de dotar la pàgina d'un major grau de dinamisme.

Les capes són molt útils per organitzar la informació d'una pàgina, jugar amb efectes o construir menús. La utilització de capes als fulls d'estil en cascada proporcionen una major optimització i netedat del codi HTML que guanya en flexibilitat, accessibilitat i usabilitat.

La gran flexibilitat per disposar-les a qualsevol lloc de la pàgina, apilar-les, moure-les per la pantalla mitjançant una línia de temps..., dóna un gran control sobre el disseny que es porta a terme.

Les taules en HTML serveixen per mostrar dades tabulars, però l'opció de donar valor 0 a l'atribut border de les taules ha fet que molts dissenyadors hagin emprat les taules per dissenyar text i imatges. L'ús de les taules és encara un dels mètodes dominants per dissenyar visualment llocs web complexos, però en realitat interfereix en la creació d'un lloc web accessible, flexible i funcional que segueixi la normativa internacional sobre l'accessibilitat. Aquest és el motiu principal de la utilització de capes enlloc de taules per a estructurar les pàgines web.

Però per a treballar amb capes és convenient disposar d'un editor especialitzat en pàgines web que les suporti i en faciliti eines per a una bona gestió. La versió 1.0 de Nvu proporciona les eines bàsiques per a la gestió de capes (botons Capa, Envia al davant i Envia al fons de la Barra d'eines de format), però aquesta versió de Nvu no està preparada totalment per a treballar amb capes, per la qual cosa la seva utilització pot ser frustrant.

La figura 64 mostra una plantilla de *Nvu* que incorpora 6 àrees identificables per etiquetes de color blau. Les tres primeres, de noms Zona1, Zona2 i Zona3 són de tipus *flow*. Les restants Zona4, Zona5 i Zona6 són de tipus *bloc*. Al costat de les etiquetes hi apareix el mateix nom de l'etiqueta com a contingut inicial de l'àrea que podem modificar.

La plantilla incorpora 6 separadors que apareixeran obligatòriament en qualsevol document basat en aquesta plantilla i que restaran bloquejats. Les àrees Zona2 i Zona5 van acompanyades per una petita icona que permetrà eliminar-les en els documents basats en aquesta plantilla. Això és degut a que aquestes àrees s'han marcat com a opcionals.

Les àrees Zona3 i Zona6 van acompanyades per una petita icona que permetrà efectuar-ne vàries còpies en els documents basats en aquesta plantilla. Això és degut a que aquestes àrees s'han marcat com a repetibles.

| Figura 64. Exemple de plantilla en <i>Nvu</i> | |
|--|----|
| 👻 Plantilla [file://plantilla32.mzt] - Nvu | |
| Ejtxer Edita Visualitza Insereix Format Iaula Eiges Ajuda | |
| Nou Obre Desa Publica Navega Àncora Enlla; Imatge Taula Formulari Verifica Imprimeix | vu |
| Text del cos | |
| Amplada variable 🔽 🎕 🕲 🕽 👖 🔢 🖉 🗮 🗮 🍬 🚧 👘 🖉 💆 | |
| Plantilla | * |
| 741px | |
| Zona1 Zona1 Zona2 © Zona2 Zona3 © Zona3 | |
| Cona4 Zona4 Zona5 © Zona5 | |
| Zona6 D Zona6 | |
| | |
| Normal I i marcadurs ni mi I i Cool ronc Himit i K Previsualizacio j | _ |
| <doqa></doqa> | |

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *plantilla32.html* amb el codi de l'exemple.

El codi HTML generat corresponent a la plantilla de la figura 64 és:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html template="true">
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Plantilla</title>
</head>
<body>
<span editable="true" id="Zonal">Zonal<br></span>
<hr style="width: 100%; height: 2px;">
<span editable="true" optional="true" id="Zona2">Zona2<br></span>
<hr style="width: 100%; height: 2px;">
<span editable="true" repeatable="true" id="Zona3">Zona3<br></span>
<hr style="width: 100%; height: 2px;">
<div editable="true" id="Zona4">Zona4<br></div>
<hr style="width: 100%; height: 2px;">
<div editable="true" optional="true" id="Zona5">Zona5<br></div>
<hr style="width: 100%; height: 2px;">
<div editable="true" repeatable="true" id="Zona6">Zona6<br></div>
<hr style="width: 100%; height: 2px;">
</body>
</html>
```

2.5.2. Utilització de plantilles en Nvu

Per a crear pàgines web basades una plantilla , podem utilitzar la fletxa avall de la dreta del botó Nou de la barra d'eines de composició i

escollim l'opció *Més opcions*. També hi podem arribar per *Fitxer*|*Nou*. En qualsevol cas apareix una finestra com la de la figura 18 on hem d'escollir l'opció *Un nou document basat en plantilla*, indicar l'arxiu que conté la plantilla i procedir a la creació de la pàgina web. La capçalera de la nova pàgina web té per codi HTML:

Fixem-nos que la marca <html> va acompanyada de l'atribut templateref indicant la ubicació de la plantilla en la que es basa aquesta pàgina web. És important utilitzar una adreça relativa al lloc on ubicarem la pàgina web enlloc de mantenir l'adreça absoluta que assigna *Nvu*.

La visualització inicial de la nova pàgina web a la zona d'edició de *Nvu* és idèntica a la de la figura 64 obtinguda en editar la corresponent plantilla. A partir d'aquest moment podem efectuar les següents operacions: a) Editar el contingut de les àrees editables

- **b)** Eliminar les àrees que eren opcionals
- c) Clonar les àrees repetibles tantes vegades com correspongui.

Per a desvincular una pàgina web de la seva plantilla només cal executar l'opció *Edita*|*Separa de la plantilla* que provoca la desaparició de l'atribut templateref de la zona <head> de la pàgina i la visualització de les àrees editables en la zona d'edició de *Nvu*.

Limitacions relatives a plantilles en la versió 1.0 de Nvu

La versió 1.0 de Nvu té algunes limitacions en la utilització de plantilles:

No permet editar el text de les àrees repetibles.

 En modificar una plantilla no pemet modificar les pàgines web basades en la plantilla, de manera que la gestió de plantilles en la versió 1.0 es limita a servir com a base per a crear pàgines web inicialment iguals a la plantilla.

2.6. Disseny d'interfícies

Ja ha quedat enrere l'època en què una web era un conjunt de textos enllaçats. Avui dia una web es defineix més aviat com un aparell per fer coses: descarregar MP3, buscar feina, participar en subhastes, fer transaccions bancàries, comprar bitllets d'avió i entrades de teatre, mirar per càmeres web com està el trànsit, parlar per xats, enviar cartes a l'altre extrem del planeta, traduir textos d'un idioma a un altre, escoltar la ràdio en línia, veure vídeos... La llista és interminable.

Això vol dir que, actualment, una web ja no és com un document que obrim amb un programa anomenat navegador. Si una web serveix per fer coses, s'assembla més a un programa que a un document.

Ara bé, els programes estan dotats del que s'anomena una interfície gràfica d'usuari (GUI, *Graphical User Interface*). La interfície del *Writer*, el *Word*, per exemple, la formarien la barra de títol i d'estat, els menús, les eines, els regles... És la part relativament fixa que permet interactuar (fer coses) amb el programa.

Les webs actuals requereixen també una interfície d'usuari. Si una web és com un aparell, exigirem (com a tots els aparells) que sigui fàcil d'utilitzar, de fer-hi coses. Per això es parla tant ara d'usabilitat o facilitat d'ús. La presència d'una interfície d'usuari entenedora augmentarà exponencialment la usabilitat de la vostra web.

2.6.1. Patrons

Ja hem precisat que la interfície d'usuari ha de ser entenedora; en cas contrari, no hi guanyeu gran cosa. Què es pot fer perquè sigui entenedora? Seguint simplement algun del patrons actuals.

Figura 65. Pàgina web inicial del porta de Yahoo a Espanya 100! España - Mozilla Firefo Edita Visualitza Historial Adreces d'Interès Eiges Aguda 🔹 🧼 - 🥑 🔯 🏠 📄 http://es.yahoo.com • 🕨 🖸 • 🕫 🖻 Hotmail gratuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windo escargar la Barra de Yahoot 🕴 Haz de Yahoot tu página de in mo es tu página de inicio perfecta? **IOO!** Buscar: Web ESDANA 🖸 Ya 💿 en la Web 🔘 en españo Mi Yahool Mi correo Astrología Hoy en Yahoo! Correo Yahool: Entrar Crear tu cuenta gratis Chat Guardaespaldas para Knut Correo Cine Messenger Una amenaza de muerte dirigida al osezno Knut pone en estado de aler Zoo de Berlín » **seguridad reforzada** rta el Coches Compras Rnut en la seb
 Grupos de ecología Corazón Deportes SUN Encuentro l osezno Knut tiene un nemigo mortal Te llevamos de concierto por Europa 1 Finanzas El asesino de Virginia grabó su "testamento" Peligro y destrozos por el volcán nevado del Huila Grupos Inmobiliaria Deportes Corazón Juegos a reserva de petróleo inaquí podría ser el doble de lo pensado 2) precio de la vivenda sube 'solo' un 7,2% en el último año 4) unan salió del hospital por trazones médicas y con permis 3) pe a la prostitución: Al menos 50 detenidos en Almeria os efectos de una noche sin BlackBary en estados Unidos talán un manuscrito de 1839 de un legendario compositor ind ho araquía da la luta a secolítica Loterias Móvil Música Yahoo! Viajes Págs, Amarillas 📓 En busca del sol En busca del sol Postales Date un capricho: tranquilidad Respuestas > Mási Ne ayas y sol te esperan en el Viales ador: IBEX 35: -1.14% C/USD: 1.358 • Viajeo al Car Servicios de Yahoo! Cottraciones: Yahoo! Barómetro Tus compras online Servicios Coches deseados: los nuevos modelos S'està esperant l. ying.com.

Anem una mica enrere per comprendre el concepte de patró i el seu origen. Ja fa uns quants anys, dos estudiants van crear el *Yahoo*! (la figura 65 ens en mostra la seva pàgina inicial), el primer cercador de la història. L'èxit va ser fulgurant i, com és natural, tothom es va llançar a fer cercadors. A l'hora de dissenyar-los es podrien haver fet amb una interfície diferent; però es tractava de robar visitants, i el *Yahoo*! funcionava tan bé que a ningú no se li va ocórrer que fos millorable.

El cas és que, si no s'hagués imitat la interfície del *Yahoo*!, no hagués passat res. Però com que aquesta interfície s'ha anat repetint a centenars de cercadors, els usuaris l'han interioritzat. Una estructura interioritzada per milions d'usuaris és un patró. Els resultats de tot plegat són dos: primer, que teniu un patró anomenat LSD (acrònim de *Logo - Search - Directory*); segon, que si dissenyeu un cercador seguint aquest patró, els visitants l'entendran de seguida, i si us inventeu un patró completament diferent, topareu amb la incomprensió de molts usuaris.

2.6.2. El model 3P

Quin dels patrons disponibles cal escollir a l'hora de dissenyar una web? Un d'adient a les vostres necessitats i que estigui fortament interioritzat. Per exemple, el model dels tres panells. Els experts en usabilitat ens diuen, a més, que aquest model satisfà a la perfecció les expectatives de l'usuari. L'ordre i les formulacions són lleugerament diferents, però tots ens diuen que quan un usuari va a parar a una web vol saber immediatament tres coses:

• On sóc? • Què hi ha? • On puc anar?

I cada àrea de la pàgina (figura 66) li dóna la resposta corresponent:

- · L'àrea de logotip (extrem superior esquerre) identifica el lloc web
- Les àrees de navegació (zona superior horitzontal i zona esquerre vertical) enumeren els llocs on pot anar
- L'àrea de contingut (zona central) mostra el que hi ha.

Vegem aquests elements amb més detall.

1) Identitat

En aquest model la identitat queda destacada a l'àrea superior esquerra amb un logotip que es va repetint a totes les pàgines. Quan el logotip no és prou expressiu, s'afegeix una etiqueta (*tagline*) o frase explicativa. Exemple: *Dibuix d'un gat simpàtic* + *Miawww*! + *Tot el que vols saber del teu gat*.

Figura 66. Exemplificació del model 3P

Manteniment dels portals d'informació

| Logotip | Tagline Utilitats |
|--|----------------------------------|
| | Navegació horitzontal / Seccions |
| Navegació vertical / Pàgines de la secció activa | Títol de la pàgina |
| | Continguts |
| | |

2) Navegació

La barra horitzontal conté vincles a cada una de les seccions de la web. Quan fem clic en una d'aquestes seccions, la columna de l'esquerra mostra els vincles a les diferents pàgines de la secció.

També a la part superior se sol posar una línia d'utilitats no relacionades directament amb els continguts: mapa de la web, correu electrònic, suggeriments, fòrums, enllaços, etc.

Ja sigui a la barra horitzontal o a la línia d'utilitats, és convenient posar algun vincle que s'anomeni *Inici* o alguna cosa semblant, ja que sovint el visitant ve d'un cercador o vincle extern i va a parar a una pàgina secundària. Sovint aquesta funció també s'assigna al logotip, però molts usuaris no ho saben.

No ens podem estar de destacar el potencial d'aquest model. A la barra horitzontal podem encabir, per exemple, sis seccions. A la columna de cada secció hi caben fàcilment 10 vincles visibles sense fer desplaçaments (és convenient que tots els vincles importants siguin visibles a la part superior de la pàgina, perquè l'usuari pugui comparar i triar sense haver d'anar amunt i avall). Això dóna un total de 60 pàgines perfectament ordenades i accessibles amb dos clics des de qualsevol lloc de la web. Si cada secció la dividíssim en sis subseccions, arribaríem a tenir 360 pàgines a tres clics de qualsevol lloc.

3) Continguts

Les àrees de navegació solen experimentar canvis contextuals (per exemple, que el botó de la secció tingui una aparença diferent quan s'és a la secció), però la part clarament variable de la web és l'àrea de continguts. En aquesta àrea és fonamental destacar visualment el títol de la pàgina. Convé utilitzar per al títol les mateixes paraules del vincle que ha portat a la pàgina per no crear desconcert.

2.6.3. Mides de la interfície

Quan un editor dissenya una revista, sap que tots els lectors rebran la mateixa revista. Però, com ja sabeu, la visualització de la web depèn de factors que no podem controlar, com per exemple la resolució del monitor del visitant. Els professionals del web disposen de mecanismes per resoldre aquest problema. Nosaltres, com a amateurs que som, ens hem de conformar a mirar les estadístiques, que diuen que la resolució predominant entre els usuaris és de 800 x 600. La resolució inferior (640 x 480) gairebé s'ha extingit, i la superior (1.024 x 764) va agafant cada vegada més quota d'utilització. Per tant, dissenyarem la interfície pensant en una resolució de 800 x 600, i procurant que també es vegi bé a resolucions superiors.

Si ara observem els navegadors *Mozilla Firefox* o *Internet Explorer*, notarem que, d'aquests 800 píxels d'amplada del monitor, una part se la "cruspeix" el navegador: contorn de la finestra i barra de desplaçament. A més, els navegadors deixen un marge per defecte entre la finestra i el contingut de la pàgina. Si no eliminem aquest marge, l'amplada disponible és de 760 píxels. Per a eliminar el marge (i arribar a 780 píxels disponibles), cal definir els atributs margin-left i margin-right a 0 píxels en la marca
body>, cosa que es pot fer amb un full d'estil:

```
<style type="text/css">
body {
    margin-left: 0px;
    margin-right: 0px;
}
</style>
```

Considerem també que el visitant ve a la web pels continguts. Per tant, l'àrea de continguts ha de ser suficientment gran. Se sol dir que ha d'ocupar un 80 % o més de l'espai disponible. Això vol dir que, a una resolució de 800 x 600, l'àrea superior hauria d'estar al voltant dels 80 píxels d'alçada, i l'àrea esquerra no hauria de tenir més de 140 píxels d'amplada.

2.7. Marcs

En ocasions ens trobem amb pàgines web que segueixen el model dels tres panells en les que l'àrea superior i esquerra de la pàgina, corresponents a les àrees de logotip i navegació de la interfície de la web, no es recarreguen quan fem clic en un vincle, sinó que es mantenen estàtiques durant la navegació.

Com s'aconsegueix això? Simplement, creant el que s'anomena una pàgina de definició de marcs. Aquesta pàgina indica al navegador que ha de dividir la pantalla en una sèrie de parts que s'anomenen marcs, i que a cada una hi ha de mostrar una pàgina HTML diferent.

Suposem, com es veu a la figura 67, una pàgina de definició de marcs anomenada *index.htm*. Aquesta pàgina pot establir, per exemple, que hi hagi un marc superior on es mostri la pàgina *superior.htm*, i que el marc inferior es divideixi al seu torn en dos marcs: un marc esquerre estret que contingui la pàgina *esquerra.htm*, i un marc més ample on es mostri la pàgina *benvinguts.htm* i totes les altres pàgines quan fem clic als vincles. Com podeu imaginar, les pàgines poden tenir qualsevol altre nom.





2.7.1. Marca <frameset>

Una pàgina de definició de marcs és una pàgina web que no té zona <body> i en el seu lloc hi ha la marca <frameset> que indica al navegador quants marcs hi ha d'haver, les mides dels marcs i la pàgina que s'ha de mostrar a cada marc.

Files i columnes

La marca <frameset> pot anar acompanyada dels atributs rows (files) per definir el número de subespais horitzontals i cols (columnes) per definir el número de subespais verticals. Ambós atributs es poden especificar simultàniament per a crear una quadrícula.
Si no s'estableix l'atribut rows, cada columna s'estén al llarg de tota la longitud de la pàgina. Si no s'estableix l'atribut cols, cada fila s'estén al llarg de tota l'amplada de la pàgina. Si no s'estableix cap dels dos atributs, el marc té exactament la mateixa grandària que la pàgina.

Els marcs es creen d'esquerra a dreta per les columnes i de dalt a baix per les files. Quan s'especifiquen ambdós atributs, les vistes es creen d'esquerra a dreta en la fila superior, d'esquerra a dreta en la segona fila, etcètera.

En aquest primer exemple dividim la pantalla verticalment en dos, és a dir, creem una meitat superior i una meitat inferior:

```
<frameset rows="50%, 50%">
...la resta de la definició...
</frameset>
```

El següent exemple crea tres columnes: la segona té una amplada fixa de 250 píxels, la qual cosa és útil, per exemple, per incloure una imatge de grandària coneguda. La primera reb el 25% de l'espai restant i la tercera el 75% de l'espai restant.

```
<frameset cols="1*,250,3*">
...la resta de la definició...
</frameset>
```

El següent exemple crea una quadrícula de 2x3 subespais.

```
<frameset rows="30%,70%" cols="33%,34%,33%">
...la resta de la definició...
</frameset>
```

Pel següent exemple suposem que la finestra del navegador té actualment una altura de 1000 píxels. Per la primera vista s'assigna el 30% de l'altura total (300 píxels). Per la segona vista s'especifica que tingui una altura exacta de 400 píxels. Això deixa 300 píxels per a repartir entre els altres dos marcs. L'altura del quart marc s'ha especificat com 2*, de manera que és el doble d'alt que el tercer marc que té altura * (equivalent a 1*). Per tant, el tercer marc tindrà 100 píxels d'altura i el quart tindrà una altura de 200 píxels.

```
<frameset rows="30%,400,*,2*">
...la resta de la definició...
</frameset>
```

Les longituds absolutes que no sumin el 100% de l'espai real disponible haurien de ser ajustades pels navegadors. Si sobra espai, aquest s'hauria de repartir proporcionalment entre cada vista. Si falta espai, s'hauria de reduir cada vista segons la relació entre l'espai indicat i l'espai total.

Vores dels marcs

En general, els navegadors dibuixen una vora de separació entre els marcs. Per eliminar-lo es pot utilitzar l'atribut frameborder=0. Tot i així, el navegador encara deixa un espai entre marcs. Per eliminar-lo cal utilitzar els atributs framespacing=0 i border=0.

Imbricació de marcs

Els grups de marcs es poden imbricar fins a qualsevol nivell. En el següent exemple, el <frameset> exterior divideix l'espai disponible en tres columnes iguals. A continuació el <frameset> interior divideix la segona àrea en dues files d'altures diferents.

```
<frameset cols="33%, 33%, 34%">
...continguts del primer marc...
<frameset rows="40%, 50%">
...continguts del segon marc, primera fila...
...continguts del segon marc, segona fila...
</frameset>
...continguts del tercer...
</frameset>
```

2.7.2. Marques <frame> i <noframes>

El contingut i l'aparença d'un marc es defineix amb la marca <frame>. La taula mostra els paràmetres que admet aquesta marca.

| Paràmetre | Utilitat |
|--------------|--|
| name | Assigna un nom a un marc per a poder-s'hi referir posteriorment |
| src | Indica la direcció del document HTML que ocuparà el marc |
| scrolling | Decideix si es col·loquen o no barres de desplaçament en el marc per a poder-nos moure pel seu contingut. El seu valor per defecte és auto que deixa la decisió al navegador. Les altres opcions són yes i no. |
| noresize | Si s'especifica, l'usuari no podrà canviar la grandària del marc |
| frameborder | A l'igual que el seu homònim en la marca <frameset>, si l'igualem a zero s'eliminarà la vora amb tots els marcs contigus que tinguin també aquest valor a zero.</frameset> |
| marginwidth | Permet canviar els marges horitzontals dins un marc. Es representa en píxels. |
| marginheight | Permet canviar els marges verticals dins un marc. Es representa en píxels. |

Taula 16. Paràmetres de la marca <frame> per a definir el contingut i aparença dels marcs.

La marca <noframes> permet especificar què mostrarà el navegador en cas que no suporti marcs (situació cada vegada més difícil de succeir) o que el navegador els tingui desactivats.

111

A la secció "Recursos" del web d'aquest crèdit podeu

trobar el fitxer *plantilla33.html* amb el codi de la solució.

<u>Exemple</u>: Dissenyar una pàgina de definició de marcs similar a la de la figura 67

Solució:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de p&agrave;gina amb marcs</title>
</head>
<frameset rows="20%,80%">
<frame name="superior" src="pagina33superior.html">
  <frameset cols="20%,80%">
    <frame name="esquerra" src="pagina33esquerra.html">
    <frame name="exemple" src="pagina33benvinguts.html">
  </frameset>
  <noframes>
    Ho sento, pero només pots veure aquesta pàgina
   si el teu navegador té la capacitat de visualitzar marcs.
  </noframes>
</frameset>
</html>
```

2.7.3. Marcs en Nvu

La versió 1.0 de l'editor *Nvu* no suporta marcs i si s'obre un document amb marcs apareix el missatge d'alerta *No és possible editar aquesta pàgina a causa d'un motiu desconegut*, però *Nvu* visualitza el contingut dels marcs com si d'un navegador es tractés. Ara bé, *Nvu* no permet altra cosa que visualitzar el codi font HTML.

3. HTML dinàmic amb JavaScript

La finalitat d'aquest nucli és ampliar les possibilitats que ens dona el llenguatge HTML i els fulls d'estil CSS amb la utilització del llenguatge JavaScript per tal d'assolir més funcionalitat en les pàgines web.

L'HTML dinàmic (abreujat com a DHTML), és el mot emprat per a definir la utilització combinada del llenguatge HTML, els fulls d'estil CSS i el llenguatge JavaScript per al disseny de pàgines web.

Els tres components del DHTML s'executen sempre a la màquina client, és a dir, en el navegador on l'usuari visualitza la pàgina web. (II)

Cal saber que hi ha altres llenguatges que donen dinamisme a les pàgines web en el propi servidor web, abans de lliurar-les al navagador. Aquestes possibilitats, però, no són l'objecte d'estudi en aquest tema. Són els anomenats llenguatges de servidor (per exemple, *ASP* i *PHP*).

Nosaltres ja som coneixedors d'HTML i de fulls d'estil CSS i ara ens cal endinsar-nos en el món del JavaScript.

3.1. JavaScript bàsic

JavaScript és un llenguatge que s'utilitza per ampliar les capacitats de l'HTML i per inserir petites accions en pàgines web. Permet crear efectes especials i aporta interactivitat.

L'execució del codi JavaScript que hi pot haver en una pàgina web (dins el document HTML) és simple: el navegador interpreta i executa les instruccions que es troba, les quals li arriben per la xarxa tal i com les va escriure l'autor del document HTML. En aquest aspecte, el llenguatge JavaScript és totalment diferent als llenguatges de programació que s'utilitzen per a obtenir aplicacions informàtiques i és més similar a les macros que incorporen algunes aplicacions ofimàtiques (es llegeixen de forma seqüencial i es van interpretant i executant).

Llenguatges de programació per a obtenir aplicacions informàtiques

En el desenvolupament d'una aplicació informàtica hi ha diferents fases que podríem resumir amb: anàlisi del problema, disseny de la solució, programació de la solució dissenyada, verificació del programa obtingut, instal·lació i posta en marxa.

En la tercera fase (programació de la solució) s'utilitza un o varis llenguatges de programació. En aquesta fase, els programadors escriuen els programes en el llenguatge que correspongui obtenint el codi font, posteriorment aquest codi font es tradueix (fase de compilació) a un llenguatge que pot entendre el sistema operatiu generant el codi objecte i, per últim, pot ser necessari l'enllaç dels diversos codis objectes per a obtenir el codi executable que és el que s'instal·larà en els ordinadors que hagin d'executar l'aplicació informàtica. Els codis objecte i executable són il·legibles i no s'assemblen en res al codi font que han escrit els programadors.

Un document HTML pot contenir diferents programes JavaScript corresponents a diferents tasques que calgui poder executar en l'agent d'usuari (navegador). Cada unitat de programació s'anomena script.

Tot el codi JavaScript que s'insereix en els documents HTML ha d'anar tancat per les marques <script> i </script>. Aquestes marques són les que HTML facilita per incloure, dins un document HTML, scripts escrits en algun llenguatge admès (no només JavaScript).

Cal informar al navegador del tipus de llenguatge emprat en l'script i per aquest motiu es facilita l'atribut language. Així doncs, un script de JavaScript s'hauria d'escriure com:

```
<script language="JavaScript">
...
</script>
```

El codi que es trobi entre les marques <script> i </script> es llegirà en el moment que la càrrega de la pàgina llegeixi el seu contingut. Poden estar a la secció <head> o a la secció <body>.

3.1.1. El primer programa JavaScript

El primer programa que s'acostuma a desenvolupar en l'aprenentatge de qualsevol llenguatge és la visualització del missatge "Hola Món!!!". Ja ho varem fer així en escriure la nostra primera pàgina web. Vegem-ne el codi:

Per inserir codi JavaScript via l'editor *Nvu* ens caldrà utilitzar la visualització *Codi font HTML* i escriure el codi on correspongui.

La visualització d'aquest document provoca el mateix resultat que provocava la primera pàgina web que havíem dissenyat (figura 1). És clar

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina34.html* amb el codi de l'exemple. que no ens calia utilitzar JavaScript per assolir el mateix objectiu, però és un primer exemple i ens servirà per introduir alguns conceptes bàsics de JavaScript.

L'anterior document HTML inclou un únic script el qual té una única instrucció amb una funció molt simple: escriure la frase "Hola Món!!!". D'aquesta tasca se n'encarrega el mètode writeln() de l'objecte document. Observem que un punt (.) uneix l'objecte document amb el mètode writeln().

Més endavant presentarem què són els objectes i els mètodes, però ara només ens cal saber que els mètodes indiquen una opció a desenvolupar i, en el nostre exemple, el mètode writeln() s'ocupa d'escriure alguna cosa dins l'objecte document en el que està inserit (és a dir, el document HTML). Però, què escriurà el mètode writeln()? Caldrà poder-li dir què és el que ha d'escriure, no?

Alguns mètodes tenen paràmetres, que no són altra cosa que valors que cal comunicar als mètodes per a que puguin executar la seva funció i que es situen entre els parèntesis que acompanyen el mètode. Si no hi ha paràmetres, els parèntesis no inclouen res en el seu interior però igualment s'han d'utilitzar. Així, en el nostre exemple, com que volíem que el mètode writeln() escrivís "Hola Món!!!", hem hagut de comunicar-li aquesta frase via paràmetre i per això l'hem inclosa a l'interior dels parèntesis.

Observem que la frase "Hola Món!!!" està situada entre cometes dobles ("). Cal utilitzar les cometes dobles quan el valor a comunicar al mètode és una cadena de text. En l'argot informàtic s'entén per *cadena (string,* en anglès) un conjunt de caràcters alfanumèrics, habitualment inclosos entre cometes, és a dir, un text. Per tant, no és un valor que pugui ser utilitzat en càlculs numèrics.

La línia document.writeln(....); està finalitzada amb un punt i coma (;). Aquest element serveix per indicar el final d'una instrucció JavaScript, però, no és imprescindible si al final de la instrucció hi ha un salt de línia, com en l'exemple. En cas de tenir més d'una instrucció en la mateixa línia, és imprescindible el ; per separar-les.

En el codi JavaScript pot ser interessant escriure comentaris, a l'igual que en el codi HTML. Recordem que en el codi HTML, els comentaris s'escriuen entre les marques <!-- i -->. En JavaScript, hi ha dues maneres d'introduir comentaris:

• L'aparició de dues barres (//) indica que tot el que hi hagi a la seva dreta i fins al final de la mateixa línia és un comentari.

• Tot el que estigui tancat entre els símbols /* i */ (pot ocupar vàries línies) és un comentari.

Així doncs, podem posar comentaris en l'anterior script sense que incideixin en la seva funcionalitat:

```
<script language="JavaScript">
document.writeln("Hola Món!!!"); // Aquesta línia escriu "Hola Món!!!"
/* Observem que l'anterior línia finalitza amb un punt i coma (;).
No és obligatòria la seva utilització si a la línia només hi ha una instrucció,
com és el cas, però en canvi sí que és imprescindible si a la línia hi tenim més d'una
instrucció. */
</script>
```

3.1.2. Per què programar? Necessitats que se'n deriven.

Observem les següents situacions que se'ns poden plantejar en la confecció d'una pàgina web i quina seria una òptima solució.

a) Suposem que es necessita visualitzar els 100 primers números parells. És clar que en el codi HTML podem escriure:

```
2 4 6 8 10 12 14 16 18 20<br>
22 24 26 28 30 32 34 36 38 40<br>
...
```

Però, oi que seria d'agrair disposar d'algun mecanisme que fes el comptatge per nosaltres? De fet, el que estem fent per a poder visualitzar els 100 primers números parells és començar pel 2 (també haguéssim pogut començar per zero) i anar sumant 2 al número anterior, comptant quantes vegades sumem per tal d'aturar-nos en portar 100 valors. Oi que és una feina repetitiva?

Els llenguatges de programació (i JavaScript n'és un) aporten estructures per a executar tasques repetitives: són les estructures repetitives.

b) Aturem-nos a pensar en el funcionament del nostre cervell per a resoldre la situació anterior. Més o menys segueix el següent procés:

Pas 1: Valor parell = 2 i ja portem 1 valor Pas 2: Valor parell = Anterior+2 = 4 i ja portem 2 valors (els d'abans +1) Pas 3: Valor parell = Anterior+2 = 6 i ja portem 3 valors (els d'abans +1)

Observem que per aconseguir el nou valor parell necessitem tenir l'anterior i per saber quants valors hem assolit també necessitem tenir el número de valors que teníem assolits en el pas anterior. Aquest procés ens és tant evident que rarament hi pensem, però en realitat, en el nostre cervell, estem guardant en tot moment l'anterior valor i el número de valors que portem assolits.

Si pretenem que un programa informàtic faci aquest procés per nosaltres, haurem de facilitar-li una manera d'emmagatzemar valors: són les variables.

c) Suposem que tenim un formulari en el que hi ha un camp (casella de selecció, per exemple) per a que l'usuari ens digui si és o no major d'edat de manera que en funció de la seva resposta, s'activin uns determinats camps del formulari. És clar que per a poder assolir aquesta funcionalitat necessitem poder escollir entre unes accions o unes altres segons el valor de la casella de selecció.

Els llenguatges de programació (i JavaScript n'és un) aporten estructures per a poder prendre decisions: són les estructures condicionals.

3.2. Variables i tipus de dades

Les variables són uns elements de programació encarregats de contenir les dades que s'utilitzen en el programa. El seu valor pot canviar en funció de com evolucioni l'execució del programa.

Així, en el procés d'escriure els 100 primers números parells hi apareixen dues variables: la que va donant el número parell a partir de l'anterior parell més 2 i la que va comptant quants números parells hem assolit.

Cal tenir en compte:

• Les variables tenen un nom que ha de ser únic dins l'script i és aconsellable que sigui representatiu del seu contingut.

Cada llenguatge té unes normes respecte el nom (llargada, caràcters possibles,...). En el cas de JavaScript, els noms de les variables han de tenir com a primer caràcter una lletra o bé el guió baix (_) i pot ser qualsevol però no pot coincidir amb cap nom igual al de les paraules reservades del llenguatge i tampoc no pot contenir espais o caràcters especials (accents,...)

• Les variables cal declarar-les amb la següent sintaxis:

var nom_variable;

Declaració de variables

La majoria de llenguatges obliguen a declarar les variables (amb alguna paraula reservada similar a var) i fins i tot a indicar-ne el seu tipus. JavaScript no és tan estricte i no ens obliga a indicar el tipus i fins i tot ens podem saltar la utilització de la paraula var, però no és aconsellabletere del 2007 Pot anar acompanyada d'una assignació de valor inicial:

var nom_variable = valor_inicial;

- La declaració d'una variable es pot fer al principi de l'script o entremig en el moment que es produeixi la seva primera utilització. És aconsellable, però, posar totes les declaracions a l'inici de l'script.
- JavaScript diferencia entre majúscules i minúscules.
- En una mateixa línia es pot declarar més d'una variable. Només cal separar-les per comes:

var parell = 2, quantsEnPortem = 1;

A l'anterior declaració hi ha dues variables: <code>parell</code> amb valor inicial 2 i <code>quantsEnPortem</code> amb valor inicial 1.

Les variables emmagatzemen dades i les dades es classifiquen segons quin sigui el seu tipus. Hi ha llenguatges de programació que obliguen a declarar el tipus de dada de la variable (el tipus de valor que pot emmagatzemar) en el moment d'efectuar la declaració de la variable.

En el llenguatge JavaScript, si en la declaració de la variable no se li assigna cap valor inicial, la variable conté el valor null i encara no es sap el tipus de dada que emmagatzemarà. Posteriorment, quan se li assigna un valor, la variable passa a tenir assignat el tipus de dada corresponent al valor que se li ha assignat.

La taula 17 mostra els diferents tipus de dada que facilita JavaScript.

Taula 17. Tipus de dades del llenguatge JavaScript

| Tipus de dada | Descripció | Exemples |
|------------------|--|---|
| Enter | Per a representar dades numèriques enteres | 33, -435, 8294, -2413 |
| Coma flotant | Per a representar dades numèriques amb decimals. La part entera es distingeix de la part decimal amb un punt separador (no una coma!!!). També admet la sintaxis exponencial amb les lletres e i E. | 33.43, - 43.5, 25.0, 1432.465, 4e5 |
| Hexadecimal | Per a representar valors enters en notació hexadecimal, posant els caràcters 0x davant el número | 0x4A3E |
| Octal | Per a representar valors enters en notació octal, posant el caràcter 0 davant el número | 0225 |
| Booleà | Per a representar dos valors molt utilitzats en programació: true (cert) i false (fals) | true, false |
| Cadena | Per a representar una tira de caràcters (lletres, espais, números i símbols) tancats entre cometes dobles (") o simples ('). La cadena ha de finalitzar amb el mateix tipus de | "Hola, Món!!!" 'Hola Món!!!' |
| | cometa amo que s'ha iniciat. | |

Notació exponencial

23.4e4 equival a 23.4 x 10^4 43.3E5 equival a 43.3 x 10^5 50.2e-4 equival a 50.2 x 10^{-4}

| Tipus de dada | Descripció | Exemples | |
|------------------|--|---|--|
| | una barra invertida, cal utilitza del llenguatge C. Els més hab | r els caràcters d'escapament ituals són: | |
| | \n per a salt de línia | $\ ' \ per a cometa simple$ | |
| | $\ \ err$ a barra invertida $\$ | \" per a cometes dobles | |

3.3. Operadors

Els operadors són els elements de codi que ens serveixen per fer operacions amb les variables i altres valors constants. En JavaScript s'utilitzen bàsicament els mateixos operadors que en el llenguatge C.

JavaScript proporciona diferents operadors dels quals ens interessa principalment l'operador d'assignació, els operadors aritmètics, els operadors lògics, els operadors de comparació i els operadors de cadenes.

La majoria d'operadors són binaris, doncs actuen sobre dos operands, però també n'hi ha algun de monari.

 L'operador d'assignació (símbol =) és un operador binari que permet assignar un valor (operand de la dreta) a una variable (operant de l'esquerra).

Exemple:

i=3; /* La variable i ha quedat plena amb el valor 3 */

• Els operadors aritmètics permeten efectuar les conegudes operacions aritmètiques (suma, resta, producte, ...) i alguna més que podem observar a la taula 18.

| Tipus | Operador | Descripció | Exemple |
|---------|----------|---|---------|
| | + | Calcula la suma de 2 operands numèrics | r + 7 |
| BINARIS | _ | Calcula la resta de 2 operands numèrics | s – a |
| | * | Calcula el producte de 2 operands numèrics | 2 * i |
| | / | Calcula la divisió de 2 operands. Si tots dos operands són enters, efectua la divisió entera (sense baixar decimals), però si un operand és real, efectua la divisió real (calculant decimals) | a / b |
| | \$ | Calcula el mòdul de dos operands enters (residu de la divisió entera) | a % b |

Taula 18. Operadors aritmètics que proporciona JavaScript

| Tipus | Operador | Descripció | Exemple |
|--------|----------|---|------------|
| (0 | _ | Facilita el valor de l'operand que va a la dreta del signe, canviat de signe. | -a |
| ONARIS | ++ | Incrementa en una unitat l'operand que va a la seva dreta o a la seva esquerra | a++ ++a |
| Σ | | Disminueix en una unitat l'operand que va a la seva dreta o a la seva esquerra | a a |

Els operadors binaris i el monari "canvi de signe" s'acostumen a utilitzar en expressions aritmètiques emmagatzemant el resultat del càlcul en una altra variable. Així, per exemple, podríem tenir:

```
var a=5, b=2, r;
r = a + b; /* r passaria a valer 7 */
r = a - b; /* r passaria a valer 3 */
r = a * b; /* r passaria a valer 10 */
r = a / b; /* r passaria a valer 2 */
r = a % b; /* r passaria a valer 1 */
r = -a; /* r passaria a valer -5 */
a++; /* a passaria a valer 6 */
b--; /* b passaria a valer 1 */
```

• Els operadors lògics permeten efectuar operacions sobre dades booleanes (només admeten 2 valors: true i false), donant un altre valor booleà. La taula 19 en mostra els operadors possibles.

| Tipus | Operador | erador Descripció | | | | | | |
|---------|----------|--|--------|--|--|--|--|--|
| BINARIS | | Calcula la "conjunció" de dos operands booleans. | | | | | | |
| | δ. δ. | El resultat és true si i només si els dos operands tenen valor true i és false en cas contrari. | a && b | | | | | |
| | | Calcula la "disjunció" de dos operands booleans. | | | | | | |
| | | El resultat és false si i només si els dos operands tenen valor false i és true en cas contrari | a b | | | | | |
| MONARI | | Calcula la "negació" d'un operand booleà. | | | | | | |
| | ! | El resultat és false si l'operand té valor true i el resultat és true si l'operand té valor false. | !a | | | | | |

Taula 19. Operadors booleans que proporciona JavaScript.

• Els operadors de comparació permeten comparar dos operands (del mateix tipus) donant com a resultat un valor booleà (true o false). La taula 19 en mostra els operadors possibles.

| Operador | Descripció | Exemple |
|----------|--|---------|
| == | Retorna true si els dos operands són iguals i false en cas contrari | a == b |
| ! = | Retorna true si els dos operands són diferents i false en cas contrari | a != b |
| > | Retorna true si el l'operand de l'esquerra és major que l'operand de la dreta i false en cas contrari | a > b |
| < | Retorna true si l'operand de l'esquerra és menor que l'operand de la dreta i false en cas contrari | a < b |

Taula 20. Operadors de comparació que proporciona JavaScript.

| Operador | Descripció | Exemple |
|----------|---|---------|
| >= | Retorna true si l'operand de l'esquerra és major o igual que l'operand de la dreta i false en cas contrari | a >= b |
| <= | Retorna true si l'operand de l'esquerra és menor o igual que l'operand de la dreta i false en cas contrari | a <= b |

Per poder utilitzar els operadors >, <, >= i <= cal que en el tipus de dada dels operadors hi hagi definida una relació d'ordre. Això és possible en els valors numèrics i també en les cadenes, però no és possible en el tipus de dades booleà.

I en les cadenes, quina relació d'ordre hi ha? Quan una cadena és major que una altra?

Sembla que tots tenim clar quan dues cadenes són iguals i quan no ho són. Així, "HOLA" i "CASA" són diferents tot i que tinguin la mateixa longitud. Però, què ens dirà l'ordinador quan es troba comparacions com:

"HOLA" > "Pep" "Joanna" <= "ANNA" "CASA" <= "CASAL"

L'ordinador aplica l'*ordre lexicogràfic* per a saber quin és l'ordre relatiu entre dues cadenes i poder, així, respondre a les comparacions entre elles.

L'**ordre lexicogràfic** entre cadenes consisteix en anar comparant els caràcters de les dues cadenes, posició per posició, i des de l'inici de les cadenes fins que una de les dues finalitzi o els caràcters siguin diferents o totes dues finalitzin.

En cas de finalitzar el procés degut a la fi d'una cadena, aquesta és menor que l'altra.

En cas de finalitzar el procés degut a dos caràcters diferents, la cadena que conté el caràcter de posició més baixa a la taula ASCII és la cadena menor.

En cas d'arribar a la fi de les dues, és que les cadenes eren iguals.

Així, tindríem que:

```
"HOLA" > "Pep" és fals doncs 'H' < 'P' (segons codi ASCII)
"Joanna" <= "ANNA" és fals doncs 'J' > 'A' (segons codi ASCII)
"CASA" <= "CASAL" és cert
"CASA" <= "arbre" és cert doncs 'C' < 'a' (segons codi ASCII)
```

 L'operador de concatenació de cadenes és un operador binari (símbol
 +) que actua sobre dos operands de tipus cadena i obté com a resultat la concatenació d'ambdós operands.

Exemple:

```
var cadena1 = "Hola";
var cadena2 = "Món!!!"
var cadena3 = cadena1 + cadena2;
/* La variable cadena3 contindrà el valor "Hola Món!!!" */
```

3.4. Estructures repetitives

La majoria de llenguatges de programació aporten estructures per a efectuar processos repetitius. JavaScript proporciona tres estructures repetitives: els bucle for i dues versions de bucles while.

3.4.1. Bucle for

Els bucles for segueixen la següent sintaxi:

```
for (var variable=valor_inicial; condició; modificar_variable)
{
    seqüencia_instruccions_a_repetir
}
```

L'execució del programa segueix els següents passos:

1) Defineix la variable i la inicialitza amb un valor inicial. Si la variable ja existeix no es posa la partícula var. Es continua amb el 2n pas.

2) Comprova si condició és avaluada com a certa. Si és falsa es finalitza la seqüència repetitiva i es passa a la instrucció posterior al for. Si és certa, s'executa seqüència_instruccions_a_repetir i es passa al 3r pas.

3) S'executa modificar_variable si existeix (pot estar en blanc) i es torna al 2n pas.

Si seqüència_instruccions_a_repetir està formada per una única instrucció, les claus anterior ({) i posterior (}) es poden eliminar.

Els bucles for s'acostumen a utilitzar quan el procés s'ha de repetir un número determinat de vegades (comptatge) i, per tant, condició no fa altra cosa que comparar si s'ha arribat al límit i modificar_variable s'encarrega d'anar incrementant el valor de comptatge per tal d'aproparse al límit que provocarà que condició sigui falsa i acabi el procés. Però també es pot utilitzar per altres processos repetitius que no tinguin a veure amb un comptatge.

La condició pot consistir en una variable booleana o en una expressió que doni com a resultat un valor booleà. En qualsevol cas l'avaluació de condició només pot ser true o false.

Alerta! Cal assegurar que el procés repetitiu finalitzarà, és a dir, que els valors de les variables aniran canviant de manera que s'arribarà a algun moment en que condició sigui avaluada com a false.

<u>Exemple</u>: *Dissenyar un script que mostri els 100 primers números parells en una pàgina web.*

Solució 1:

<script language="JavaScript">
var anterior = 0;
for (var quantsEnPortem = 0; quantsEnPortem < 100; quantsEnPortem++)
{
 anterior = anterior + 2;
 document.writeln(anterior);
}
</script>

L'execució d'aquest codi provoca la visualització de la figura 68.



Comentaris:

- La variable anterior serveix per a saber quin és el darrer número parell generat i utilitzar-lo com a base per a calcular un nou parell (anterior = anterior + 2) que és el valor que visualitzem en el document amb el mètode writeln(). Per a que el primer valor parell generat sigui 2 ens cal inicialitzar anterior amb el valor zero.
- Aquest càlcul (anterior = anterior + 2) s'ha de repetir 100 vegades doncs ens demanen que mostrem els 100 primers números parells. Necessitem, per tant, una variable que ens serveixi de comptador (quantsEnPortem) i que la inicialitzem amb el valor zero.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina35.html* amb el codi de la solució 1.

- Fixem-nos que cada vegada que obtenim un nou parell, incrementem el seu valor en una unitat (quantsEnPortem++).
- Observem que la condició per a continuar el bucle (quantsEnPortem < 100) permet anar repetint el procés fins a 100 vegades. Cal anar amb compte amb el símbol de comparació, doncs si poséssim quantsEnPortem <= 100 resultaria que apareixerien 101 números parells, doncs en haver-ne obtingut 100 encara seria certa la condició i es calcularia un nou número parell.

Solució 2:

```
<script language="JavaScript">
var anterior = 0;
for (var quantsEnPortem = 0; quantsEnPortem < 100; )
{
    anterior = anterior + 2;
    document.writeln(anterior);
    quantsEnPortem++;
}
</pre>
```

Comentari:

Observem que aquesta solució és molt similar a l'anterior. Es diferencia únicament en que la tercera part de l'interior del parèntesi que acompanya a for està buida, modificant-se la variable quantsEnPortem dins seqüència_instruccions_a_repetir. No és recomanable utilitzar aquesta possibilitat; s'aconsella que la variable (o variables) que intervenen en condició (i que per tant provocaran en algun moment la finalització del bucle) es modifiquin en la tercera part de l'interior del parèntesi que acompanya a for per tal que sigui fàcilment identificable tot allò que té a veure amb la finalització del bucle.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina36.html* amb el codi de la solució 2.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina37.html* amb el codi de la solució 3.

Solució 3:

```
<script language="JavaScript">
for (var quantsEnPortem = 1; quantsEnPortem <= 100; quantsEnPortem++)
{
    document.writeln(quantsEnPortem*2);
}
</script>
```

Comentari:

• En aquesta solució ens aprofitem del fet que el primer parell és 2, el segon és 4, el tercer és 6,... és a dir, hi ha una correlació entre el número del parell (primer, segon, tercer,...) i el seu valor (2, 4, 6,...), adonant-nos que el valor parell de la posició n és 2*n. Per tant, no ens cal la variable anterior per a recordar quin era el darrer parell i

poder calcular-ne el següent sumant-ne 2, doncs podem calcular el parell a partir de la posició que ocupa multiplicant.-la per 2.

- Observem que el càlcul del número parell (quantsEnPortem*2) no ens cal guardar-lo en cap variable, sinó que directament podem utilitzar aquesta expressió com a paràmetre del mètode writeln().
- Fixem-nos que en aquesta solució, quantsEnPortem la inicialitzem amb l per a que el corresponent primer parell sigui 2. Llavors ens cal anar repetint el procés mentre quantsEnPortem <= 100.
- Com que el bloc repetitiu només conté una instrucció, haguéssim pogut estalviar-nos d'escriure les claus { i }.

3.4.2. Bucle while

Els bucles while segueixen la següent sintaxi:

```
while (condició)
{
    seqüencia_instruccions_a_repetir;
}
```

L'execució del programa comprova si condició és avaluada com a certa. Si és falsa es finalitza la seqüència repetitiva i es continua l'execució del programa amb la instrucció posterior al while. Si és certa, s'executa seqüència_instruccions_a_repetir i es torna a iniciar el procés de comprovació de condició.

Si seqüència_instruccions_a_repetir està formada per una única instrucció, les claus anterior ({) i posterior (}) es poden eliminar.

Alerta! Cal assegurar que el procés repetitiu finalitzarà, és a dir, que els valors de les variables aniran canviant de manera que s'arribarà a algun moment en que condició sigui avaluada com a false.

<u>Exemple</u>: Dissenyar un script que mostri els 100 primers números parells en una pàgina web.

Solució:

```
<script language="JavaScript">
var quantsEnPortem = 1;
while (quantsEnPortem <= 100)
{
    document.writeln(quantsEnPortem*2);
    quantsEnPortem++;
}
</script>
```

Fixem-nos que amb el while aconseguim la mateixa funcionalitat que amb el for.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina38.html* amb el codi de la solució.

3.4.3. Bucle do...while

Els bucles do ... while segueixen la següent sintaxi:

```
do
{
   seqüencia_instruccions_a_repetir;
}
while (condició);
```

La diferència fonamental amb el while és que en aquesta estructura s'assegura l'execució de seqüència_instruccions_a_repetir com a mínim l vegada.

El programa executa seqüència_instruccions_a_repetir i posteriorment avalua condició. Si és certa, es torna a iniciar el procés, el qual finalitza en arribar a una situació en la que condició és falsa.

Si seqüència_instruccions_a_repetir està formada per una única instrucció, les claus anterior ({) i posterior (}) es poden eliminar.

Alerta! Cal assegurar que el procés repetitiu finalitzarà, és a dir, que els valors de les variables aniran canviant de manera que s'arribarà a algun moment en que condició sigui avaluada com a false.

<u>Exemple</u>: *Dissenyar un script que mostri els 100 primers números parells en una pàgina web.*

Solució:

```
<script language="JavaScript">
var quantsEnPortem = 1;
do
{
    document.writeln(quantsEnPortem*2);
    quantsEnPortem++;
}
while (quantsEnPortem <= 100)
</script>
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina39.html* amb el codi de la solució.

3.4.4. Instruccions break i continue

De manera addicional a la utilització de les diferents estructures repetitives, hi ha dues instruccions per a:

- Aturar l'execució del bucle i sortir-ne: instrucció break
- Aturar la repetició actual i tornar al principi: instrucció continue

Instrucció break

Aquesta instrucció serveix per aturar l'execució del procés repetitiu i continuar amb la instrucció següent a l'estructura repetitiva.

Considerem l'exemple següent:

```
<script language="JavaScript">
var quantsEnPortem = 1;
do
{
    document.writeln(quantsEnPortem*2);
    break;
    quantsEnPortem++;
}
while (quantsEnPortem <= 100)
</script>
```

Malgrat que el bucle està pensat per a visualitzar els 100 primers números parells, fixem-nos que una vegada visualitzat el primer parell (el 2) hi ha un break, fet que provocarà la interrupció del bucle.

Instrucció continue

Aquesta instrucció serveix per a tornar al principi de l'estructura iterativa sense executar les instruccions que hi hagi dins l'estructura repetitiva per sota seu.

Considerem l'exemple següent:

```
<script language="JavaScript">
var quantsEnPortem = 1;
do
{
    document.writeln(quantsEnPortem*2);
    continue;
    quantsEnPortem++;
}
while (quantsEnPortem <= 100)
</script>
```

Malgrat que el bucle està pensat per a visualitzar els 100 primers números parells, fixem-nos que una vegada visualitzat el primer parell (el 2) hi ha un continue, fet que provocarà que les instruccions que hi ha per sota (increment de quantsEnPortem) no s'executin. El programa se'n va a avaluar la condició del while i si és certa (que en aquest cas ho serà) torna a executar el bloc repetitiu. Així doncs, aquest exemple provoca un bucle infinit, doncs anirà pintant 2 infinites vegades!

Consell referent a la utilització de break i continue

Les dues instruccions estan pensades per a trencar la normal evolució d'una estructura repetitiva i s'haurien d'utilitzar només per a situacions excepcionals que poden donar-se durant l'execució del bucle.

©Isidre Guixà i Miranda

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina40.html* amb el codi de l'exemple.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina41.html* amb el codi de l'exemple.

Setembre del 2007

126

Normalment no s'utilitzen com en els exemples precedents, sinó que formen part d'una estructura condicional (les veiem a continuació), és a dir, s'executaran si té lloc una determinada condició.

Les condicions normals de finalització de l'estructura repetitiva cal utilitzar-les adequadament a la condició del for, while o do...while.

3.5. Estructures condicionals

Els llenguatges de programació faciliten estructures per a poder prendre decisions dins l'execució d'un programa. JavaScript proporciona tres estructures condicionals: la instrucció if (simple i composta) i la instrucció switch.

3.5.1. Estructura condicional if simple

La seva sintaxis és:

```
if (condició)
{
    seqüència_instruccions_a_executar;
}
```

En arribar a la instrucció if, l'execució del programa comprova si condició és avaluada com a certa. En cas de ser certa s'executa seqüència_instruccions_a_executar. Si és falsa no s'executa el bloc seqüència_instruccions_a_executar.

Si seqüència_instruccions_a_executar està formada per una única instrucció, les claus anterior ({) i posterior (}) es poden eliminar.

<u>Exemple</u>: Dissenyar un script que mostri els 100 primers números parells en una pàgina web saltant línia cada 10 números. Solució:

```
<script language="JavaScript">
var quantsEnPortem=1;
do
{
    document.writeln(quantsEnPortem*2);
    if (quantsEnPortem % 10 == 0) document.writeln('<br>');
    quantsEnPortem++;
}
while (quantsEnPortem<=100);
</script>
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina42.html* amb el codi de la solució.

L'execució d'aquest script genera la imatge de la figura 69.

Comentaris:

• És clar que la solució d'aquest exemple necessita la utilització d'una estructura repetitiva i podríem haver-ne utilitzat qualsevol de les conegudes.

| Figura 69. Programa JavaScript que visualitza 100 números parells de 10 en 10. |
|--|
| 😻 Exemple d'utilització d'IF simple - Mozilla Firefox |
| Eitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda |
| 🖕 • 📄 • 🧭 🕜 🏠 🗋 file:///E:/pagina42.html 🔹 🕨 💽 • Google 🔍 |
| 🕒 Hotmail gratuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windows |
| 2 4 6 8 10 12 14 16 18 20 |
| 22 24 26 28 30 32 34 36 38 40 |
| 42 44 46 48 50 52 54 56 58 60 |
| 62 64 66 68 70 72 74 76 78 80 |
| 82 84 86 88 90 92 94 96 98 100 |
| 102 104 106 108 110 112 114 116 118 120 |
| 122 124 126 128 130 132 134 136 138 140 |
| 142 144 146 148 150 152 154 156 158 160 |
| 162 164 166 168 170 172 174 176 178 180 |
| 182 184 186 188 190 192 194 196 198 200 |
| |
| ret |

Figura 60, Programa, JavaSprint que visualitza 100

• És necessària, també, la utilització d'una estructura condicional per a detectar en quin moment s'ha visualitzat 10 números parells i poder provocar el salt de línia (inserint una marca
> en el codi HTML).

Com saber si justament portem visualitzats 10, 20, 30,... números parells? Ben segur que haguéssim pogut efectuar una pregunta similar a:

if (quantsEnPortem==10 || quantsEnPortem==20 || quantsEnPortem==30 ...)

És a dir, hauríem d'anar preguntant fins a 100... I si enlloc de demanar els 100 primers números parells en demanem els 10.000 primers? Evidentment el codi a escriure se'ns dispararia, no?

Per a saber quan un valor x és múltiple de n (en el nostre cas, n=10) només cal avaluar si x % n == 0 és certa (és a dir, si la divisió entera de x entre n dona exacte o, dit d'altra manera, dona residu zero). 🖽

• Per últim, com aconseguir un salt de línia? És clar que en el codi HTML el salt de línia el podem assolir amb la marca
, però ara ens trobem dins codi JavaScript i la marca
> no és coneguda per aquest llenguatge.

Tinguem present que el mètode writeln() de l'objecte document permet que JavaScript insereixi codi HTML a la pàgina. Així doncs, si volem que en el document HTML hi aparegui una marca
> només hem d'utilitzar el mètode writeln() passant-li com a paràmetre la cadena "
".

Exemple: Dissenyar un script que mostri els 100 primers números parells en una pàgina web dins una taula de 10 columnes.

Solució:

©Isidre Guixà i Miranda

```
<script language="JavaScript">
var quantsEnPortem=1;
document.writeln('');
do
{
    if (quantsEnPortem % 10 == 1) document.writeln('');
    document.writeln(""+quantsEnPortem*2+"");
    if (quantsEnPortem % 10 == 0) document.writeln('');
    quantsEnPortem++;
}
while (quantsEnPortem<=100);
document.writeln('</table>');
</script>
```

L'execució d'aquest script, que genera la imatge de la figura 70, ha d'anar construint la taula, és a dir, ha d'anar inserint les marques HTML:

- És clar que a l'inici ha d'inserir la marca i en finalitzar la marca .
- Cada vegada que comença una fila ha d'inserir la marca i en finalitzar la fila ha d'inserir la marca
 Ara bé, com saber quan comença i quan finalitza una fila? Doncs si pensem una mica ens adonem que la fila comença quan hem de visualitzar els parells que ocupen el lloc 1, 11, 21,... i que la fila finalitza quan hem visualitzat els parells que ocupen el lloc 10, 20, 30,...
 - La situació d'inici es detecta quan la condició quantsEnPortem % 10 == 1 és certa.
 - La situació de final es detecta quan la condició quantsEnPortem
 % 10 == 0 és certa.

| igura 70. Programa JavaScript que visualitza 100 parells en una taula de 10 columnes | | | | | | | | | | |
|--|--------------|------------|----------|------------|----------|---------|----------|----------|---------|---------------------|
| 🕹 Ex | emple | e d'ut | ilitza | nció d | l'IF si | imple | - Mo | zilla | Firef | ox 🔳 🗖 🔁 |
| <u>F</u> itxer | <u>E</u> dit | a <u>V</u> | isualitz | a <u>H</u> | istorial | Adj | reces | d'interi | ès Ei | nes Ajuda 🔇 |
| \ | • | - (| C | | | B | file:/// | E:/paç | jina43. | html 🔹 🕨 💽 Google 🔍 |
| 🕒 Но | itmail g | pratuito | • 🖻 | Persor | halizar | vínculo | os 🗋 | Wind | ows Me | adia 🗋 Windows |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | |
| 22 | 24 | 26 | 28 | 30 | 32 | 34 | 36 | 38 | 40 | |
| 42 | 44 | 46 | 48 | 50 | 52 | 54 | 56 | 58 | 60 | |
| 62 | 64 | 66 | 68 | 70 | 72 | 74 | 76 | 78 | 80 | |
| 82 | 84 | 86 | 88 | 90 | 92 | 94 | 96 | 98 | 100 | |
| 102 | 104 | 106 | 108 | 110 | 112 | 114 | 116 | 118 | 120 | |
| 122 | 124 | 126 | 128 | 130 | 132 | 134 | 136 | 138 | 140 | |
| 142 | 144 | 146 | 148 | 150 | 152 | 154 | 156 | 158 | 160 | |
| 162 | 164 | 166 | 168 | 170 | 172 | 174 | 176 | 178 | 180 | |
| 182 | 184 | 186 | 188 | 190 | 192 | 194 | 196 | 198 | 200 | |
| Fet | | | | | | | | | | |
| | _ | _ | | | | _ | _ | _ | | |

 Cada vegada que vol visualitzar un parell ha de crear la corresponent cel·la amb la marca , visualitzar el parell i tancar la cel·la amb la marca

 Els tres passos es poden fer amb tres crides al mètode writeln() però en l'exemple s'ha fet en una sola crida passant com A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina43.html* amb el codi de la solució. a paràmetre el resultat de la concatenació dels tres textos: ""+quantsEnPortem+"".

3.5.2. Estructura condicional if composta

La seva sintaxis és:

```
if (condició)
{
    seqüència_instruccions_a_executar_A;
}
else
{
    seqüència_instruccions_a_executar_B;
}
```

En arribar a la instrucció if, l'execució del programa comprova si condició és avaluada com a certa o com a falsa. En cas de ser certa s'executa seqüència_instruccions_a_executar_A. Si és falsa s'executa seqüència_instruccions_a_executar_B.

Si seqüència_instruccions_a_executar està formada per una única instrucció, les claus anterior ({) i posterior (}) es poden eliminar.

Exemple: Dissenyar un script que mostri els 100 primers números enters de manera que els senars apareguin en color blau i els parells en color vermell.

Solució:

```
<script language="JavaScript">
for (var nombre=1; nombre<=100; nombre++)
{
    if (nombre % 2 == 1) document.writeln('<font style="color:blue;">');
    else document.writeln('<font style="color:red;">');
    document.writeln(nombre+"</font>");
}
</script>
```

Comentaris:

- És clar que hi ha d'haver un bucle per a construir els 100 números a visualitzar.
- Per cada número a visualitzar s'ha d'activar un color o un altre en funció de si és senar o parell. Per a detectar-ho només cal veure quin és el residu de la divisió entera per 2 (si és 0 vol dir que el número és parell i si és 1 vol dir que el número és senar). En funció de quina situació es doni activem el color que correspongui i per a fer-ho utilitzem una instrucció if ... else.
- Observem que abans de visualitzar cada número activem el color que correspongui amb la marca i, per a generar codi HTML

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina44.html* amb el codi de la solució.

correcte cal, després de visualitzar cada número, tancar-la amb .

3.5.3. Estructura condicional switch

La seva sintaxis és:

```
switch (expressió)
{
   case valor1: seqüència_instruccions_a_executar_1;
        break;
   case valor2: seqüència_instruccions_a_executar_2;
        break;
   ...
   default: seqüència_instruccions_a_executar_1;
        break;
}
```

Aquesta estructura s'utilitza quan tenim múltiples possibilitats com a resultat de l'avaluació d'una expressió.

El seu funcionament és simple. S'avalua expressió. Si el resultat coincideix amb valor1, s'executa la seqüència d'instruccions corresponents; si el resultat coincideix amb valor2, s'executa la seqüència d'instruccions relacionades amb aquest valor i així successivament per a les diverses opcions case definides. Finalment, si el resultat no coincideix amb cap de les opcions case contemplades, s'executa l'opció per defecte default si existeix (és opcional).

La instrucció break és necessària per a sortir de l'estructura switch. Si en un case no s'utilitza, el programa continuarà amb l'execució de les instruccions que hi hagi a continuació encara que siguin d'altres case, fins que trobi un break o fins que finalitzi el switch.

<u>Exemple</u>: Dissenyar un script que generi un número enter aleatori entre 1 i 12 i ens digui el nom del corresponent mes (gener, febrer,...).

Per a l'execució d'aquest exemple necessitem saber com generar un número aleatori. JavaScript té l'objecte Math per a efectuar càlculs matemàtics el qual conté el mètode random() per a obtenir un número decimal aleatori entre 0 i l (exactament >=0 i <1). Per tant, si el multipliquem per 12 obtindrem un número decimal aleatori >=0 i <12. Però ens interessa tenir valors enters i per aconseguir-ho, l'objecte Math ens proporciona el mètode floor() que descarta els decimals. Així doncs, aplicant aquest mètode al número >=0 i <12 obtenim un número enter entre 0 i ll i si desitjàvem un número enter entre l i l2 només ens caldrà incrementar-lo en una unitat.

Solució:

```
<script language="JavaScript">
var nombre, mes;
nombre = Math.random() * 12;
nombre = Math.floor(nombre) + 1;
switch (nombre)
  case 1: mes="Gener"; break;
  case 2: mes="Febrer"; break;
  case 2: mes "Març"; break;
case 4: mes="Abril"; break;
  case 5: mes="Maig"; break;
  case 6: mes="Juny"; break;
  case 7: mes="Juliol"; break;
case 8: mes="Agost"; break;
  case 9: mes="Setembre"; break;
  case 10: mes="Octubre"; break;
case 11: mes="Novembre"; break;
case 12: mes="Desembre"; break;
document.writeln("S'ha generat aleatòriament el valor "
+nombre+" que correspon al mes "+mes);
</script>
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina45.html* amb el codi de la solució.

<u>Exemple</u>: Dissenyar un script que generi un número enter aleatori entre 1 i 12 i ens digui el nom del corresponent mes i dels que el segueixen fins a final d'any.

Solució:

```
<script language="JavaScript">
var nombre:
nombre = Math.random() * 12;
nombre = Math.floor(nombre) + 1;
document.writeln("S'ha generat aleatòriament el valor "
+nombre+" i els mesos a partir d'ell fins a final d'any
 són:");
switch (nombre)
  case 1: document.writeln("Gener,");
  case 2: document.writeln("Febrer,");
case 3: document.writeln("Març,");
  case 4: document.writeln("Abril,");
  case 5: document.writeln("Maig,");
  case 6: document.writeln("Juny,");
  case 7: document.writeln("Juliol,");
  case 8: document.writeln("Agost,");
  case 9: document.writeln("Setembre,");
  case 10: document.writeln("Octubre,");
  case 11: document.writeln("Novembre,"
  case 12: document.writeln("Desembre.");
</script>
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina46.html* amb el codi de la solució.

Observem que en aquest cas ens va de conya no utilitzar la instrucció break, doncs d'aquesta manera aconseguim que es visualitzi el mes corresponent al valor aleatori generat seguit de la resta de mesos fins a final d'any.

3.6. Funcions

En l'elaboració de programes ens trobem amb determinats processos (conjunts d'instruccions) que poden ser pensats de forma independent al programa i que són de fàcil resolució pensant-los de manera independent

132

al programa sencer. A més, en ocasions, aquests processos pot ser que sigui necessari executar-los més d'una ocasió en un mateix programa o, fins i tot, en diferents programes.

Imaginem-nos, per exemple, el fet de fer aparèixer a totes les nostres pàgines web la informació corresponent a la darrera actualització de la pàgina. Oi que el procés d'obtenir aquesta data (no ens preocupa en aquests moments com fer-ho) deu ser el mateix sigui quina sigui la pàgina web en la que ens trobem?

Les funcions permeten agrupar el conjunt d'instruccions que constitueixen un procés en un únic lloc i sota un nom que l'identifiqui, de manera que se'n pugui fer referència (cridar-lo) allà on calgui executar-lo.

Altres mots que denoten mecanismes similars són: subprogrames, procediments i accions.

En programar utilitzem funcions constantment, ja sigui les escrites per nosaltres, ja sigui les definides en el sistema, doncs tots els llenguatges de programació aporten un munt de funcions per a executar processos habituals, com per exemple, obtenir l'hora, imprimir un missatge en pantalla o convertir variables d'un tipus a un altre.

Ja hem utilitzat alguna funció de les que facilita JavaScript:

- write(), que és una funció associada al document de la pàgina que permet inserir un text a la pàgina.
- random(), que és una funció que permet generar un número decimal aleatori entre 0 i 1.

3.6.1. Com escriure, situar i cridar una funció?

Una funció s'ha de definir seguint la sintaxis:

```
function nom_de_la_funció ()
{
    seqüència_d'instruccions_a_executar;
}
```

El nom de la funció segueix les mateixes normes que els noms de les variables.

En principi podem situar les funcions en qualsevol part de la pàgina, sempre entre etiquetes <script> i </script>. Hi ha, però, una limitació a

©Isidre Guixà i Miranda

l'hora de situar-la: cal que estigui present abans que se n'efectuï qualsevol crida. (I)

Un lloc molt recomanable per situar-les és a la capçalera doncs es suposa que a la capçalera no s'utilitzaran encara i sempre podrem utilitzar-les en qualsevol lloc de la pàgina.

Per executar una funció (i per tant totes les instruccions que conté) l'hem de cridar utilitzant el seu nom seguit dels parèntesis.

<u>Exemple</u>: Dissenyar una pàgina web que contingui una funció que escrigui un número aleatori entre 11 i 20 i se'n faci diverses crides.

Solució:

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina47.html* amb el codi de la solució.

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"> <html> <head> <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type"> <title>Exemple de crida a FUNCIÓ</title> <script> function nombreAleatori () var nombre = Math.random() * 10; // Nombre decimal aleatori >=0 i <10</pre> nombre = Math.floor(nombre); // Nombre enter >=0 i <10</pre> nombre = nombre + 11; // Nombre enter entre >=11 i <=20 document.writeln(nombre); } </script> </head> <body> 1a. crida a la funció nombreAleatori: <script language="JavaScript"> nombreAleatori(); </script>
 2a. crida a la funció nombreAleatori: <script language="JavaScript"> nombreAleatori(); </script>
 3a. crida a la funció nombreAleatori: <script language="JavaScript"> nombreAleatori(); </script> </bodv> </html>

3.6.2. Utilització de paràmetres

Les vegades que hem utilitzat el mètode writeln() hem introduït entre els seus parèntesis el text que volíem inserir a la pàgina web. Aquest text és un valor que comuniquem al mètode writeln() per a que aquest sàpiga què és el que ha d'escriure.

La funció nombreAleatori desenvolupada en el darrer exemple serveix per al càlcul d'un número enter aleatori entre 11 i 20. I si volem un número aleatori entre 25 i 30? Haurem de crear una nova funció? Seria molt interessant poder comunicar a la funció els valors inferior i superior entre els que volem generar un número aleatori i, per tant, la funció ens serviria per a moltes ocasions. Això ho podem aconseguir afegint paràmetres a la funció.

Els paràmetres de les funcions s'utilitzen per a passar valors a l'interior de la funció per a que puguin ser utilitzats en les accions a desenvolupar.

Els paràmetres de les funció també s'anomenen valors d'entrada de la funció. (II)

Una funció pot rebre tants paràmetres com sigui menester i per a indicar-ho es situen separats per comes entre els parèntesis que acompanyen el nom de la funció.

La crida d'una funció amb paràmetres ha de contenir, dins els parèntesis, tants valors o variables o expressions com paràmetres tingui la definició de la funció. (II)

Cada paràmetre d'una funció és una variable que existeix dins la funció i que s'inicialitza amb el corresponent valor indicat en la crida. La correspondència s'efectua per posició. És a dir, suposem que tenim la funció de nom X amb tres paràmetres:

```
function X (par1, par2, par3)
{ ... }
```

Les crides a la funció X hauran de ser del tipus:

X (v1, v2, v3);

on v1, v2 i v3 poden ser valors o variables o expressions amb 1 resultat i, en cada crida de la funció x es creen tres variables par1, par2 i par3, inicialitzades respectivament pels valors que representen v1, v2 i v3.

<u>Exemple</u>: Dissenyar una pàgina web que contingui una funció que sàpiga escriure un número aleatori entre dos valors indicats i se'n faci diverses crides.

Solució:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
<meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
<title>Exemple de crida a FUNCI&Oacute; amb PAR&Agrave;METRES</title>
<script>
function nombreAleatori (inf, sup)
```

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina48.html* amb el codi de la solució.

```
{
        /* Aquesta funció tindrà bon funcionament si inf<=sup
           Per tant, per si és el cas que algun programador la utilitzi amb valors canviats,
           fem la corresponent comprovació i girem els valors si cal*/
        if (inf>sup)
                        // Necessitem una variable auxiliar per a intercanviar els valors
        { var aux=inf;
          inf = sup;
          sup = aux;
        // Per a fer un seguiment del codi suposem que inf=11 i sup=20
        var nombre = Math.random() * (sup-inf+1); // Nombre decimal aleatori >=0 i <10
        nombre = Math.floor(nombre); // Nombre enter >=0 i <10</pre>
        nombre = nombre + inf; // Nombre enter entre >=inf i <=sup
        document.writeln(nombre);
  </script>
</head>
<bodv>
Crida a la funció nombreAleatori(15.20):
<script language="JavaScript">
nombreAleatori(15,20);
</script><br>
Crida a la funció nombreAleatori(20,45):
<script language="JavaScript">
nombreAleatori(20,45);
</script><br>
Crida a la funció nombreAleatori(30,10):
<script language="JavaScript">
nombreAleatori(30,10);
</script>
</body>
</html>
```

En l'exemple hem efectuat tres crides a la funció nombreAleatori() indicant en cadascuna d'elles els valors inferior i superior entre els que cal cercar el número. En les tres crides els valors han estat constants ((15,20), (20,45) i (30,10)) però haguessin pogut estar variables o expressions de resultat enter.

Els paràmetres de les funcions en JavaScript es passen per valor, cosa que vol dir que si el paràmetre és una variable i en l'interior de la funció se'n modifica el valor, en finalitzar l'execució de la funció la variable tenia el mateix valor que en el moment d'iniciar-se l'execució de la funció.

3.6.3. Valors de retorn

En ocasions ens interessarà que una funció, a banda d'efectuar accions, doni un resultat final.

Per especificar el valor resultat d'una funció, s'utilitza la paraula reservada return seguida del valor (constant o variable o expressió) resultat.

La recepció del resultat d'una funció es pot guardar dins una variable mitjançant l'operador d'assignació = o directament utilitzar-lo en una expressió o com a paràmetre d'una crida a una altra funció. (I)

Així, per exemple, en la instrucció

nombre = Math.random() * (sup-inf+1);

observem que la funció random() retorna un número que utilitzem directament com a primer operand del producte amb (sup-inf+1).

Un altre exemple d'utilitzar el valor de retorn el veiem en la següent següència de crides:

```
nombre = Math.random() * (sup-inf+1);
nombre = Math.floor(nombre);
nombre = nombre + inf;
document.writeln(nombre);
```

que es podria resumir en:

```
nombre = Math.floor(Math.random() * (sup-inf+1));
nombre = nombre + inf;
document.writeln(nombre);
```

o encara més, en:

```
nombre = Math.floor(Math.random() * (sup-inf+1))+inf;
document.writeln(nombre);
```

o ja, simplificant al màxim:

document.writeln(Math.floor(Math.random() * (sup-inf+1))+inf);

i ni tan sols ha estat necessària la utilització de la variable nombre. A vegades, però, tanta simplificació de codi porta a fer difícil un seguiment de les accions que s'executen.

<u>Exemple</u>: Dissenyar una pàgina web que contingui una funció que retorni un número aleatori entre dos valors indicats i se'n faci diverses crides.

Solució:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de retorn d'una FUNCI&Oacute; amb PAR&Agrave;METRES</title>
  <script>
    function nombreAleatori (inf, sup)
        /* Aquesta funció tindrà bon funcionament si inf<=sup
           Per tant, per si és el cas que algun programador la utilitzi amb valors canviats,
           fem la corresponent comprovació i girem els valors si cal*/
        if (inf>sup)
        { var aux=inf;
                        // Necessitem una variable auxiliar per a intercanviar els valors
          inf = sup;
          sup = aux;
        // Per a fer un seguiment del codi suposem que inf=11 i sup=20
        var nombre = Math.random() * (sup-inf+1); // Número decimal aleatori >=0 i <10</pre>
```

A la secció "Recursos" del web d'aquest crèdit podeu

trobar el fitxer *pagina49.html* amb el codi de la solució.

138

```
nombre = Math.floor(nombre); // Número enter >=0 i <10</pre>
        nombre = nombre + inf; // Número enter entre >=inf i <=sup</pre>
        return nombre;
    }
  </script>
</head>
<body>
Crida a la funció nombreAleatori(15,20):
<script language="JavaScript">
  document.writeln(nombreAleatori(15,20));
</script>
<br>
Crida a la funció nombreAleatori(20,45):
<script language="JavaScript">
  document.writeln(nombreAleatori(20,45));
</script>
<br>
Crida a la funció nombreAleatori(30,10):
<script language="JavaScript">
  document.writeln(nombreAleatori(30,10));
</script>
</body>
</html>
```

En una mateixa funció es pot col·locar més d'un return. Lògicament només retornarà una cosa, però depenent del que hagi succeït dins la funció, podrà ser d'un tipus o un altre, amb uns valors o uns altres.

Tipus de dada de retorn en una funció

La majoria de llenguatges que proporcionen *funcions*, obliguen a que la funció retorni un determinat tipus de valor i cal indicar-lo en la seva declaració. JavaScript no és tant restrictiu.

3.7. Àmbit de les variables. Variables locals i globals.

S'anomena àmbit de les variables al lloc on aquestes estan disponibles.

En la majoria de llenguatges de programació, quan es declara una variable, aquesta està disponible en el lloc on s'ha declarat. Com que en JavaScript es declaren en una pàgina, les variables declarades tenen com a àmbit d'actuació tota la pàgina i són accessibles dins d'ella. Així doncs, no es pot accedir a variables que hagin estat declarades en una altra pàgina.

S'anomena variables globals d'una pàgina, les variables definides dins qualsevol script de la pàgina i són accessibles des de qualsevol punt de la pàgina.

JavaScript, però, permet definir variables locals dins les funcions.

S'anomena variables locals d'una funció les variables declarades amb la paraula var dins la funció, les quals són accessibles només dins la funció on han estat declarades. Els paràmetres de les funcions són variables locals de la funció.

No hi ha problema en declarar una variable local amb el mateix nom que una global; en aquest cas la variable global serà visible des de tota la pàgina excepte en l'àmbit on estigui declarada la variable local, ja que en aquesta ubicació el nom de la variable global està ocupat per la local i és aquesta la que té validesa.

Les variables utilitzades dins una funció sense ser declarades amb la paraula var són globals de la pàgina. (I)

Considerem el codi següent corresponent a la pàgina de la figura 71.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de variables globals i locals d'una funci&oacute;</title>
  <script>
    function X ()
    {
       var num=10;
       document.writeln("num = "+num+"<br>");
    function Y ()
       num=20;
       document.writeln("num = "+num+"<br>");
  </script>
</head>
<bodv>
<script language="JavaScript">
  num=5:
  document.writeln("Punt 1. num = "+num+"<br>");
  document.writeln("Punt 3. num = "+num+"<br>");
document.writeln("Punt 4.");Y();
document.writeln("Punt 5. num = "+num+"<br>");
</script>
<script>
  document.writeln("Punt 6. num = "+num+"<br>");
   num=30;
  document.writeln("Punt 7. num = "+num+"<br>");
</script>
</body>
```

Fem el seguiment de l'execució dels scripts de l'anterior codi:

- Punt 1. La variable num és global a la pàgina i té 5 per valor.
- Punt 2. La variable num de la funció X és local a la funció i val 10.
- Punt 3. La variable num torna a ser la global a la pàgina i val 5.
 Queda clar que la variable num de la funció X era local a la funció.
- Punt 4. La variable num de la funció Y és global (doncs no està declarada amb la paraula var). El seu valor és 20.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer pagina50.html

amb el codi de la solució.

</html>

- Punt 5. La variable num global val 20. Queda demostrat que la variable num de la funció Y era la global de la pàgina.
- Punt 6. S'ha sortit de l'script anterior i s'ha iniciat un nou script.
 La variable num continua valent 20. Queda comprovat que la variable num és global a tota la pàgina.
- Punt 7. El valor de la variable global num ha estat modificat per 30.

| Figura 71. Comprovació de l'àmbit de les variables en el codi del document pagina50 | .htm |
|---|------|
| 🕹 Exemple de variables globals i locals d'una funció - Mozilla Firefox 🛛 🔲 🔲 | × |
| Ejitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda | 0 |
| | 2 |
| 🕒 Hotmail gratuito 📄 Personalizar vínculos 📄 Windows Media 📄 Windows | |
| Punt 1. num = 5 | ^ |
| Punt 2. num = 10 | |
| Punt 3. num = 5 | |
| Punt 4. num = 20 | = |
| Punt 5. num = 20 | |
| Punt 6. num $= 20$ | |
| Punt 7. num = 30 | - |
| Fet | |

3.8. Esdeveniments

Cada cop que fem un clic amb el ratolí , passem el ratolí per sobre d'un objecte, canviem la grandària d'una finestra, es tanca una finestra,... s'està provocant un esdeveniment.

Un esdeveniment és qualsevol fet que té lloc durant l'execució d'un programa.

La majoria dels esdeveniments que es produeixen no tenen cap interès en el funcionament del programa i són ignorats. Però en algunes situacions ens pot interessar que el programa faci algunes accions específiques quan es produeix un determinat esdeveniment. En aquests casos, hem de detectar-lo i fer que el programa actuï en conseqüència.

La taula 21 mostra una llista dels esdeveniments més habituals de detectar en les pàgines web amb JavaScript.

Taula 21. Llista d'esdeveniments més habituals de controlar en JavaScript.

| Esdeveniment | Acció que el provoca |
|--------------|--|
| blur | Quan un element de formulari perd el focus, per exemple, quan es fa clic en un altre lloc diferents d'aquest element |
| click | Quan es fa un clic amb el ratolí damunt un element |

| Esdeveniment | Acció que el provoca | |
|--------------|--|--|
| change | Quan l'usuari canvia el valor d'un camp del formulari | |
| focus | Quan un element rep el focus | |
| load | Quan es carrega una pàgina en el navegador | |
| mouseOver | Quan l'usuari mou el cursor per sobre d'un vincle hipertextual | |
| submit | Quan es va clic damunt el botó per a enviar les dades d'un formulari | |
| unLoad | Quan l'usuari abandona una pàgina | |

Els controladors d'esdeveniments s'utilitzen per a definir els esdeveniments objecte de detecció.

Els controladors d'esdeveniments tenen una sintaxis com la següent:

onXXXX = codi_d'accions_a_realitzar;

on les lletres XXXX indiquen el tipus d'esdeveniment a detectar i el codi indicat a la dreta del símbol = és l'acció a realitzar quan tingui lloc aquest esdeveniment.

Així, els controladors d'esdeveniments corresponents als esdeveniments de la taula 21 són: onBlur, onClick, onChange, onFocus, onLoad, onMouseOver, onSubmit i onUnLoad.

Aquests controladors s'apliquen dins les marques HTML que els poden suportar (<body>, , ,...) amb l'estructura indicada.

Si el codi que ha d'executar un controlador d'esdeveniments és molt extens, no s'acostuma a incloure a la dreta del signe =. Habitualment, se sol definir una funció que inclou tot el codi que ha d'executar l'esdeveniment i solament s'assigna el nom de la funció al controlador d'esdeveniments.

S'ha de tenir en compte que si una marca HTML que incorpora un controlador d'esdeveniment és inclosa dins una altra marca que també incorpora un altre controlador d'esdeveniment, la marca inclosa hereta també l'esdeveniment de la marca que l'inclou. Aquest efecte s'anomena **bombolleig d'esdeveniments**.

Considerem el següent codi que ens servirà per a exemplificar els conceptes introduïts.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina51.html* amb el codi de l'exemple.

<meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"> <html> <head>

```
<title>Exemple de control d'esdeveniments</title>
</head>
<body onLoad="alert('S\'ha carregat la pàgina.');"
onUnload="alert('Adéu.');">
  Text 1.
   Text 2.
   <br>
<center>
<form onSubmit="alert('Ho sento, aquest formulari no pot enviar res.'); return false;">
    Quadre 1: <input type="text" onChange="alert('Ha canviat el contingut del Quadre 1.');"><br>
    Quadre 2: <input type="text" onBlur="alert('El Quadre 2 ha perdut el focus.');"><br>

</form>
</center>
</body>
</html>
```

Comentaris:

- En primer lloc comentem la utilització, en aquest codi, del mètode alert() que permet enviar un missatge (passat per paràmetre) a l'usuari de la pàgina web, el qual haurà d'acceptar per a poder continuar amb la càrrega i/o visualització de la pàgina.
- La marca <body> conté els esdeveniments onLoad i onUnLoad.
 - L'esdeveniment onLoad provocarà que aparegui el missatge
 "S'ha carregat la pàgina" cada vegada que la pàgina es carregui.
 - L'esdeveniment onUnLoad provocarà que en abandonar la pàgina aparegui el missatge "Adéu".
- La primera fila de la taula té definit l'esdeveniment onClick a la marca . Això vol dir que aquest esdeveniment s'executarà cada vegada que es produeixi un clic dins aquesta fila.
- Dins la primera fila hi ha una marca que també té definit l'esdeveniment onClick. Per tant, quan s'efectuï un clic dins la marca , es detectaran dos esdeveniments onClick: el definit directament a la marca i l'heretat de la marca (és l'efecte "bombolleig d'esdeveniments" que comentàvem més amunt). L'ordre d'execució és de més intern (marca) cap a més extern (marca).
- La segona fila de la taula té definit l'esdeveniment onMouseOver que es dispara cada vegada que el ratolí passi per damunt la segona fila de la taula.

- En el formulari hi ha definit un esdeveniment onSubmit que es dispararà quan l'usuari premi el botó de tipus submit que incorpora el formulari. A més, com que no volem que les dades del formulari s'arribin a enviar, afegim al missatge d'alerta la instrucció return false; que provoca l'avortament de l'envio.
- El quadre de text "Quadre 1" té definit l'esdeveniment onChange que s'activarà cada vegada que l'usuari en canviï el contingut.
- El quadre de text "Quadre 2" té definit l'esdeveniment onBlur que s'activarà cada vegada que el quadre de text perdi el focus.

3.9. Objectes i mètodes

Un objecte és una estructura que conté dades i les funcions (anomenades mètodes) per manipular dites dades.

Aquest concepte és la base de la programació orientada a objectes, tècnica de programació que inclouen la majoria de llenguatges de programació actuals.

Les dades dels objectes poden ser, a la seva vegada, altres objectes (que tindran dades i les corresponents funcions per a gestionar-les) o valors finals que s'acostumen a anomenar propietats.

JavaScript incorpora un seguit d'objectes que ens poden ser de molta utilitat en el desenvolupament de les nostres pàgines web. Algun d'ells ja l'hem utilitzat, com és l'objecte Math amb els mètodes random() i floor().

Alguns dels objectes que JavaScript ens proporciona són únicament per a proporcionar-nos funcionalitats concretes (càlculs), com l'objecte Math. En canvi, altres objectes, com Date són pensats per a que en puguem definir variables per a emmagatzemar-hi valors.

3.9.1. Objecte Date

Aquest objecte està pensat per a gestionar dates i hores. S'invoca de qualsevol de les tres formes següents:

```
var data = new Date();
var data = new Date(any, mes, dia);
var data = new Date(any, mes, dia, hora, minut, segon);
```

Si no utilitzem cap paràmetre, l'objecte data contindrà la data i hora actuals (del rellotge del sistema). En cas contrari cal tenir en compte que els mesos es compten a partir de zero.

La taula 22 mostra alguns dels mètodes que facilita l'objecte Date.

| Taula 22. Mètodes a utilitzar en la gestió d'objectes Date de | JavaScript. | t. |
|---|-------------|----|
|---|-------------|----|

| Mètodes | | Explicació | |
|---------------|---------------------|--|--|
| getTime() | setTime(milisegons) | Obté i posa, respectivament, el moment temporal (hora-minut-segons) comptat en milisegons transcorreguts des de l'1de gener del 1970 | |
| getYear() | setYear(any) | Obté i posa, respectivament, l'any de la data. Segons quin any el poden retornar només en dues xifres. | |
| getFullYear() | setFullYear() | Obté i posa, respectivament, l'any de la data amb 4 xifres (aconsellable!) | |
| getMonth() | setMonth(mes) | Obté i posa, respectivament, el mes de la data (comptant a partir de zero) | |
| getDate() | setDate(dia) | Obté i posa, respectivament, el dia de la data | |
| getHours() | setHours(hores) | Obté i posa, respectivament, l'hora de la data (sense tocar minuts ni segons) | |
| getMinutes() | setMinutes(minuts) | Obté i posa, respectivament, els minuts de la data (sense tocar hores ni segons) | |
| getSeconds() | setSeconds(segons) | Obté i posa, respectivament, els segons de la data (sense tocar hores ni minuts) | |
| getDay() | | Obté el dia de la setmana de la data en forma de número que va de 0 (diumenge) a 6 (dissabte) | |

3.9.2. Objecte Math

L'objecte Math no està construït per a tenir variables de tipus Math, sinó com un contenidor on ficar diverses constants matemàtiques com Math.PI i mètodes que faciliten càlculs matemàtics com els recollits a la taula 23.

| Mètode | Significat | Expressió d'exemple |
|----------------------------|---|--------------------------------|
| abs(num) | Valor absolut de num | Math.abs(-2) retorna 2 |
| ceil(num) | Retorna l'enter més petit que sigui major o igual | Math.ceil(-2.7) retorna -2 |
| | que num | Math.ceil(2.7) retorna 3 |
| Floor (num) | Retorna l'enter més gran que sigui menor o igual | Math.floor(3.7) retorna 3 |
| r 1001 (IIulii) | que num | Math.floor(-3.7) retorna -4 |
| Dound (num) | Deterne l'enter més prener e pum | Math.round(3.7) retorna 4 |
| Rouna (num) | Retorna i enter mes proper a num | Math.round(-3.7) retorna -4 |
| random() | Retorna un número decimal aleatori major o igual a zero i menor a 1 | |
| min(num1, num2) | Retorna el valor menor de num1 i num2 | Math.min(2.4, 3.7) retorna 2.4 |
| <pre>max(num1, num2)</pre> | Retorna el valor major de num1 i num2 | Math.max(2.4, 3.7) retorna 3.7 |
| pow(base, exp) | Retorna la potència de base elevat a exp | Math.pow(2,3) retorna 8 |
| sqrt(num) | Retorna l'arrel quadrada de num | Math.sqrt(16) retorna 4 |

Taula 23. Mètodes que facilita l'objecte Math de JavaScript.
3.10. El Model d'Objectes del Document (DOM)

El DOM, acrònim anglès de **Document Object Model** (Model d'Objectes del Document), és una forma de representar tots els elements d'un document estructurat (per exemple, una pàgina web HTML o un document XML) com objectes que tenen els seus propis mètodes i propietats.

La guerra entre navegadors que va tenir lloc en els anys 90 entre *Netscape Navigator* i *Internet Explorer* de *Microsoft* va crear greus problemes per als programadors de pàgines web, ja que malgrat que ambdós navegadors utilitzaven JavaScript com a llenguatge de programació, els objectes no es comportaven de la mateixa manera (el DOM no era únic). Així doncs, el *W3C* va decidir crear un model d'objectes únic per a que tots els fabricants el poguessin adoptar facilitant la compatibilitat plena.

El DOM proporciona una interfície per accedir, afegir i canviar dinàmicament el contingut estructurat en documents amb llenguatges com JavaScript. (I)

La figura 72 proporciona una introducció a l'estructura del Model d'Objectes del Document. El DOM, juntament amb la gestió d'esdeveniments, proporciona la interfície que permet als scripts comunicar-se amb el navegador.

Figura 72. Introducció a l'estructura del DOM per als documents HTML



L'objecte més alt de la jerarquia és window. Conté tot el que hi ha a la finestra. Quan ens referim a la pantalla, navegador, marcs, adreça o historial, ho hem de fer a partir de l'objecte window.

Un altre objecte important és document que conté tots els elements de la pàgina: cos, paràgrafs, enllaços, formularis, imatges, estils, scripts, ...

Ens podem referir a qualsevol propietat d'un objecte de la jerarquia utilitzant la sintaxi del punt, que ja hem emprat moltes vegades en utilitzar el mètode document.writeln().

<u>Exemple</u>: Dissenyar un script que mostri el nom i versió del navegador que està visualitzant la pàgina.

```
<script>
document.writeln("Navegador: "+navigator.appName+"<br>")
document.writeln("Navegador: "+navigator.appVersion);
</script>
```

En aquest script hem utilitzat el mètode writeln() de l'objecte document inclòs a la finestra actual i les propietats appName i appVersion de l'objecte navigator de la finestra actual.

Observem que per referir-nos a la finestra actual no cal indicar res. En fer referència als objectes document i navigator, JavaScript ja sap que ens referim a la finestra actual. Però si volem, podem utilitzar el nom window o self per a referir-nos-hi i, per tant, el codi anterior podria ser:

```
<script>
window.document.writeln("Navegador: "+window.navigator.appName+"<br>")
window.document.writeln("Navegador: "+window.navigator.appVersion);
</script>
```

En aquest darrer exemple queda força clara la jerarquia que cal recórrer, amb la sintaxi del punt, per arribar des de l'objecte situat més amunt a la jerarquia (window) fins a les propietats i/o mètodes d'objectes més interns.

3.11. Acció de JavaScript en formularis

Els formularis HTML tenen la possibilitat d'incloure, a més dels botons de tipus submit i reset que provoquen, respectivament, l'enviament i la neteja de les dades del formulari, altres botons, als quals se'ls ha de poder associar una acció (o conjunt d'accions). Per aconseguir-ho necessitem utilitzar JavaScript.

Així mateix, és possible que ens interessi actuar davant l'acció de l'usuari sobre qualsevol objecte del formulari (camps de text, botons de selecció, caselles de verificació,...) i això també es pot assolir amb JavaScript.

La manera més efectiva, però no única, de tractar el contingut del formulari, és interceptar l'esdeveniment onClick.

EI DOM a fons

Conèixer tots els objectes del DOM per a JavaScript, amb les seves dades, propietats i mètodes és una tasca molt feixuga que s'escapa de l'àmbit d'aquest crèdit. Exemple: Dissenyar un formulari que tingui un camp de text (de només lectura), un botó, una casella de selecció i uns botons de selecció i aconseguir que cada vegada que l'usuari premi el botó o activi un botó de selecció o la casella de selecció, aparegui un missatge explicatiu de l'acció en el camp de text.

Solució:

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina52.html* amb el codi de la solució.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple d'utilitzaci&oacute; de JavaScript en Formularis</title>
</head>
<bodv>
onClick="missatge.value='S\'ha premut el botó'
                                                          "><br>
  Casella: <input name="casella" type="checkbox"
  onClick="missatge.value='S\'ha premut la casella'"><br/>opci&oacute; 1 <input name="grup" value="bgl" type="radio"
  onClick="missatge.value='S\'ha premut 1\'opció 1'"><br>
Opci&oacute; 2 <input name="grup" value="bg2" type="radio"
 onClick="missatge.value='S\'ha premut l\'opció 2'"><br>
Opci&oacute; 3 <input name="grup" value="bg3" type="radio"
    onClick="missatge.value='S\'ha premut l\'opció 3'"><br>
  <input readonly="readonly" name="missatge"><br>
</form>
</bodv>
</html>
```

Observem que damunt cada objecte en el que hem d'interceptar l'acció de l'usuari hem definit l'esdeveniment onClick acompanyat de l'acció a realitzar, que en aquest cas consistia en modificar el valor del camp de text (propietat value) amb un text concret.

<u>Exemple</u>: Dissenyar una pàgina que tingui dos botons: un per a obrir la pàgina del darrer exemple en una altra finestra i l'altre per a tancar la finestra oberta pel primer botó.

Solució:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de control d'esdeveniments</title>
  <script>
    var finestra
  </script>
</head>
<body>
<form enctype="text/plain" name="Prova d'esdeveniments">
  <input name="obrir" value="Obrir nova finestra" type="button"
    onClick="finestra=window.open('pagina50.html');"><br>
  <input name="tancar" value="Tancar l'altra finestra" type="button"</pre>
   onClick="finestra.close();"><br>
</form>
</body>
</html>
```

Observem que per a obrir una nova finestra utilitzem el mètode open() de l'objecte window. Aquest mètode retorna un identificador de l'objecte A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina53.html* amb el codi de la solució. window corresponent a la nova finestra oberta que guardem a la variable finestra gràcies a l'assignació finestra=window.open(...). Per a tancar la nova finestra necessitem executar-hi el mètode close() de l'objecte window, però no pas sobre la nostra finestra, sinó sobre la finestra que volem tancar, l'identificador de la qual havíem guardat a la variable finestra; per això escrivim finestra.close().

En aquest exemple hem utilitzat el mètode open() en la seva forma més simple, és a dir, utilitzant un únic paràmetre per indicar la direcció de la pàgina que desitgem obrir a la nova finestra. Hi ha, però, altres possibilitats que es gestionen amb més paràmetres.

En l'execució de l'exemple anterior, cada vegada que premem el primer botó s'obre una nova finestra, però a la variable finestra només queda enregistrat l'identificador de la darrera finestra oberta. Així, l'efecte de prémer el segon botó per a tancar la finestra només tindrà efecte sobre la darrera finestra oberta. Podríem millorar el codi de manera que només es pogués obrir la finestra si no havia estat encara oberta:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de control d'esdeveniments</title>
  <script>
    var finestra
  </script>
</head>
<body>
<form enctype="text/plain" name="Prova d'esdeveniments">
  <input name="obrir" value="Obrir nova finestra" type="button"
    onClick="if (finestra==null) finestra=window.open('pagina50.html');
  else alert('La finestra ja està oberta');"><br><input name="tancar" value="Tancar l'altra finestra" type="button"
    onClick="if (finestra==null) alert('La finestra no està oberta');
              else {finestra.close(); finestra=null;}"><br>
</form>
</body>
</html>
```

Per aconseguir-ho observem que ens fixem en el valor que tingui la variable finestra. Recordem que una variable, quan es crear té el valor null. Així, si no té el valor null és per què conté algun valor, és a dir, l'identificador de l'objecte window corresponent a la nova finestra. Per tant, si la variable finestra no val null hem d'entendre que la nova finestra ja s'ha obert i per tant no l'hem de tornar a obrir. El valor de la variable finestra també l'hem d'utilitzar per a saber si la podem tancar o, si pel contrari, no està oberta.

Hi ha encara un problema... Imagineu que l'usuari ha tancat manualment la finestra oberta. Llavors, el valor de la variable finestra de poc ens serveix i hauríem de tenir alguna manera de detectar si la finestra encara és viva. Això ho podem aconseguir amb la utilització de la propietat close de l'objecte window, que ens retorna un booleà de valor true si la finestra sobre la que actua és tancada i false en cas contrari:

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer pagina55.html amb el nou codi de la solució.

©Isidre Guixà i Miranda

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina54.html* amb el nou codi de la solució.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple de control d'esdeveniments</title>
  <script>
    var finestra
  </script>
</head>
<body>
<form enctype="text/plain" name="Prova d'esdeveniments">
  <input name="obrir" value="Obrir nova finestra" type="button"
    onClick="if (finestra==null || finestra.closed==true) finestra=window.open('pagina50.html');
  else alert('La finestra ja està oberta');"><br><input name="tancar" value="Tancar l'altra finestra" type="button"
    onClick="if (finestra==null || finestra.closed==true) alert('La finestra no està oberta');
              else finestra.close();
              finestra=null;"><br>
</form>
</body>
</html>
```

3.12. Exemples d'scripts molt emprats

En la web podem trobar moltes pàgines dedicades a publicar scripts de JavaScript molt utilitzats en el disseny de pàgines web. Aquest apartat pretén fer arribar algunes idees per a utilitzar en el disseny web.

3.12.1. Darrera modificació d'una pàgina

Aquest és un senzill script per a mostrar la data de la darrera modificació d'un document.

A vegades és molt interessant saber en quina data es va realitzar la darrera modificació i dona una imatge de pàgina actualitzada. Evidentment, per a mostrar en una pàgina la data de la darrera actualització, es pot fer manualment, introduint la data en el codi HTML, però llavors hem de recordar de fer-ho cada vegada que efectuem alguna modificació a la pàgina.

Ens interessa, per tant, que la pàgina per si sola mostri la data de la darrera modificació. L'script que permet fer això és molt simple i es basa en la utilització de la propietat lastModified de l'objecte document:

<script> document.write(document.lastModified) </script>

3.12.2. Botó per a tornar

Aquest és un senzill script per a mostrar un botó per a tornar enrere com el que tenim a la barra d'eines del navegador.

<input type="button" value="Tornar" onClick="history.go(-1)">

Observem que l'script es basa en la utilització del mètode go() de l'objecte history de la finestra actual.

L'objecte history d'una finestra, conté la informació de totes les pàgines visitades per aquella finestra. Així, la propietat length informa del número de pàgines visitades per la finestra i la crida al mètode go() amb paràmetre -1 permet anar a la pàgina anterior i amb paràmetre 1 a la pàgina posterior.

3.12.3. Data catalana

Aquest és un senzill script que mostra la data del dia en llengua catalana.

```
<script language = "JavaScript">
  var ara=new Date();
  switch (ara.getMonth())
  { case 0: nomMes="Gener";break;
 case 1: nomMes="Febrer";break;
    case 2: nomMes="Marc";break;
case 3: nomMes="Abril";break;
    case 4: nomMes="Maig";break;
    case 5: nomMes="Juny"; break;
    case 6: nomMes="Juliol";break;
    case 7: nomMes="Agost";break;
    case 8: nomMes="Setembre";break;
    case 9: nomMes="Octubre"; break;
    case 10: nomMes="Novembre";break;
    case 11: nomMes="Desembre";break;
  switch (ara.getDay())
  { case 0: nomDia="Diumenge";break;
    case 1: nomDia="Dilluns"; break;
    case 2: nomDia="Dimarts"; break;
    case 3: nomDia="Dimecres";break;
    case 4: nomDia="Dijous";break;
    case 5: nomDia="Divendres";break;
    case 6: nomDia="Dissabte";break;
  document.write(nomDia + ", a " + ara.getDate() + " de " + nomMes + "
de " + ara.getFullYear() );
</script>
```

3.12.4. Hora actual

Aquest és un senzill script que mostra l'hora actual:

```
<script language = "JavaScript">
  var ara=new Date();
  var hora=ara.getHours();
  var minuts=ara.getMinutes();
  var segons=ara.getSeconds();
  if (hora<10) hora="0"+hora;
  if (minuts<10) minuts="0"+minuts;
  if (segons<10) segons="0"+segons;
  document.write(hora + ":" + minuts + ":" + segons);
</script>
```

3.12.5. Rellotge

Per aconseguir un rellotge necessitem un temporitzador. L'objecte window disposa dels mètodes:

• setTimeout() per a iniciar un temporitzador

clearTimeout() per a aturar un temporitzador actiu

El mètode setTimeout() admet dos paràmetres: el primer, el nom d'una funció a executar; el segon, el número de milisegons que ha de passar entre cada execució de la funció indicada en el primer paràmetre. Aquest mètode retorna un valor numèric que identifica el temporitzador i es pot utilitzar com a paràmetre del mètode clearTimeout() en cas que es vulgui aturar el temporitzador.

Així doncs, podem tenir una funció horaActual() que calculi l'hora en el moment en que es cridi i fer que ella mateixa posi en marxa un temporitzador que la cridi al cap de 1000 milisegons (és a dir, després de l segon). D'aquesta manera aconseguim un rellotge!

Per a que el rellotge estigui visible, la funció horaActual() haurà d'abocar l'hora en una determinada ubicació. Això s'acostuma a fer definint un camp de text en un formulari dins la pàgina i fent que la funció horaActual() aboqui el contingut dins el camp de text.

A la secció "Recursos" del web d'aquest crèdit podeu trobar el fitxer *pagina56.html* amb el codi del rellotge.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Rellotge</title>
<script language = "JavaScript">
function horaActual()
  var ara=new Date():
  var hora=ara.getHours();
  var minuts=ara.getMinutes();
  var segons=ara.getSeconds();
  if (hora<10) hora="0"+hora;
  if (minuts<10) minuts="0"+minuts;
  if (segons<10) segons="0"+segons;
  document.forms.formulariRellotge.rellotge.value=hora + ":" + minuts + ":" + segons;
window.setTimeout("horaActual()",1000);
</script>
</head>
<body onLoad="horaActual();">
<form name="formulariRellotge">
    Hora actual: <input type="text" name="rellotge">
</form>
</body>
</html>
```

Observem com s'ha d'indicar, dins la funció horaActual(), el camí per accedir al camp de text on ha d'abocar l'hora a cada segon:

document.forms.formulariRellotge.rellotge.value = ...

El camí és lògic doncs cal abocar l'hora dins la propietat value del camp anomenat rellotge del formulari de nom formulariRellotge que forma part dels formularis (forms) del document actiu.