

Ús bàsic d'HTML i CSS

Implantació d'Aplicacions Web

IES Sant Vicent



Aquesta obra està llicenciada sota la Llicència Creative Commons Atribució-NoComercial-CompartirIgual 4.0 Internacional. Per a veure una còpia d'aquesta llicència, visita

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Introducció al disseny web

En aquest tema tractarem de donar una visió el més completa possible del que és el disseny web, que juntament amb Javascript, conformen tota la part que afecta el client d'una aplicació web, és a dir, la programació en el costat del client.

A l'hora d'afrontar el disseny web d'un lloc, fonamentalment acudirem a dos tipus de documents, que poden formar un només o estar enllaçats entre si:

- Els documents **HTML**, mitjançant els quals dotarem d'estructura i contingut a cada pàgina, és a dir, indicarem quins tipus de continguts tindran (imatges, paràgrafs, títols, subtítols, etc), i en quina ordre.
- Els documents **CSS**, mitjançant els quals dotarem d'estil a les pàgines, és a dir, indicarem aspectes com el tipus de lletra, grandària, colors, vores, marges, etc.

Com veurem al llarg d'aquest capítol, tots dos conceptes són independents, és a dir, podem tindre pàgines HTML que no tinguin cap estil CSS (encara que no és l'habitual), i documents d'estils CSS que no apliquem a cap pàgina (encara que serien inútils). Així, aquestes dues eines de disseny poden caminar en paral·lel, vincular-se l'una a l'altra en les pàgines que ens interessin, o bé embeure els estils en la pròpia pàgina. Veurem més endavant quins avantatges i desavantatges té cadascuna d'aquestes maneres de dissenyar.

Al llarg del capítol, s'aniran intercalant alguns conceptes d'HTML amb uns altres de CSS, de manera que s'anirà avançant per igual en totes dues matèries. En altres llibres és habitual comptar primer una cosa (normalment HTML) i després l'altra (normalment CSS), però són dos espais bastant amplis, i podria ocórrer que ens avorrim d'estar treballant amb un d'ells sense veure quines possibilitats reals té en una web, combinat amb l'altre.

Introducció a HTML

Veurem a continuació que una pàgina web no és més que un arxiu de text (que podem fer amb el mateix bloc de notes) on, mitjançant una sèrie d'etiquetes, configurem l'estructura i el contingut que tindrà aquesta pàgina. Aquestes etiquetes que emprem formen un llenguatge, anomenat **HTML** (sigles de *HyperText Markup Language*) que és el llenguatge amb el qual s'escriuen la majoria de pàgines web. És un llenguatge estàndard, reconegut a tot el món, i definit pel W3C (*World Wide Web Consortium*), una entitat sense ànim de lucre.

Història

La seua **història** es remunta als anys 80, quan es van proposar sistemes d'hipertext per a compartir documents. Veurem que l'hipertext és un mecanisme mitjançant el qual, des d'un document, podem accedir a altres relacionats amb aquest, normalment mitjançant el que es denominen *links* o enllaços. Durant els anys 90 va tractar de definir-se un **estàndard** mundialment reconegut, amb les primeres versions d'HTML. Va ser amb HTML 2.0 en 1996 quan es va aconseguir el primer estàndard, publicat llavors per una altra organització, anomenada IETF (*Internet Engineering Task Force*). Ja després va ser el W3C qui va publicar les noves versions que passarien a ser estàndard: HTML 3.2 en 1997 (que permetia entre altres coses incorporar miniaplicacions Java a les pàgines), i HTML 4.0 en 1998 (que permetia incloure fulles d'estil CSS i xicotets programes anomenats *scripts*). Durant un temps, l'activitat entorn d'HTML es va detindre, fins que en els últims anys s'ha donat una nova espenta des de la publicació dels primers esborranys de la nova versió, HTML 5, en 2008, amb millores molt significatives. Paral·lelament a aquest procés, des de l'any 2000, l'organisme W3C ha vingut desenvolupant un altre estàndard, anomenat **XHTML**, que podem considerar una mescla d'HTML i XML. Va partir de la versió 1.0, i ja s'han publicat esborranys de versions posteriors, com la 1.1 o la 2.0.

Estructura bàsica d'una pàgina HTML

Si provem de carregar en el navegador qualsevol pàgina web, i accedim amb clic dret (o amb el menú *Veure* del navegador) al seu codi font, s'obrirà una espècie de bloc de notes, i el que puguem llegir en ell probablement ens sona a xinés. La pàgina original que veiem en el navegador, amb els seus tipus de lletra, imatges, colors, etc., internament amaga un munt de text aparentment sense sentit, ple de paraules estranyes que després no apareixen per cap part en la pàgina. Per exemple, un fragment del codi font de la pàgina principal de Google podria ser una cosa així:

```
<html ...>
...
<head>
...
</head>
<body class="hp vasq" onload="...">
<div class="ctr-p" id="viewport">
...
```

En general, qualsevol pàgina web escrita amb HTML té una estructura comuna composta per:

```
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

En aquesta estructura, podem diferenciar els següents apartats:

- Tota l'estructura i contingut de la pàgina està englobada pel parell d'etiquetes **<html>** i **</html>**
- La secció englobada entre les etiquetes **<head>** i **</head>** es denomina **capçalera**, i en ella es defineixen aspectes generals de la pàgina, però no la seua estructura i contingut. Per exemple, ací definirem el joc de caràcters emprat, o el títol (*title*) de la pàgina, que podrem veure en la pestanya corresponent del navegador.
- La secció entre les etiquetes **<body>** i **</body>** defineix l'estructura i contingut de la pàgina (el que veiem en la finestra del navegador). Dins d'aquesta secció podem utilitzar una sèrie d'etiquetes per a definir uns certs tipus de continguts, com per exemple encapçalats, paràgrafs, llistes, taules, imatges, etc.

Algunes versions d'HTML, com XHTML, afigen altres elements bàsics a aquesta estructura, com una etiqueta `<?xml...>` al principi de tot, o una etiqueta `<DOCTYPE...>` sota l'anterior, que defineixen el tipus de document XHTML que s'està escrivint. Però l'important ara com ara és fer-nos una idea que els continguts s'organitzen en etiquetes, i que (normalment) aquestes van per parelles, una que inicia una secció, i una altra que la tanca (precedida per una barra /). També és important recalcar que les etiquetes van sempre en **minúscula** (*html*, *head*, *body*), encara que després el contingut i altres atributs que veurem després sí que poden contindre majúscules.

Ús bàsic de l'etiqueta `<head>`

Hem explicat anteriorment que l'etiqueta `<head>` s'utilitza per a definir aspectes generals de la pàgina, independentment del contingut i estructura que tindrà. Fonamentalment, la utilitzarem per a:

- Indicar el **títol** de la pàgina (visible en la pestanya del navegador)
- Indicar el **joc de caràcters** que usarem (codificació de l'arxiu, a fi que després es puguin veure bé símbols específics d'uns certs idiomes, com per exemple els accents). Alguns dels jocs més utilitzats són UTF 8 o ISO-8859-1 (per a alfabet llatí). Aquest apartat convé que siga el primer en el *head*, perquè el navegador el llija com més prompte millor.
- Indicar la(s) fulla(s) d'estil **CSS** que s'aplicaran a la pàgina
- Indicar altres **recursos externs** (com per exemple arxius de codi Javascript) que s'utilitzaran en la pàgina.

Ús bàsic d'HTML i CSS

De les últimes dues opcions parlarem més endavant. Quant a les dues primeres, si volem afegir un títol a la pàgina, el farem mitjançant el parell d'etiquetes **title** (dins del bloc *head*), i si volem especificar un joc de caràcters, el farem amb l'etiqueta **meta**. Per exemple, si volem posar com a títol el text "Benvingut/a", i utilitzar el joc de caràcters ISO-8859-1, posaríem alguna cosa com això dins del bloc *head*:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
  <title>Benvingut/a /</title>
</head>
...
```

L'etiqueta *meta* s'utilitza en general per a definir *metainformació* de la pàgina (és a dir, informació sobre la informació que hi ha en la pàgina). Pot haver-hi diverses dins del bloc *head*, i amb elles podem indicar diferents elements a més del joc de caràcters, com per exemple l'autor de la pàgina, o un conjunt de paraules clau útils perquè els cercadors puguin localitzar millor la pàgina. Per exemple, en la següent pàgina indiquem l'autor i un conjunt de paraules clau de la pàgina:

```
<html>
<head>
  <title>...</title>
  <meta author="Sergi Aracil"/>
  <meta keywords="curs, HTML, CSS, web, PHP, aplicacions"/>
</head>
...
```

Notar també que l'etiqueta *meta* no té parella, va ella sola, i acaba amb una barra `</>` (encara que si no la posem, també és correcta en HTML no estricte).

Ús bàsic de l'etiqueta `<body>`

L'etiqueta `<body>` defineix el contingut i estructura de la pàgina, és a dir, la qual cosa es veu en la finestra del navegador quan carreguem la pàgina (textos, imatges, etc.). A l'interior d'aquesta etiqueta podrem utilitzar una gran varietat d'etiquetes, que veurem en les següents seccions, per a definir cadascun dels tipus de text i apartats de la pàgina. Per a començar, si posem qualsevol text entre el parell d'etiquetes *body*, es mostrarà aqueix text tal qual en el navegador, encara que sense comptar els salts de línia (Intros) que posem.

```
...
<body>
Hola, bon dia.
Això és una pàgina web de prova.
</body>
</html>
```

Es veuria una cosa així en el navegador:

Hola, bon dia. Això és una pàgina web de prova.

Comentaris en HTML

Igual que qualsevol altre llenguatge de programació, HTML permet afegir **comentaris** en el codi. Aquests comentaris poden servir per a dues coses: explicar una part del codi (en

aquest cas, de l'estructura o continguts de la pàgina web) que no quede clara, o delimitar / marcar una zona del codi per a poder-la localitzar o identificar més fàcilment. Els comentaris en HTML comencen amb el símbol `<!--` i acaben amb el símbol `-->`. El que hi haja entre aquests símbols no es mostra en el contingut de la pàgina, només es pot llegir mirant el codi font. Vegem un exemple:

```
<html>
<head>
  <!-- Afegim el joc de caràcters ISO-8859-1 -->
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
  <!-- Títol de la pàgina -->
  <title>Benvingut/a /<title>
</head>
<body>
  <!-- En aquest exemple, simplement mostrarem un text de salutació -->
  Hola, bon dia.
</body>
</html>
```

Què necessitem per a escriure pàgines web?

Qualsevol editor de textos simple serveix, com per exemple el bloc de notes (si estem en Windows) o gEdit (si estem en Linux). Una opció molt recomanable en Windows és utilitzar **Notepad++**. És un editor gratuït, que permet editar diferents tipus d'arxius (per exemple, pàgines HTML, pàgines PHP, etc.) i té realç de sintaxi (és a dir, que mostra amb diferents colors els diferents elements del document, per a poder-los diferenciar millor). En general, qualsevol editor de textos senzill amb realç de sintaxi (com per exemple **gEdit** en Linux, o **TextWrangler** en Mac OS X) ens pot servir.

Exercici 1

Crea amb el teu editor de textos un nou document, i crida'l **pagina1.html**. Dins, afig una estructura bàsica d'etiquetes HTML perquè el document tinga el següent:

- Un títol que siga “La meua primera pàgina HTML”
- Un joc de caràcters ISO-8859-1 o UTF-8.
- Un text (en el *body*) amb el teu nom

Guarda la pàgina (si pot ser, en format UTF-8), i prova a obrir-la amb un navegador per a comprovar que es veu correctament (incloent els accents que puga haver-hi).

Etiquetes HTML bàsiques de text

Veurem en aquesta secció algunes de les principals etiquetes que podem utilitzar en la secció *body* d'una pàgina web per a dotar-la de diferents tipus de textos estructurats.

Paràgrafs

Els paràgrafs en HTML es marquen amb la parella d'etiquetes **p**. Un paràgraf és un element separat de la resta. Quan ho acabem i comencem un altre, es té un punt i apart automàtic, i una separació automàtica entre paràgrafs.

```
<body>
  <p>Això és un paràgraf.</p>
  <p>Això és un altre paràgraf.</p>
</body>
```

Que es veuria en el navegador més o menys així:

Això és un paràgraf.

Això és un altre paràgraf.

Seccions o “headings”

Si tot el contingut d'una pàgina web anaren paràgrafs, es faria bastant pesat llegir i localitzar la informació. Per a organitzar millor el contingut, hem de separar-lo en seccions o encapçalats (en anglès, *headings*), cadascuna amb el seu corresponent títol. Així, podem tindre seccions o *headings* principals, dividides al seu torn en subseccions, i aquestes al seu torn en més subseccions, etc., i cadascuna amb la seua(s) corresponent(s) paràgraf(s). En total, podem tindre fins a 6 nivells. Els principals es marquen amb l'etiqueta **h1**, els de segon nivell amb l'etiqueta **h2**... i així successivament fins a **h6** (encara que generalment se sol arribar fins a **h3** o **h4**).

Per exemple, una pàgina amb 2 apartats principals sobre HTML i CSS i cadascun amb 2 subapartats dins, podria tindre una estructura com:

```
<body>

  <h1>HTML</h1>

  <h2>Introducció a HTML</h2>
  <p>HTML és un llenguatge que...</p>
  <p>La seua història es remunta a...</p>

  <h2>Etiquetes bàsiques d'HTML</h2>
  <p>Algunes de les etiquetes que podem utilitzar són...</p>

  <h1>CSS</h1>

  <p>CSS és...</p>

  <h2>Estructura d'un document CSS</h2>
  <p>Un document CSS es compon de...</p>
```

```
<h2>Principals tipus de regles</h2>
<p>Els principals tipus de regles CSS són...</p>

</body>
```

En un navegador, això podria veure's més o menys així:

HTML

Introducció a HTML

HTML és un llenguatge que...

La seua història es remunta a...

Etiquetes bàsiques d'HTML

Algunes de les etiquetes que podem utilitzar són...

CSS

CSS és...

Estructura d'un document CSS

Un document CSS es compon de...

Principals tipus de regles

Els principals tipus de regles CSS són...

Exercici 2

Crea una pàgina web anomenada **currículum.html** que tinga el següent:

- Com a títol de la pàgina (etiqueta *title* dins de *head*), posa-li el teu nom
- En el cos (*body*), haurà de tindre:
 - Un encapçalat principal (*h1*) anomenat “Dades personals”, on hauràs d'indicar en un o dos paràgrafs informació sobre tu: nom i cognoms, adreça i telèfon (et pots inventar l'adreça i el telèfon si vols).
 - Un altre encapçalat principal (*h1*) anomenat “Formació”, que tindrà dins dos subapartats (*h2*): un anomenat “Idiomes” on hauràs d'indicar els idiomes que pots parlar o escriure correctament (en un o dos paràgrafs), i un altre anomenat “Estudis” on també en un dos paràgrafs indiques els estudis que has aconseguit fins ara, incloent els teus estudis actuals (graduat d'ESO, Batxillerat, Cicles Formatius, etc.)

Algunes etiquetes de format

Fins ara no hem parlat de gens de format en pàgines HTML. De fet, des del propi HTML no es pot (o no s'aconsella) donar format als documents, és a dir, definir tipus de lletra, grandàries, colors, etc. Això es farà amb estils CSS, com veurem.

No obstant això, existeixen algunes etiquetes en HTML que sí que poden usar-se per a donar una mica de “format” a unes certes parts del document, amb la finalitat de ressaltar aqueixes parts sobre la resta. En concret, podem:

- Posar un text en **negreta**, utilitzant la parella d'etiquetes **strong**
- Posar un text en **cursiva**, utilitzant la parella d'etiquetes **em**
- Indentar un text cap a la dreta (afegir-li una sagnia per l'esquerra) amb l'etiqueta **blockquote**

Les dues primeres etiquetes (*strong* i *em*) se solen utilitzar dins de paràgrafs o elements similars (llestes, sagnies, etc.), per a ressaltar una part d'aqueix paràgraf. L'etiqueta per a sagnies (*blockquote*) és una alternativa al paràgraf (no s'utilitza dins del paràgraf, sinó en lloc del paràgraf), per a sagnar un text cap a la dreta.

Vegem alguns exemples:

```
...  
<body>  
  <p>Això és un paràgraf amb <strong>negreta</strong>.</p>  
  <p>Aquest altre paràgraf té <em>text en cursiva</em>.</p>  
  <blockquote>Aquest paràgraf està <em>indentat</em> a la dreta.</blockquote>  
</body>
```

En un navegador, això quedaria una cosa així:

Això és un paràgraf amb **negreta**.

Aquest altre paràgraf té *text en cursiva*.

Aquest paràgraf està *indentat* a la dreta

Altres etiquetes de format

Existeixen altres etiquetes per a format que no veurem en aquest tema, com l'etiqueta per a posar text subratllat (està desaconsellada per a no confondre a l'usuari i fer-li creure que pot punxar damunt del text), ratllat, superíndex, subíndex, etc.

L'etiqueta *span*

L'etiqueta **span** s'utilitza per a “marcar” (encara que després no es veu gens diferent en el navegador) una part d'un text, normalment continguda dins d'un altre major (dins d'un paràgraf, un *h1*, etc.).

```
<p>Conten que assegut sota una pomera, <span>Isaac Newton</span> va descobrir la llei de la gravetat.</p>
```

Per si soles no suposen un canvi visual en la pàgina, però combinades amb fulles d'estil CSS podem fer que aqueixa part del text marcada entre *span* tinga un estil diferent del que li envolta (diferent color, grandària, etc.), i així ressaltar el text d'una forma diferent a posar simplement negreta i/o cursiva.

Etiquetes “especials”: *br* i *hr*

Fins ara, hem vist que totes les etiquetes HTML (excepte l'etiqueta *meta* de la capçalera) van per parelles: una comença alguna cosa i l'altra ho acaba. Existeixen algunes etiquetes

que van soles, com és el cas de les etiquetes *br* i *hr* (entre altres que veurem més endavant).

L'etiqueta **br** s'utilitza per a forçar un salt de línia (un punt i a part) allà on es pose. Normalment quan usem un paràgraf, automàticament es té el punt i a part en acabar-lo, però pot ser que ens interesse baixar de línia en algun punt intermedi, o fora d'un paràgraf, per alguna raó. En aquest cas, utilitzarem l'etiqueta *br* com segueix:

```
<body>
<p>Això seria un paràgraf llarg, que ocupa diverses línies. Si no posem res, ell
només s'ajusta amb el seu punt i a part, però si posem l'etiqueta br <bbr />es
produeix un salt de línia en el punt on hàgem col·locat aqueixa etiqueta.</p>
</body>
```

Això es veuria així en el navegador:

Això seria un paràgraf llarg, que ocupa diverses línies. Si no posem res, ell només s'ajusta amb el seu punt i a part, però si posem l'etiqueta *br* es produeix un salt de línia en el punt on hàgem col·locat aqueixa etiqueta.

Per part seua, l'etiqueta **hr** serveix per a inserir una línia horitzontal en el lloc on s'haja col·locat. S'utilitza normalment per a separar continguts en una pàgina web de forma més visible. Per exemple:

```
...
<body>
<p>Això és un paràgraf.</p>
<bhr />
<p>Això és un altre paràgraf separat per una línia horitzontal.</p>
...
```

Això es veuria així en el navegador:

Això és un paràgraf.

Això és un altre paràgraf separat per una línia horitzontal

Alguns símbols especials

Com podríem representar en una pàgina web el símbol “<”? Ho utilitzem per a les etiquetes, i en el moment en què el posem, el navegador creurà que estem començant una etiqueta i no el mostrarà.

Igual que aqueix símbol, o el seu contrari “>”, existeixen uns certs símbols que, quan els escrivim directament en el document, no es representen bé (o directament, no es representen) en la pàgina, perquè el navegador no els reconeix com a símbols imprimibles. En el seu lloc, hem d'escriure un codi alternatiu. Per exemple, en el cas de “<” haurem de posar *<*. Ací veiem un resum d'alguns símbols útils:

Símbol	Codi
<	<
>	>
&	&
“	"

‘	'
(espai)	

Respecte als espais, encara que si posem un espai entre paraula i paraula, es representarà bé, quan vulguem utilitzar l'espai com a element indentador (per exemple, posar un text cinc espais cap a dins) ja no va si pot ser. En aquests casos haurem d'utilitzar el codi de l'espai () i repetir-lo tantes vegades com espais vulguem posar.

[illegible]

Quant als accents i lletres especials (com la “ñ” en espanyol), si utilitzem la codificació adequada (indicada en les etiquetes *meta*), els podrem escriure tal qual. En cas contrari, també tenen la seua codificació particular:

- Els **accents aguts** (els normals en espanyol), es representen amb el símbol &, seguit de la lletra que vulguem accentuar (en majúscules o minúscules, però sense l'accent), la paraula “acute”, i el punt i coma. Així, per exemple, la “a” minúscula accentuada (á), es representaria amb á i la “o” majúscula accentuada (O), es representaria amb Ó
- Els **accents greus** (o oberts, propis d'idiomes com el català o el francès) es representen igual que els aguts, però substituint la paraula “acute” per “grave”. Així, la “a” minúscula amb accent obert (à), es posaria à
- La «**eñe**» es representa amb el símbol & seguit d'una n o N (segons si volem ñ o Ñ), i la paraula “tilde”, finalitzat tot en punt i coma. Així, una «**eñe**» minúscula (ñ) es representaria ñ

Exercici 3

Crea una nova pàgina anomenada **historialInternet.html**. Dins, copiarem aquest contingut:

Historia

En 1969 surge la primera red interconectada, cuando se crea el primer enlace entre las universidades de UCLA y Stanford por medio de la línea telefónica conmutada. Es el proyecto ARPANET, del que se tiene el mito de que se construyó simplemente para sobrevivir a ataques nucleares. Sin embargo, no fue este el único motivo, sino que, además, otra razón fue que los nodos de conmutación eran poco fiables.

En 1986 la NSF comenzó el desarrollo de NSFNET que se convirtió en la principal *Red en árbol* de Internet, complementada después con las redes NSINET y ESNET, todas ellas en Estados Unidos. Paralelamente, otras redes troncales en Europa, tanto públicas como comerciales, junto con las americanas formaban el esqueleto básico ("backbone") de Internet.

Nombres de dominio

La *Corporación de Internet para los Nombres y los Números Asignados* (ICANN) es la autoridad que coordina la asignación de identificadores únicos en Internet, incluyendo nombres de dominio, direcciones de Protocolos de Internet, números del puerto del protocolo y de parámetros. Un nombre global unificado (es decir, un sistema de nombres exclusivos para sostener cada dominio) es esencial para que Internet funcione.

- Encara que no es veu en l'exercici, en la capçalera *head* del document has de posar com a títol (*title*) la frase "Història d'Internet", i com a codificació l'ISO-8859-1.

- El primer títol (*Història*) és un encapçalat de nivell 1 (*h1*). Davall té una línia de divisió horitzontal (*hr*)
- Davall de la línia hi ha dos paràgrafs explicatius. Els números d'any van en negreta, i pot haver-hi algun text en cursiva també.
- El segon títol (*Noms de domini*) és un encapçalat de nivell 2 (*h2*). Davall d'ell té un altre paràgraf, on també hauràs de posar en cursiva el que estiga en cursiva.

Introducció a CSS

Abans de continuar aprofundint en HTML, farem un alt en el camí per a començar a parlar de CSS. Fins ara hem vist que un document HTML no té format, mes allà de les negretes, cursives i poc més. Com podem definir el tipus de lletra d'un document, o el color, o la grandària, entre altres coses?

Per a dotar d'estil a una pàgina, a partir de la versió 4 d'HTML, disposem del que es diuen *fulles d'estil en cascada* (en anglès *Cascading Style Sheets*, abreujat **CSS**). Es tracta de fragments de text que defineixen estils per als diferents elements de les pàgines web, i que poden anar tant embeguts en la mateixa pàgina web, com enllaçats des d'un fitxer a part. D'aquesta manera s'aconsegueix separar d'una banda el contingut (que anirà en pàgines HTML) de l'estil (que anirà en documents CSS).

Història

La primera versió de CSS (CSS1) va ser publicada en 1996. En ella es van definir les directrius per a uns certs estils bàsics, com a tipus de lletra, color de text, alineació, marges, vores...

Li va seguir poc temps després CSS2, publicada en 1998, i que corregia i ampliava les possibilitats de CSS1, afegint posicionaments, ombres, textos bidireccionals, etc. Aquesta versió va tindre una revisió, la CSS2.1, que corregia alguns errors trobats i eliminava algunes funcionalitats que molts navegadors no suportaven.

Finalment, arribem a la versió de CSS3. A diferència de les anteriors, que són especificacions globals de molta informació continguda, aquesta versió està formada per una sèrie de mòduls, i cadascun afig un paquet de funcionalitats al que ja hi havia en CSS2, per la qual cosa es manté la compatibilitat amb aquesta versió. De fet, es va començar a treballar amb CSS3 en 1999, i cada mòdul ha tingut una evolució diferent. Alguns han arribat a estàndards o recomanacions oficials, uns altres estan a punt de ser-ho, i altres encara estan en vies de desenvolupament.

Funcionament bàsic

Per a definir un estil sobre una pàgina web en CSS, s'utilitza el que es diuen **regles**. Una regla es compon del nom de l'element sobre el qual s'aplicarà l'estil i, entre claus i separats per punt i coma, els estils que aplicarem sobre aqueix element (per exemple, tipus de lletra, grandària, color, vores...). Per exemple, si volguérem establir que tots els paràgrafs d'una pàgina web anaren de color roig i de grandària 12 punts tindríem una regla com la següent:

```
p { color:red; font-size: 12pt; }
```

Observem que el primer que posem és el nom de l'element (en aquest cas, de l'etiqueta HTML) sobre el qual aplicarem l'estil, anomenat **selector**, i després entre claus els diferents estils a aplicar (en l'exemple, el color del text i la grandària del mateix). Aquest conjunt d'estils es diu **declaració**, i està format per parelles de propietats i valors (per exemple, la propietat *color* i el valor que tindrà).

Si volguérem afegir un altre estil (una altra regla) perquè els encapçalats de nivell 1 tinguin color verd, podríem afegir-ho a continuació (o abans) de l'anterior:

```
p { color: red; font-size: 12pt; }  
h1 { color: green; }
```

Veurem més endavant diversos exemples d'estils que podem incloure en aquestes claus, dividits en categories, però primer veurem com podem aplicar aquests estils a una pàgina HTML.

Aplicar els estils a les pàgines

Per a aplicar un estil (o un conjunt d'estils) a una pàgina web, tenim diverses alternatives:

- Definir l'estil concret en un element concret de la pàgina web, mitjançant un atribut *style*. Per exemple, si volem que un paràgraf concret tinga el text de color roig, podem posar:

```
<p style="color: red;">Paràgraf de color roig.</p>
```

Aquesta opció no s'aconsella molt, llevat que siga un cas puntual d'un element puntual amb un estil particular, ja que en cas contrari costaria molt revisar tot el codi HTML buscant estils si cal fer algun canvi.

- Definir el conjunt d'estils dins del bloc *head*, entre una parella d'etiquetes *style*. Per exemple, per a aplicar els dos estils de l'exemple anterior (per a paràgrafs i *h1*) a tots els paràgrafs i *h1* d'una pàgina, posaríem alguna cosa com això en el *head*:

```
<html>  
<head>  
  ...  
  <style>  
    p { color: red; font-size: 12pt; }  
    h1 { color: green; }  
  </style>  
</head>  
  ...
```

Aquesta opció tampoc és molt recomanable si volem aplicar el mateix estil en diverses pàgines, perquè hauríem de repetir el mateix bloc *style* en totes, amb els consegüents problemes de duplicitat que tindríem si volem fer qualsevol canvi.

- Definir tots els estils en un document CSS apart (guardat amb extensió *.css*) i després enllaçar aqueix arxiu amb la pàgina web, des del *head* d'aquesta. Per exemple, si l'arxiu d'estils l'hem anomenat *estils.css* i està en la mateixa carpeta que la pàgina web on aplicar-ho, posaríem una cosa així en el *head* (l'atribut *type* és opcional):

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

Si l'arxiu estiguera dins d'una subcarpeta anomenada *estils*, la línia anterior canviaria per aquesta altra (per a indicar la subcarpeta):

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

Aquesta opció és la més recomanada, perquè queda tot centralitzat en l'arxiu CSS, i qualsevol canvi que calga fer, es fa sobre l'arxiu i repercuteix alhora en totes les pàgines que l'enllacen. A més, d'aquesta manera es pot dividir millor el treball: una(s) persona(s) s'encarrega(n) de l'HTML, i una altra(s) del CSS.

Estils CSS bàsics de text

Veurem a continuació alguns estils que podem aplicar per a modificar el format del text d'una pàgina HTML: tipus de lletra, grandàries, alineacions, colors... així com alineacions i altres efectes.

Format de caràcter

Les següents propietats afecten a cadascun dels caràcters de l'element HTML. Seria l'equivalent a anar al menú *Format > Font* en Word, o *Format > Caràcter* en OpenOffice o LibreOffice.

- Per a definir el **tipus de lletra** utilitzarem la propietat **font-family**. En ella podem posar un tipus de font determinat (entre cometes dobles si té espais), o diversos separats per comes. En aquest cas, el navegador triarà la primera de les fonts que tinga instal·lades el sistema.

```
p { font-family: "Courier New", Arial; }
```

- Per a definir la **grandària de la lletra** usem la propietat **font-size**, seguit de la grandària de la lletra. Normalment s'expressa en punts (pt), però més endavant veurem que podem utilitzar altres unitats de mesura.

```
h1 { font-family: Verdana; font-size: 30pt; }
```

- Per a definir el **gruix de la lletra** (de manera diferent a posar simplement una negreta), usem la propietat **font-weight**. Per a simular un aspecte de negreta, se li dona el valor de **bold**. Altres valors possibles són *normal*, *bolder* (encara més negreta del normal), *lighter* (més fina del normal), o un valor numèric (400 per a normal, 700 per a negreta...). Per exemple, si volem que els *blockquote* siguin una mica més gruixuts del normal, simulant negretes, podríem posar:

```
blockquote { font-weight: bold; }
```

- Per a definir l'estil **de la lletra** (de manera diferent a posar simplement una cursiva), usem la propietat **font-style**. Per a simular un aspecte de cursiva, se li dona el valor de *italic*. Altres valors possibles són *normal* o *oblique* (aquest últim dona un aspecte similar a *italic*). Per exemple, si volguérem que els *blockquote* estiguessin en cursiva, podríem posar:

```
blockquote { font-style: italic; }
```

- Per a definir el **color** de la lletra s'usa la propietat **color**. Podem posar un color simple en anglès (*blue*, *red*, *yellow*...) com a exemple. Més endavant veurem que existeixen altres maneres d'especificar molts més colors.

```
h2 { color: blue; }
```

- Per a establir la **decoració** del text (és a dir, altres efectes com subratllats, ratllats, etc.), s'usa la propietat **text-decoration**. Si volem fer un efecte de subratllat, li donarem el valor *underline*. Si volem un aspecte de ratllat, li donarem el valor *line-through*. Podem especificar altres valors, com *none* (cap decoració, valor per defecte per a la majoria d'etiquetes HTML) o *overline* (subratllat per damunt o soberrratllat). Per exemple, si volem posar un subratllat inferior en un *h1*:


```
h1 { text-decoration: underline; }
```

Format de paràgraf

Les següents opcions afecten el comportament de l'element com a conjunt: la seua alineació, interlineat, etc. Seria l'equivalent a les opcions del menú *Format > Paràgraf* en editors de text com a MS Word o OpenOffice.

- Per a modificar l'alineació **horitzontal** de l'element, usem la propietat **text-align**, que pot prendre els típics valors de qualsevol processador de textos: *left*, *right*, *center* o *justify*.

```
p { text-align: justify; }
```

- Per a modificar l'alineació **vertical** de l'element (i col·locar-lo així verticalment dins d'un altre que el continga) està la propietat **vertical-align**, que pot valdre *top* (a dalt), *bottom* (a baix) o *middle* (enmig), entre altres valors. No obstant això, aquesta propietat no sol funcionar molt bé en la majoria dels casos, per exemple, quan volem centrar un text en el mitjà d'una caixa.

```
p { vertical-align: middle; }
```

- Per a modificar la **tabulació o indentació** del text (l'equivalent a una sagnia esquerra), usem la propietat **text-indent**, seguida d'una unitat (veurem més endavant diferents tipus d'unitats). Per exemple, per a deixar una tabulació de 2 cm:

```
p { text-indent: 2cm; }
```

- Per a establir l'interlineat (separació de les línies dins del mateix element, com per exemple un paràgraf), s'usa la propietat **line-height**, seguida d'una mesura. Per exemple, per a un interlineat de 1,5 línies usaríem alguna cosa com:

```
p { line-height: 1.5em; }
```

Exercici 4

Crea un document HTML anomenat **estilsText.html**. Afig-li en el *head* una etiqueta *style* amb els estils embeguts dins. Definirem els següents estils:

- Els encapçalats de primer nivell (*h1*) hauran de tindre lletra Arial de 30 punts, color roig i subratllat.
- Els encapçalats de nivell 2 (*h2*) hauran de ser de lletra Arial, color blau, grandària 20 punts. Hauran de tindre una alineació horitzontal dreta.
- Els paràgrafs hauran de ser lletra *"Times New Roman"*, grandària 12 punts, color gris, alineació horitzontal justificada i interlineat de 1,5 línies.

En el *body* del document, prova de crear un element almenys de cada tipus (*h1*, *h2* i paràgraf de 3 o més línies), per a comprovar l'estil de cada cosa.

Selectors

Hem vist que una regla CSS es compon del nom de l'element sobre el qual s'aplica l'estil (el selector), i dels estils a aplicar entre claus. Quant als selectors, poden ser de diferents tipus, però els més habituals són els quatre que veurem a continuació.

El selector d'etiqueta

El **selector d'etiqueta** aplica l'estil a totes les etiquetes HTML d'aqueix tipus que hi haja en la pàgina (i a les subetiquetes que estiguen contingudes en ella). Per exemple, si apliquem el color roig als paràgrafs, també seran roges totes les negretes i cursives que hi haja dins dels paràgrafs. Per a utilitzar aquest selector, n'hi ha prou amb posar el nom de l'etiqueta HTML sobre la qual aplicar l'estil, com en els exemples que hem vist en la introducció a CSS:

```
p { ... }  
h1 { ... }
```

Si utilitzem el selector **body**, estarem definint **estils generals** per a tota la pàgina (perquè totes les altres etiquetes estan contingudes dins de body). Així, s'utilitza fonamentalment per a definir tipus de lletra generals (per exemple, que tota la pàgina tinga font Arial, siga com siga la grandària), color o imatge de fons de la pàgina, etc.

```
body { font-family: Arial; background-color: yellow; }
```

Podem també aplicar els mateixos estils a un conjunt d'etiquetes, posant-les totes juntes separades per comes en el selector i després, en altres selectors, definir estils particulars per a alguna d'elles:

```
h1, h2, h3 { color: blue; }  
h1 { font-size: 30pt; }  
h2 { font-size: 24pt; }  
h3 { font-size: 20pt; }
```

El selector descendent

El **selector descendent** s'utilitza per a aplicar un estil a una etiqueta només quan estiga continguda dins d'una altra. Per exemple, si volem que les negretes contingudes dins de paràgrafs (directament o indirectament) siguen de color verd, usariem aquest selector:

```
p strong { color: green; }
```

Notar que es posa primer el nom de l'etiqueta contenidora, i després, separat per espai, el nom de l'etiqueta continguda.

El selector de classe

El **selector de classe** és molt útil quan no volem aplicar un estil a totes les etiquetes d'un tipus en un document, sinó només a algunes d'elles. En aquest cas, es marquen "" aqueixes etiquetes en l'arxiu HTML amb un atribut *class*, i un nom que vulguem. Per exemple, si volem que alguns paràgrafs d'un document tinguen lletra Tahoma, primer marquem en el document HTML aqueixos paràgrafs amb un atribut *class* i el nom que

vulguem (per exemple, *miEstilo*). A més, pot haver-hi altres paràgrafs que no tinguin aqueixa classe, com en aquest exemple:

```
<html>
...
<body>
...
<p class="miEstilo">Aquest paràgraf és de l'estil miEstilo.</p>
<p>Aquest altre paràgraf és normal.</p>
<p class="miEstilo">Aquest paràgraf també és d'estil miEstilo.</p>
...
```

Després, en el document CSS afegim el selector de classe, que té el nom que li hem donat a la classe, precedit per un punt (en el nostre exemple seria *.miEstilo*)

```
.miEstilo { font-family: Tahoma; }
```

Aquest selector també s'aplica a altres elements d'un altre tipus que tinguin la mateixa classe. Per exemple, si posàrem un *h1* amb la mateixa classe, també tindria aquest estil, i per tant lletra Tahoma:

```
<html>
...
<body>
...
<h1 class="miEstilo">Encapçalat amb estil miEstilo</h1>
<p class="miEstilo">...
...
```

Si volem evitar això, i aplicar l'estil només a les classes d'una certa etiqueta, es pot anteposar el nom de l'etiqueta al de l'estil en el CSS. Per exemple, si volem que l'estil només s'aplique als paràgrafs amb classe *miEstilo*, posem:

```
p.miEstilo { font-family: Tahoma; }
```

També podem definir més d'una classe per a un element HTML, separant les classes per espais dins de l'atribut *class*:

```
<p class="miEstilo ressaltat">Aquest paràgraf és...</p>
```

Així, podríem definir un estil *.miEstilo* i un altre estil *.ressaltat*, i cadascun s'aplicaria a les etiquetes que ho tingueren (les que tinguin els dos, rebran els dos estils).

El selector «id»

En un document HTML, podem identificar cada element amb un identificador diferent. Això es fa mitjançant un atribut *id* que podem posar en qualsevol etiqueta. Per exemple, un paràgraf concret pot tindre l'identificador *resum*:

```
<p id="resum">En resum, aquesta pàgina tracta de ... </p>
```

No pot haver-hi en una mateixa pàgina dos elements (encara que siguin etiquetes diferents) amb el mateix identificador. Això serà útil quan tractem el tema dels formularis, més endavant, per a identificar unívocament cada camp, i per a accedir a aqueixos elements des de Javascript.

Una vegada tenim posat un identificador en una etiqueta, podem definir un estil que siga únic per a aqueix identificador. Per exemple, si volem que el paràgraf *resum* de l'exemple

anterior tinga un tipus de lletra diferent de la resta (per exemple, Verdana color gris), en el CSS posem el nom de l'identificador precedit d'un coixinet (#):

```
#resum { font-family: Verdana; color: gray; }
```

Igual que ocorria amb el selector de classe, podem anteposar el nom de l'etiqueta al nom de l'id , però com l'id és únic, això no té molt sentit fer-ho, excepte si apliquem el mateix CSS en diverses pàgines, i en unes altres pot haver-hi altres elements amb el mateix *id*

```
p#resum { font-family: Verdana; color: gray; }
```

Exercici 5

Indica quin selector utilitzaries per a definir un estil per a cadascun d'aquests casos:

1. Tots els elements d'una pàgina
2. Tots els paràgrafs d'una pàgina
3. Un encapçalat de nivell 2 amb *id* "especial"
4. Totes les negretes de classe "clase1"
5. Tots les cursives contingudes en encapçalats de nivell 2

Exercici 6

Donats els següents selectors:

```
p { ... }  
p.miClase { ... }  
p .miClase { ... }  
p, #miID { ... }  
p span em { ... }  
.miClase .miClase2 { ... }
```

Indica a quins textos del següent fragment HTML s'aplicarien:

```
<p>blablablabla <span>blablablabla</span> blablablabla</p>  
<p class="miClase">blabla <span>blablabla<em>blabla</em></span> </p>  
<h1 class="miID">blablablabla <span class="miClase">aaaa</span> </h1>  
<p>blablablabla <span class="miClase">blablabla</span> blabla</p>  
<p class="miClase">blabla <span class="miClase2">blabla</span> </p>
```

Unitats de mesura i colors

Ja hem vist algunes propietats bàsiques de CSS per a modificar l'estil del text. Algunes d'aquestes propietats fan ús d'unitats de mesura (per exemple, per a definir la grandària de la font) o de colors (per a definir el color del text). Veurem ara quin tipus de valors podem donar en aquests tipus de propietats.

Unitats de mesura

En CSS podem utilitzar dos tipus d'unitats de mesura: relatives o absolutes. Les primeres dependran d'algun factor extern (com la resolució de la pantalla, o la grandària de font establert), i les segones no, mesuren sempre el mateix independentment d'altres factors externs.

Unitats de mesura relatives

Podem distingir les següents unitats de mesura relatives:

- **em**: s'utilitza per a definir mesures relatives a la grandària en punts utilitzat actualment. Per exemple, si la grandària (per defecte o per CSS) és de 12 punts i especifiquem una mesura de 0.7em, estarem multiplicant la mesura predefinida (els 12 punts) per 0.7, la qual cosa equivaldria a 8.4 punts de grandària real. Així, 1em equival al 100% de la grandària per defecte.

```
p { font-size: 0.8em; }
```

D'aquesta manera, partint d'una mesura base, es poden definir totes les altres mesures relatives a aquesta, i tot s'escala en proporció.

- **px**: s'utilitza per a definir mesures en píxels. És una mesura relativa perquè la seua grandària dependrà de la resolució del monitor.

```
p { font-size: 15px; }
```

- També podem utilitzar **percentatges**, de manera que l'element afectat prengui aqueix percentatge de la seua grandària màxima permesa.

Unitats de mesura absolutes

Les unitats de mesura absolutes que podem emprar són **cm** (centímetres), **mm** (mil·límetres) i **pt** (per a donar mesures en punts, normalment per a fonts).

```
p { font-size: 12pt; }
```

Exercici 7

Crea un document HTML anomenat **tamanyesEM.html**. Dins, embegut en el *head* amb una etiqueta *style*, defineix aquests estils:

- Tot el document (body) tindrà una grandària de lletra de 20 punts.
- Un selector de classe anomenat *xicotet* que pose la grandària de lletra a 0.8em
- Un altre selector de classe anomenat *normal* que deixi la grandària predefinida (1em)

- Un altre selector de classe anomenat *gran* que pose la grandària de lletra en 1.2em

Els tres estils han d'aplicar-se a tres paràgrafs diferents en el document, per a veure com canvia la grandària de lletra en cadascun d'ells. Afeg també un h1 per a veure quina grandària pren.

Colors

Podem indicar una propietat de color (com per exemple, la propietat *color* per al color del text) de diferents formes:

- Utilitzant algun dels **colors predefinits** que admet CSS. Es tracta de colors simples, amb el nom en anglès: *black*, *white*, *yellow*, *blue*, *red*...

```
p { color: blue; }
```

- Utilitzant la codificació **RGB en decimal** del color. Tots els colors es poden descompondre en una combinació dels colors primaris, que en informàtica són el roig (Red), el verd (Green) i el blau (Blue). Qualsevol color es pot expressar com una combinació de valors d'aquests tres, i aquests valors van des del 0 (res d'aqueix color) fins al 255 (tot aqueix color). Per exemple, el mateix color roig es pot expressar com (255, 0, 0). El morat (roig + blau) es pot expressar com (255, 0, 255). El blanc (unió de tots els colors) seria (255, 255, 255), i el negre (absència de color) seria (0, 0, 0).

Per a indicar un color amb aquest format, es posa la paraula *rgb*, seguida dels tres valors numèrics entre parèntesis, separats per comes. Així, l'exemple anterior es podria posar també com:

```
p { color: rgb(0, 0, 255); }
```

- Utilitzant la codificació **RGB en hexadecimal** del color. Segueix la mateixa filosofia que el cas anterior, però en aquest cas, en lloc de donar valors de 0 a 255 per a cada color primari, es donen els valors en format hexadecimal (des del 00 fins a FF, que equivaldria al 255 decimal). Aquest format, a pesar que pot semblar el més enrevessat, és en realitat el que més s'utilitza, perquè cada color es representa així amb 6 dígits hexadecimals, precedits per un coixinet, la qual cosa fa fàcil identificar els colors en la fulla d'estils. Així, el negre es representaria per #000000, el blanc per #FFFFFF, i el roig per #FF0000. L'exemple anterior quedaria així:

```
p { color: #0000FF; }
```

Podem trobar en Internet diverses pàgines amb codis de color, de manera que fent clic en el color que vulguem, ens apareixerà el codi hexadecimal que hem d'afegir a la CSS.

Exercici 8

Crea un document HTML anomenat **colorsCSS.html**. Dins, embegut en el *head*, afeg una etiqueta *style* per a definir tres estils amb selector de classe : un anomenat *groc* per a definir un color de text groc, un altre anomenat *roig* per a definir un color de text roig, i un altre anomenat *verd* per a definir un color de text verd. Utilitza per a això format RGB hexadecimal.

Després, afeg tres paràgrafs al document, cadascun amb un estil (i per tant, un color) diferent.

Els divisors HTML

Veurem ara una etiqueta que ha resultat ser tremendament útil des d'HTML 4, a pesar que al principi no s'utilitzava molt. Es tracta de l'etiqueta **div**. S'utilitza per a definir una regió, i dividir així el document en regions diferents. Per exemple, si volem tindre dues regions diferents en un document, podríem utilitzar una taula amb dues cel·les, com veurem en una altra secció, o com faríem en Word.

No obstant això, l'ús de taules per a estructurar la informació està desaconsellat o obsolet en HTML, ja que són un mecanisme massa rígid per a organitzar el text en la pàgina. Què passaria si canviem l'amplària de la finestra? La taula continuaria intentant ocupar el seu espai, i els textos en les caselles s'anirien acurtant i anant cap avall. Per a solucionar aquest problema, i facilitar un disseny més *responsive* (sensible als canvis de grandària de la pantalla), s'utilitzen els divisors.

Així, per a crear dues regions diferents en una pàgina, podríem tindre un codi com aquest:

```
<html>
...
<body>
  <div>Una regió</div>
  <div>Una altra regió</div>
</body>
</html>
```

Si provem això en una pàgina i ho visualitzem en el navegador, el que tindrem serà una mica com això:

Una regió

Una altra regió

És a dir, l'efecte ara com ara és similar a si haguérem utilitzat dos paràgrafs. Però l'avantatge d'aquests divisors és que, amb CSS, tenim moltíssimes possibilitats per a col·locar i donar grandària a aquestes dues regions, com veurem en parlar del model de caixes CSS a continuació.

El model de caixes en CSS

El model de caixes en CSS és un dels aspectes més complicats de dominar, però que més versatilitat pot donar a les nostres webs. Aquest model es basa en el fet que cada element d'un document HTML (cada paràgraf, cada *h1*, cada *div*, etc.) té una caixa, un rectangle que l'engloba i el separa de la resta.

Encapçalat 1

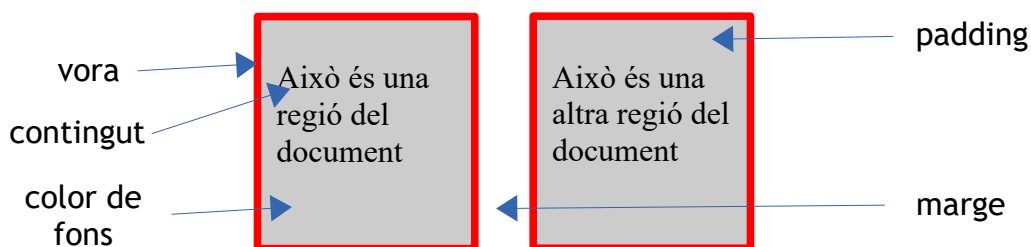
Paràgraf amb el seu contingut. Paràgraf amb el seu contingut. Paràgraf amb el seu contingut. Paràgraf amb el seu contingut. Paràgraf amb el seu contingut. Paràgraf amb el seu contingut.

Encapçalat 2

Un altre paràgraf.

Així, CSS permet definir les característiques de cadascuna d'aquestes caixes (altura, amplària, posició, color...). Per a això, hem de tindre en compte que cada caixa es compon de 7 elements:

- El **contingut** HTML que tinga
- La **vora**, una línia (visible o no) que tanca al contingut
- El **farcit** o **padding**, que és l'espai entre el contingut i la vora
- El **marge** (**margin**), que és l'espai entre la caixa i altres caixes adjacents.
- La **imatge de fons** (**background-image**), si hi haguera alguna
- El **color de fons** (**background-color**), si hi haguera algun, i que sempre quedarà per davall de la imatge de fons
- La **grandària** de la caixa (amplària i altura)



Propietats CSS de les caixes

Cada característica de les caixes té unes propietats en CSS per a poder-se configurar:

- Per al **marge**, podem usar la propietat **margin** i definir separats per espais els marges dels 4 costats (en l'ordre a dalt, dreta, a baix, esquerra), o usar les

propietats **margin-top**, **margin-right**, **margin-bottom** i **margin-left** i definir cada marge per separat. Estes 2 regles són equivalents:

```
p { margin: 0 10px 0 20px; }  
p { margin-right: 10px; margin-left: 20px; }
```

Si en la propietat *margin* usem un sol valor, els 4 marges prenen aqueix mateix valor. Si usem dos valors, el primer s'aplica a dalt i a baix, i el segon a esquerra i dreta. Si usem tres valors, el primer és per a dalt, el segon per a esquerra i dreta, i el tercer per a baix.

- Per al **farciment** (*padding*), tenim propietats similars al cas del marge: una propietat global que es diu **padding**, i quatre propietats per a cadascun dels costats: **padding-top**, **padding-right**, **padding-bottom** i **padding-left**, que s'utilitzen de la mateixa forma que les que hem vist per al marge.
- Per a la **vora**, podem modificar tres propietats d'aquest: el seu estil (simple, doble, puntejat, etc.), el seu color i el seu gruix. Per a això, tenim les propietats **border-style**, **border-color** i **border-width**, respectivament. Entre els estils més utilitzats està el simple continu (*solid*), el doble (*double*), el puntejat (*dotted*) o cap (*none*, valor per defecte). Per exemple, per a una vora d'un *h1* de color roig i traç doble, amb gruix de 5 píxels, posaríem:

```
h1 { border-style: double; border-color: red; border-width: 5px; }
```

Podem definir estils de vora per a cada costat, col·locant el costat enmig de la propietat. Per exemple, per a modificar el color de la vora inferior, posaríem **border-bottom-color**, i per a canviar el gruix de la vora dreta, posaríem **border-right-width**. Així, el següent exemple estableix una vora inferior i dret simple, de color blau i 2 píxels de gruix:

```
h1 { border-bottom-style: solid; border-right-style: solid;  
    border-bottom-color: blue; border-right-color: blue;  
    border-bottom-width: 2px; border-right-width: 2px; }
```

També podem usar les propietats *border-top*, *border-bottom*, *border-left* i/o *border-right* (i fins i tot la propietat *border* a seques) i definir-lis les seues tres propietats (estil, gruix i color) separades per espais:

```
h1 { border-bottom: solid 2px blue; }
```

- Per al **color de fons**, tenim la propietat **background-color**. El color s'especifica igual que en la propietat *color*. Per exemple:

```
p { background-color: #FF2233; }
```

- Per a la **imatge de fons**, tenim la propietat **background-image**, a la qual hem de dir-li la ruta (normalment relativa a la carpeta de l'aplicació web) on està la imatge a utilitzar, precedida de la funció *url*, i entre parèntesi i cometes dobles.

```
h1 { background-image: url("../imatges/fons.png"); }
```

Si la imatge no és prou gran, podem indicar que aquesta es repetisca vertical i/o horitzontalment, afegint la propietat **background-repeat**, que pot valdre *repeat-x* (repetir horitzontalment), *repeat-y* (repetir verticalment), *repeat* (repetir en totes direccions) o *no-repeat* (no repetir, valor per defecte)

```
h1 { background-image: url("imatges/fons.png");  
    background-repeat: repeat-x; }
```

- Finalment, podem canviar la **grandària** de la caixa modificant les seues dimensions (amplària i altura) amb les propietats **width** i **height**, utilitzant el tipus de mesura que vulguem (normalment, percentatges o píxels).

```
h1 { width: 70%; height: 50px; }
```

També existeixen les propietats **max-width**, **min-width**, **max-height** i **min-height** per a establir uns rangs d'amplària i altura variables.

Exercici 9

Crea un document anomenat **caixesCSS.html** que tinga embeguda una etiqueta *style* (o utilitzant un fitxer estil9.css) amb un selector de classe anomenat *caixa*, que definisca els següents estils per a una caixa:

- Marge esquerre i dret del 20%, superior de 30 píxels i inferior de 0
- Farcit (*padding*) de 10 píxels a cada costat
- Vora superior roja de 4 píxels de gruix, dret blau de 6 píxels i inferior verd de 2 píxels, tots amb estil sòlid.
- Color de fons groc

Finalment, crea un paràgraf (més o menys llarg, d'unes 3 o 4 línies almenys) i aplica-li aquest estil.

Posicionament de les caixes

A l'hora de col·locar cadascuna de les caixes (elements HTML) en la pàgina, el navegador segueix una sèrie de pautes, segons les característiques de cada caixa i l'estil que tinga configurat en el CSS. Existeixen diferents tipus de posicionament:

- Normal:** el que utilitza el navegador per defecte, si no se li indica el contrari en el CSS. Depèn de les característiques de cada caixa (si és un element en línia o de bloc, si la seua grandària permet tindre un altre element al costat, el posicionament que té l'element que el conté, etc.)
- Relatiu:** es desplaça la caixa respecte al que seria el seu posicionament normal
- Absolut:** la caixa se situa en una posició absoluta respecte del seu element contenidor
- Fix:** la caixa se situa en una posició fixa en pantalla, independentment de si l'usuari puja o baixa la pàgina.
- Flotant:** la caixa se situa tot el possible a l'esquerra o dreta, dins de la línia horitzontal en què es troba.

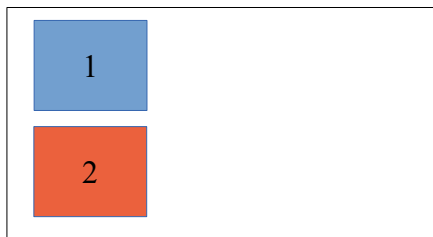
El tipus de posicionament s'estableix amb la propietat **position**, que pot valdre qualsevol dels tipus anteriors, excepte el flotant: *static* (normal), *relative*, *absolute* o *fixed*. Per exemple, la següent caixa té un posicionament relatiu 20 píxels per davall i 10 píxels més a la dreta del seu posicionament normal:

```
div { position: relative; top: 20px; left: 10px }
```

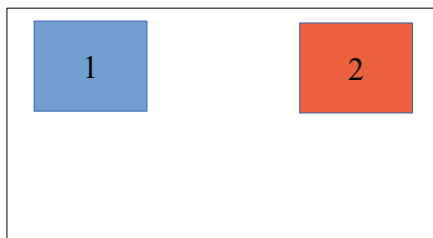
En el cas del posicionament flotant, s'especifica amb la propietat **float**, que pot valdre *left*, *right* o *none*.

- Si no existeix cap mena de posicionament flotant, les caixes (els elements de bloc, com *h1*, *p*, *div*, etc.) es posicionen un davall de l'altre
- Si algun element té posicionament flotant a esquerra o dreta, queda alineat a l'esquerra o dreta de la línia que ocupa. Si són diversos els que tenen aqueix mateix posicionament, queden col·locats l'un al costat de l'altre, fins a completar la línia (es continua en la següent línia si no hi ha més espai).

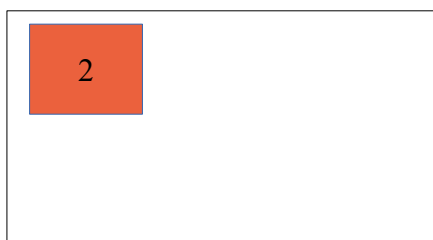
Vegem alguns exemples:



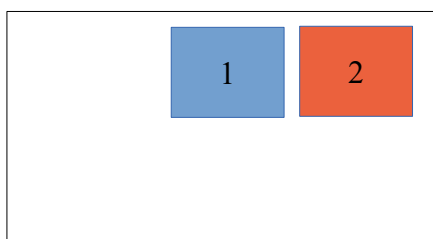
En aquest primer cas, les caixes 1 i 2 no tenen cap mena de posicionament flotant, així que es col·loquen una davall de l'altra. Les caixes poden ser qualsevol element de bloc (*h1*, paràgrafs, *div*, *blockquote*...)



En aquest cas, la caixa 2 té un posicionament flotant a la dreta. La caixa 1 pot tindre un posicionament flotant a l'esquerra, o cap (és indiferent).



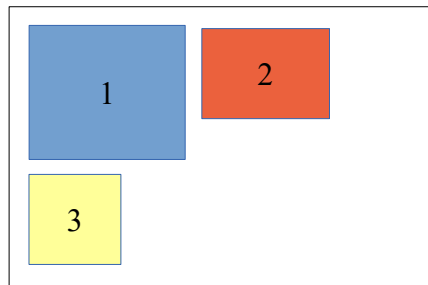
En aquest cas, la caixa 2 té un posicionament flotant a l'esquerra, i la caixa 1 no té posicionament, amb el que queda per darrere de la caixa 2 (tapada).



En aquest últim cas, totes dues caixes tenen un posicionament flotant a la dreta. L'ordre en què les posem en la pàgina HTML serà l'ordre en què apareguen en el navegador (en aquest cas, s'hauria posat primer la caixa 2 i després la 1).

En el cas que necessitem que una caixa quede pegada a la vora esquerra o dreta i que cap caixa li "destorbe" per a col·locar-s'hi allí, podem netejar la línia amb la propietat **clear**, que pot valdre *left*, *right* o *both*, depenent de quin costat vulguem ajustar les caixes. Per exemple, en el següent cas, si volem que la caixa 3 quede ajustada al marge esquerre, li posaríem en el seu estil:

```
.caja1 { float: left; }  
.caja2 { float: left; }  
.caja3 { float: left; clear:left }
```

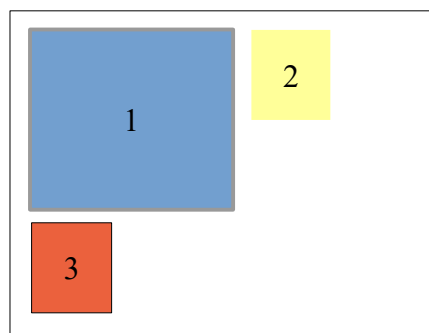


Exercici 10

Crea un document HTML anomenat **caixes.html**. Dins, crea tres caixes (tres *div*) i, amb l'estil embegut en una etiqueta *style* (o utilitzant un fitxer estil10.css), fes que:

- La primera caixa tinga color de fons blau, vora grisa contínua de 2 píxels de gruix per tot el contorn i marge dret i inferior de 10 píxels. Ha de tindre una amplària (*width*) del 50% i una altura (*height*) de 100 píxels.
- La segona caixa tinga color de fons groc clar, sense vora. Ha de tindre una amplària del 20% i una altura de 50 píxels.
- La tercera caixa ha de tindre color de fons roig clar, amb vora negra doble d'1 píxel de gruix. Ha de tindre una amplària del 20% i una altura de 50 píxels.
- Les tres caixes han de tindre un posicionament flotant a l'esquerra, però la tercera ha de tindre una propietat *clear* per a estar alineada amb el marge esquerre.

Al final, ha de quedar-te alguna cosa semblança a això (encara que amb altres proporcions segurament):



Prova de canviar la grandària de la finestra per a comprovar com s'ajusta l'amplària de les caixes a la nova grandària.

Altres elements HTML

Ara que ja sabem una mica d'HTML i de CSS, aprofundirem una mica més en el primer, i conèixer algunes etiquetes més per a construir continguts més variats. En aquest apartat aprendrem a crear diferents tipus de llistes, a col·locar imatges en pàgines web, a crear taules i, una cosa molt important, posar enllaços en pàgines web per a anar a altres pàgines o recursos.

Llistes

En HTML podem crear tres tipus de llistes:

- **Numerades o ordenades.** Van tancades entre una parella d'etiquetes **ol**. Cada element de la llista va entre una parella d'etiquetes **li**. Per exemple:

```
<html>
...
<body>
...
<p>Classificació de lliga:</p>
<ol>
  <li>At. Madrid</li>
  <li>Barcelona</li>
  <li>Reial Madrid</li>
</ol>
...
```

que es veuria més o menys així:

Classificació de lliga:

1. At. Madrid
2. Barcelona
3. Reial Madrid

- **Sense numerar.** Van tancades entre una parella d'etiquetes **ul**, i dins es tenen etiquetes **li** per a cada element de la llista, com en el cas anterior. Exemple:

```
<html>
...
<body>
...
<p>Repartiment d'actors:</p>
<ul>
  <li>Mark Ruffalo</li>
  <li>Scarlett Johanson</li>
  <li>Matt Damon</li>
</ul>
...
```

que es veuria més o menys així:

Repartiment d'actors

- Mark Ruffalo

- Scarlett Johanson
- Matt Damon

- Llistes de definicions . Van tancades entre un parell d'etiquetes **dl**, i estan formades per parells d'elements: un terme (tancat entre un parell d'etiquetes **dt**) i la seua definició (tancat entre un parell d'etiquetes **dd**). S'utilitzen per a fer llistes a manera de diccionaris, amb paraules i les seues definicions o traduccions. Per exemple:

```
<html>
...
<body>
...
<p>Vocabulari en anglés:</p>
<dl>
  <dt>Stylesheet</dt>
  <dd>Fulla d'estil</dd>

  <dt>Markup</dt>
  <dd>Marcat, etiquetat</dd>
</dl>
```

que es veuria més o menys així:

Vocabulari en anglés:

Stylesheet

Fulla d'estil

Markup

Marcat, etiquetatge

A vegades ens pot interessar fer llistes una mica més complexes. HTML permet niar llistes dins d'unes altres, siguen del tipus que siguen. N'hi ha prou amb començar una llista a continuació d'un element de la llista contenidora. Per exemple:

```
<html>
...
<body>
...
<p>Llistat de planetes:</p>
<ul>
  <li>Interiors:</li>
  <ol>
    <li>Mercuri</li>
    <li>Venus</li>
    <li>Terra</li>
    <li>Mart</li>
  </ol>
  <li>Exteriors:</li>
  <ol>
    <li>Júpiter</li>
    <li>Saturn</li>
    <li>Urà</li>
    <li>Neptú</li>
  </ol>
</ul>
```

que es veuria més o menys així:

Llistat de planetes:

- Interiors:
 1. Mercuri
 2. Venus
 3. Terra
 4. Mart
- Exteriors:
 1. Júpiter
 2. Saturn
 3. Urà
 4. Neptú

Exercici 11

Crea una pàgina anomenada **llistes.html** que tinga el següent contingut:

Saga Harry Potter

1. Harry Potter y la piedra filosofal
2. Harry Potter y la cámara secreta
3. Harry Potter y el prisionero de Azkaban
4. ...

Asignaturas

- Implantación de Aplicaciones Web
- Administración de S.O.
- Servicios en red
- ... etc.

Temario

1. Introducción a las Aplicaciones Web
2. Configuración del entorno
3. Diseño web con HTML y CSS
 - Estructura de un documento HTML
 - Textos básicos y encabezados
 - Estilos básicos de texto
 - El modelo de cajas...
 - ...
4. Javascript
5. ...

Veiem que hi ha tres tipus de llistes: ordenada, sense ordenar i múltiple. Cadascuna d'elles va precedida per un encapçalat de nivell 1 (*h1*). Posa-li com a títol (*title*) a la pàgina el text "Practicar amb llistes".

Exercici 12

Crea una pàgina anomenada **prehistoria.html**, que tinga el següent contingut:

El concepto de Prehistoria y su periodización

La **Prehistoria** es la ciencia que trata de la historia del mundo y del hombre antes de cualquier documento escrito. Se **diferencia** así de la Historia en que no queda tradición oral ni escrita de ella (sólo restos materiales arqueológicos), y en que su duración es de más de 1 millón de años, frente a los 6.000 que abarca la Historia.

¿Cómo podemos subdividir la Prehistoria?

- Edad de Piedra
 - Paleolítico (Piedra Antigua)
 - Mesolítico o Epipaleolítico
 - Neolítico (Piedra Nueva)
- Edad de Bronce
 - Calcolítico, Edad del Cobre o Eneolítico
 - Bronce Antiguo
 - Bronce Medio
 - Bronce Reciente
- Edad de Hierro (*Protohistoria*)
 - Hierro I
 - Hierro II

El títol (*title*) de la pàgina ha de ser "Prehistòria", amb codificació ISO-8859-1 o UTF 8 com en altres exercicis. L'encapçalat principal ("El concepto de Prehistoria y su periodización") és un *h1*, seguit d'un paràgraf explicatiu (compte amb les negretes). Després ve un *h2* amb un altre encapçalat ("Com podem..."), seguit d'una línia horitzontal, i després una llista amb dos nivells.

Enllaços

Els **enllaços** (també anomenats enllaços, enllaços o *links*) són potser l'element més important de les pàgines web, ja que permeten "saltar" d'una pàgina a una altra, fent clic en ells. Un altre element que ens permet canviar de pàgina són els formularis, quan els emplenem o premem algun dels seus botons, però això no són enllaços.

Els enllaços existeixen en quasi qualsevol pàgina web. Per exemple, si fem una cerca en Google, en fer clic en algun dels resultats que ens ofereix, estem fent clic en un enllaç.

A l'hora de col·locar un enllaç en una pàgina, hem de distingir dos elements:

- El **text** (o imatge) sobre el qual punxem en la pàgina HTML, i que pot ser el que nosaltres vulguem.
- La **URL** (adreça web) a la qual volem anar en punxar en aqueix enllaç. Aquesta direcció haurà de ser la correcta de la pàgina que volem visitar.

Per a afegir enllaços en un document HTML s'utilitza la parella d'etiquetes **a**. Tenen un únic atribut obligatori, anomenat **href**, que serà la URL de la pàgina a la qual volem anar. El text (o la imatge) que vulguem posar per a fer clic sobre ell, haurà d'anar entre el parell d'etiquetes **a**. Vegem un exemple:

```
<p>Anar al <a href="http://www.ua.es">web de la Universitat d'Alacant</a>. </p>
```

que es veuria així:

Anar al [web de la Universitat d'Alacant](http://www.ua.es).

Altres atributs útils

L'etiqueta **a** admet, a més de **href**, altres atributs que poden ser-nos útils. Destacarem dos d'ells:

- L'atribut **title**, que serveix perquè, quan passem el ratolí per damunt de l'enllaç (sense fer clic en ell), ens aparega un requadre explicatiu sobre on anirem amb aqueix enllaç. El text del requadre serà el que posem en aquest atribut **title**.
- L'atribut **target**, que serveix per a indicar on volem obrir l'enllaç. Si no posem aquest atribut, l'enllaç s'obrirà en la mateixa finestra de la pàgina que tenim actualment. Però podem donar-li altres valors, com per exemple **"_blank"**, perquè s'obriga en una finestra nova.

El següent exemple modifica l'enllaç anterior, afegint-li un **title** i fent que s'obriga en una finestra nova:

```
<p>Anar a la <a href="http://www.ua.es" title="Anar al web de la UA" target="_blank">web de la Universitat d'Alacant</a>. </p>
```

Rutes relatives i absolutes

Quan volem anar a una URL d'una web diferent a la nostra, posem l'adreça completa (com el cas de l'exemple anterior). Però si volem saltar a una altra pàgina de la nostra mateixa web, podem indicar en l'atribut **href** una ruta relativa o absoluta a la pàgina on estem. Per exemple, si tenim una aplicació web de diverses pàgines, repartides en carpetes i subcarpetes, com en aquest exemple:

```
ElMeuWeb
├── pagina1.html
├── pagina2.html
└── altres
    ├── pagina3.html
```

Suposem que estem editant la *pagina1.html*. Si volguérem posar un enllaç per a anar a la *pagina2.html*, podríem fer-ho de dues formes:

- Com una ruta **relativa** des de la carpeta on està la *pagina1* que estem editant. En aquest cas, com totes dues pàgines estan en la mateixa carpeta, l'enllaç seria una cosa així:

```
<a href="pagina2.html">Anar a pàgina 2</a>
```

- Com una ruta **absoluta** a l'arrel de l'aplicació. Com la *pagina2* està en la carpeta arrel de l'aplicació *ElMeuWeb*, l'enllaç quedaria així:

```
<a href="/pagina2.html">Anar a pàgina 2</a>
```

En canvi, si des de *pagina1* volguérem posar un enllaç a *pagina3.html*, els enllaços relatiu i absolut quedarien, respectivament:

```
<a href="altres/pagina3.html">Anar a pàgina 3</a>  
<a href="/altres/pagina3.html">Anar a pàgina 3</a>
```

En el cas contrari (un enllaç en la *pagina3* per a anar a *pagina1*), els enllaços relatiu i absolut, respectivament, serien així:

```
<a href="../pagina1.html">Anar a pàgina 1</a>  
<a href="/pagina1.html">Anar a pàgina 1</a>
```

En general, tots dos tipus de rutes són equivalents, encara que les rutes absolutes poden donar-nos problemes en servidors de proves, ja que l'arrel de l'aplicació web no sol coincidir amb l'arrel del servidor (el servidor sol tindre l'arrel en */localhost/* i l'aplicació web en una subcarpeta dins d'aquesta, per exemple */localhost/ElMeuWeb/*), i en posar la barra "/" en la ruta absoluta els servidors interpreten l'arrel del servidor.

Exercici 13

Crea un document HTML anomenat **instituts.html**. Dins, genera una llista (no ordenada) amb el nom d'almenys 4 o 5 instituts de Cicles Formatius o Secundària que conegues, i posa un enllaç en cadascun que vaja a la seua corresponent pàgina web, fent que s'obriguen en una finestra nova (diferent a l'actual amb el llistat).

Punts d'ancoratge i enllaços interns

A més de poder enllaçar amb altres pàgines i altres webs, els enllaços també permeten saltar a altres zones dins d'una mateixa pàgina. Per a això, hem de seguir dos passos:

1. Anar a la zona a la qual volem saltar, i definir un **punt d'ancoratge** (en anglès, *anchor*). Per a això, usem la mateixa etiqueta *a dels* enllaços, però en aquest cas li posarem un atribut *name* amb el nom que vulguem (sempre que no es repetisquen en la mateixa pàgina). Per exemple:

```
<body>  
...  
<a name="introduccio"></a>  
<h1>Introducció</h1>  
<p>blablablabla...</p>
```

2. Després, anem on vulguem posar l'enllaç, i utilitzem la mateixa etiqueta *a dels* enllaços. En l'atribut *href* posarem, en lloc d'un nom d'arxiu o una URL, el nom de l'ancoratge que hem usat abans, precedit d'un coixinet "#". Per a l'exemple anterior, podríem anar al principi de la pàgina i posar un enllaç com aquest:

```
<body>  
<a href="#introduccio">Introducció</a>
```

```
...  
<a name="introduccio"></a>  
<h1>Introducció</h1>  
...
```

Exercici 14

Crea una pàgina anomenada **harrypotter.html** amb la següent estructura:

- Títol (title): Harry Potter
- Introducció (encapçalat nivell 1)
- Un xicotet paràgraf o dos que diguen de què va la saga
- Episodis (encapçalat nivell 1)
 - Harry Potter i la pedra filosofal (nivell 2)
 - Harry Potter i la cambra secreta (nivell 2)
 - Harry Potter i el presoner d'Azkaban (nivell 2)
 - Harry Potter i el calze de foc (nivell 2)
 - Harry Potter i l'ordre del Fènix (nivell 2)
 - Harry Potter i el misteri del Príncep (nivell 2)
 - Harry Potter i les relíquies de la mort (nivell 2)
 - Dins de cada apartat de nivell 2, afig 1 o 2 paràgrafs que contenen de què va l'episodi
- Resum de personatges (encapçalat nivell 1)
 - Dins tindrà una llista no numerada amb els noms de 5 dels actors i, entre parèntesis, a quin personatge interpreten

Posa ancoratges en cada encapçalat (de nivell 1 i 2) i al principi de tot un índex d'enllaços que permeten, punxant-los, anar a cada apartat de la pàgina. Al final, haurà de quedar-te alguna cosa com això:

1. [Introducción](#)
2. [Episodios de la saga](#)
 1. [Harry Potter y la piedra filosofal](#)
 2. [Harry Potter y la cámara secreta](#)
 3. [Harry Potter y el prisionero de Azkaban](#)
 4. [Harry Potter y el cáliz de fuego](#)
 5. [Harry Potter y la orden del Fénix](#)
 6. [Harry Potter y el misterio del príncipe](#)
 7. [Harry Potter y las reliquias de la muerte](#)
3. [Resumen de personajes](#)

Introducción

blablabla...

Episodios de la saga

Harry Potter y la piedra filosofal

blablabla...

Taules

Les taules són un element molt útil en els processadors de text perquè permeten estructurar la informació en la pàgina de diferents formes, simulant columnes, o permetent textos a diferents nivells o en diferents orientacions.

Fins fa no gaire, també s'utilitzaven amb molta freqüència en les pàgines web per a estructurar d'aquesta mateixa forma el contingut, però la W3C les desaconsella o les considera obsoletes, perquè fan que el disseny de la pàgina siga molt rígida (una taula no canviarà la seua disposició per a poder-se veure millor en un mòbil, per exemple), i dificulten la lectura pels lectors de pantalla.

En qualsevol cas, veurem breument com es creen taules en HTML, tant per si ens és necessari crear-les en algun moment, com per a saber-les identificar en qualsevol pàgina web que estiguem manipulant.

El contingut d'una taula va entre la parella d'etiquetes **table**. Dins, distingim dos elements:

- Cada **fila** de la taula, es marca amb un parell d'etiquetes **tr**.
- Cada **cel·la** dins d'aqueixa fila, es marca amb un parell d'etiquetes **td**. El contingut d'aqueixa cel·la serà el que posem entre aqueix parell d'etiquetes **td**, i pot ser qualsevol cosa (text pla, paràgrafs, llistes, imatges, una altra taula...).
 - La primera fila de la taula, opcionalment, es pot marcar amb etiquetes **th** en lloc de **td**, per a indicar que és la capçalera de la taula (i així poder-li posar un estil diferent).

Vegem un exemple:

```
<table>
```

```
<tr>
  <td>Fila 1, Columna 1</td>
  <td>Fila 1, Columna 2</td>
</tr>
<tr>
  <td>Fila 2, Columna 1</td>
  <td>Fila 2, Columna 2</td>
</tr>
</table>
```

Això es veuria més o menys així en el navegador:

Fila 1, Columna 1	Fila 1, Columna 2
Fila 2, Columna 1	Fila 2, Columna 2

Atributs de les taules

Les etiquetes que usem per a construir taules (*table*, *tr*, *td*...) admeten una sèrie d'atributs per a afegir algunes propietats.

Així, l'etiqueta *table* admet els següents atributs:

- **border**: gruix de la vora. Si no volem tindre bord, posaríem una vora de 0.
- **width**: amplària de la taula (en píxels o en percentatge sobre l'amplària màxima)
- **cellspacing**: marge entre l'interior de la vora de cada casella i el contingut
- **summary**: resum del contingut de la taula, útil per a lectors de pantalla.

Per part seua, els elements *tr* i *td* admeten aquests atributs:

- **align**: alineació horitzontal del text. Pot valdre *left*, *right* o *center*.
- **valign**: alineació vertical del text. Pot valdre *top*, *bottom* o *middle*.
- **rowspan**: només parella etiquetes *td*, expandeix la cel·la X files més cap avall
- **colspan**: només parella etiquetes *td*, expandeix la cel·la X cel·les més cap a la dreta
- **width**: es pot utilitzar en etiquetes *td* per a indicar l'amplària específica d'aqueixa columna (en píxels o percentatge), encara que el W3C recomana usar CSS.

Vegem tot això en un exemple:

```
<table width="100%" border="1">
  <tr>
    <td rowspan="2" valign="middle" style="width:50%">Un</td>
    <td>Dos</td>
    <td>Tres</td>
    <td>Quatre</td>
  </tr>
  <tr>
    <td>Cinc</td>
    <td colspan="2" align="right">Sis</td>
  </tr>
</table>
```

Això es veuria més o menys així en el navegador:

U	Dos	Tres	Quatre
	Cinc	Sis	

Exercici 15

Crea una pàgina anomenada **taules.html**. En ella hauràs de crear dues taules:

- Una sense vores, amb un encapçalat de 4 columnes (th): Nom i cognoms, E-mail, Adreça, Telèfon, i després 4 files amb aqueixes dades emplenades per a 4 persones que conegues o t'inventes.
- Una altra taula amb vora de gruix 1 i amplària del 50%, on la primera fila tinga 3 columnes combinades i el text en negreta "Trilogies". Després, hi haurà informació sobre trilogies de cinema. Per a cadascuna, afegirem una fila amb les 3 columnes combinades i el títol de la trilogia (El Senyor dels Anells, El Hobbit, Matrix, etc), i després una altra fila amb els títols de les 3 pel·lícules de la trilogia

Nombre	E-mail	Direcció	Teléfono
Juan García	jgarcia@gmail.com	C/La Puerta, 15, 1ªA	611223344
Adrián Llopis	all1979@hotmail.com	Av. Holanda 41, 6ºD	612345678
Aitana Martínez	a.mart@yahoo.com	C/Blasco Ibáñez, 4, 4ªK	699887766
Felipe Sánchez	fel.san.jim@gmail.com	C/Juan Ramón Jiménez, 51, Bajo	615263748

Trilogías		
El Señor de los Anillos		
La comunidad del Anillo	Las dos torres	El retorno del Rey
El Hobbit		
Un viaje inesperado	La desolación de Smaug	La batalla de los cinco ejércitos
Matrix		
Matrix	Matrix Reloaded	Matrix Revolutions

Imatges

En un processador de textos a l'estil de Word o OpenOffice, per a col·locar una imatge que ens agrada, n'hi ha prou amb copiar-la i pegar-la en el document, si no la tenim disponible en alguna carpeta del nostre ordinador. Però per a afegir imatges en documents web, hem de tindre-les localitzades en la mateixa carpeta (o una subcarpeta) del nostra web.

Si són poques imatges, podem tindre-ho tot junt en una mateixa carpeta, però si l'aplicació web és gran, o hi ha diverses imatges, el més normal és crear una carpeta anomenada *imatgens* o *imgs*, i deixar dins les imatges que necessitem.

Una vegada tinguem ja guardada la imatge que volem posar, usarem l'etiqueta **img** per a col·locar-la en la pàgina HTML. Aquesta etiqueta té un únic atribut obligatori, anomenat **src**, que funciona igual que l'atribut *href* vist per als enllaços: hem d'indicar la ruta (absoluta o relativa) cap a la imatge que volem afegir. També podem posar una URL cap a

una imatge d'una altra pàgina, però correm el risc que la imatge deixi de veure's si la pàgina tanca o està caiguda.

Per exemple, si volem afegir la imatge *foto.png* que està dins de la subcarpeta *imgs*, podríem afegir-la amb alguna cosa com:

```

```

Observa que l'etiqueta *img* és una etiqueta que no té parella. Ella sola s'obri i tanca. Dins, a més de l'atribut *src*, podem tindre altres atributs opcionals:

- **width**: especifica l'amplària que volem que tinga la imatge. Li la podem donar en píxels o en percentatge. El millor sempre és no haver d'usar aquest atribut, i redimensionar la imatge a la grandària que vulguem, perquè ocupe menys espai i carregue més ràpid en la pàgina.
- **height**: especifica l'altura que volem que tinga la imatge (normalment en píxels). Si no posem res, es redimensiona segons l'amplària que hàgem posat. Igual que en el cas anterior, el millor és no haver de definir aquests paràmetres en l'etiqueta *img*, sinó redimensionar la imatge directament a la grandària que vulguem.
- **alt**: és un text alternatiu que mostrar en el cas que el navegador no puga carregar la imatge, o aquesta vesprada a carregar. En passar el ratolí per damunt de la imatge, es mostrarà aquest text alternatiu. Per a segons quines versions d'HTML o XHTML, aquest atribut també és obligatori (en cas contrari, la pàgina no serà correcta).

Icona de l'aplicació web

Últimament s'està posant cada vegada més de moda que les aplicacions (tant en mòbils com en ordinadors i fins i tot les aplicacions web) tinguin la seua pròpia icona que les identifique. De fet, en molts webs, quan carreguem la pàgina, apareix la seua icona en la barra d'adreces, o en la pestanya del navegador on hem obert el web (depenent del navegador). Com aconseguim això?

En primer lloc, com qualsevol altra imatge, haurem de tindre la icona localitzada en alguna carpeta o subcarpeta del nostre web. Normalment les icones tenen extensió *.ico* o *.png*, i en el cas de la icona de la web, se li sol cridar *favicon* (pel que l'arxiu sencer sol dir-se *favicon.ico* o *favicon.png*). La icona sol tindre unes dimensions de 16x16 o 32x32 píxels. Una vegada tinguem la icona localitzada, haurem d'afegir la següent línia en el *head* de la(s) pàgina(s) web on vulguem afegir la icona:

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon" />
```

Existeixn algunes variants. Per exemple, si la icona compleix els estàndards de grandària, podem posar *rel="icon"* i *type="image/png"* (en el cas que siga un arxiu PNG). També per a alguns navegadors és suficient amb deixar un arxiu anomenat *favicon.ico* en l'arrel de la nostra aplicació web, i ho detecten automàticament.

Exercici 16

Modifica l'exercici de Harry Potter que hem fet en parlar d'enllaços, i fes els següents canvis:

- Fes que aparega la portada de cada pel·lícula just després del seu títol. Busca les portades en Internet i copia-les en una subcarpeta *imgs* dins de l'exercici.

- Les imatges han d'aparèixer centrades, entre el títol de la pel·lícula i el seu resum, sense text als costats. Utilitza el CSS que consideres per a això.

Opcionalment, afeg també una foto de cada personatge de la llista del resum de personatges, a grandària 100x100, al costat del nom del personatge.

Exercici 17

Crea una pàgina anomenada **biografia.html**. Crearàs en ella la biografia d'un personatge famós (el que vulgues), de manera que et quede alguna cosa com això:

Groucho Marx



Groucho Marx (n. Nueva York, 2 de octubre de 1890 - m. Los Angeles, 19 de agosto de 1977) fue un actor, comediante y escritor estadounidense, conocido principalmente por ser uno de los miembros de la familia cómica Hermanos Marx.

Julius Henry Marx nació en Nueva York, creció en el seno de una modesta familia de inmigrantes alemanes judíos. Su madre era Miene (Minnie) Schoenberg (1864-1929) que emigró a Estados Unidos desde Alemania junto a sus padres y sus hermanos y su padre era Simon (Samuel) Marx (nacido como Marrix) (1860?-1933), inmigrante alemán, nacido en Alsacia cuando formaba parte de Francia y anexada posteriormente al Imperio Alemán tras la Guerra Franco-Prusiana en 1871, con lo que Samuel pasó de ser ciudadano francés a ciudadano alemán. Orgulloso de su pasado francés fue conocido toda su vida como Frenchy.

Marx falleció en Los Angeles a causa de una neumonía, dejando atrás tres ex-esposas, Ruth, Kay y Eden, tres hijos, Arthur, Miriam y Melinda, 18 películas, 14 de ellas junto a sus hermanos, y millones de admiradores. Una de sus célebres frases fue: "No deseo pertenecer a ningún club que acepte como socio a alguien como yo". Groucho Marx fue incinerado, sus cenizas se guardan en el Eden Memorial Park, siendo falso el epitafio "Perdonen que no me levante" que popularmente se creía que estaba en su tumba.

Com veus, hauràs de crear una taula, de vora 1, amb una columna i dues files. En la primera fila, centrada, haurà d'aparèixer la imatge del personatge que hages triat, a grandària 300 x 400. En la segona fila, hauran d'aparèixer almenys un parell de paràgrafs que contenen alguna cosa de la seua vida.

A més, la pàgina ha de tindre com a títol (*title*) el text "Biografia de XXXXX", sent XXXXX el nom del personatge que hages triat. Abans de la taula, ha d'haver-hi un encapçalat de primer nivell (*h1*) amb el nom d'aqueix personatge.

Elements de bloc i en línia

Ara que ja coneixem unes quantes etiquetes HTML (*h1*, *p*, *ul*, *img*...), parlarem d'un concepte relacionat amb totes elles, i que hem esmentat amb anterioritat en aquest llibre, sense entrar en els detalls. Totes aquestes etiquetes o elements HTML que estem aprenent es divideixen en dos grups:

- Els elements **de bloc**, que són etiquetes que, per defecte, constitueixen un bloc separat verticalment dels altres. És el cas de moltes etiquetes, com *h1*, *p*, *div*... Quan posem una d'aquestes etiquetes, el contingut que va just abans i just després d'elles pren un punt i a part, és a dir, hi ha una separació vertical entre aquests elements i els que els envolten.

- Els elements **en línia**, que són etiquetes que, per defecte, es mostren adjacents les unes de les altres, sense separació vertical. És el cas, per exemple, d'etiquetes com *strong*, *em*, o els enllaços. Quan posem alguna cosa en negreta, no es passa automàticament a la següent línia per a mostrar-ho, sinó que va tot seguit en la mateixa línia on estava.

Aquestes propietats de les etiquetes, com veurem, es poden modificar des de CSS, però convé fer-nos una idea que existeixen aquests dos tipus d'elements per a entendre millor com es col·loquen en el navegador les coses quan visualitzem una pàgina web.

Altres opcions de CSS

Veurem ara algunes propietats més que podem afegir en les regles CSS per a canviar l'estil dels elements.

Visibilitat

Podem modificar la forma en què es mostren els elements en les pàgines amb la propietat **display**, que pot prendre, entre altres, els següents valors:

- **none**: serveix per a eliminar l'element, i que els altres ocupen el seu lloc. Aparentment, aquesta opció no té cap utilitat. Per a què posarem un element en la pàgina si després no volem mostrar-lo? De moment és una opció inútil, però més endavant veurem que, combinat amb Javascript, pot ser una utilitat molt potent per a donar funcionalitat a la web.
- **block**: serveix per a mostrar l'element com si fora un element de bloc
- **inline**: serveix per a mostrar l'element com si fora un element en línia

Per exemple, si volem que un paràgraf de classe "linia" deixi de ser un element en bloc, i es mostre com un element en línia, posaríem:

```
p.linia { display: inline; }
```

Així, si tenim una pàgina HTML com aquesta:

```
<html>
...
<body>
...
<p>Un paràgraf normal.</p>
<p class="linia">Un paràgraf en línia.</p>
<p class="linia">Un altre paràgraf en línia.</p>
...
```

En el navegador es veuria:

Un paràgraf normal.

Un paràgraf en línia. Un altre paràgraf en línia

Notar com el primer paràgraf, en ser normal, té el seu punt i a part, però els altres dos no es separen entre si, en mostrar-se com a elements en línia.

Estils per a enllaços

Els enllaços són un element molt potent en una pàgina web, però el seu estil és difícil de controlar. De fet, res més posar l'enllaç, se'ns veurà de color blau en la pàgina, i quan fem clic sobre ell, la resta de vegades apareixerà de color morat, indicant que és un enllaç que ja hem visitat abans.

Imaginem que volem fer una pàgina web amb el color de fons blau. Què passaria amb els enllaços? A penes es veurien. O fins i tot si usem altres colors, pot ser que el color dels enllaços desentone amb el que hi ha voltant.

Per a canviar l'estil dels enllaços mitjançant CSS, és una mica més complicat que per a un element normal (com un paràgraf, per exemple), ja que els enllaços tenen diferents estats, i hem de canviar l'estil per a cadascun d'aqueixos estats.

L'estil per a l'estat normal d'un enllaç es canvia amb el selector *a:link* (igual que per a canviar l'estil dels paràgrafs usem el selector *p*). Per exemple, si volem posar el text de l'enllaç de color blanc:

```
a:link { color: white }
```

Però si fem clic en l'enllaç, com ja passarà a ser un enllaç visitat, la pròxima vegada que carreguem la pàgina ja no apareixerà blanc, sinó morat (color per defecte per a enllaços visitats). Per a l'estil per a enllaços visitats, usem el selector *a:visited*:

```
a:visited { color: white }
```

Adicionalment, l'enllaç té dos estats més sobre els quals podem actuar: quan passem el ratolí per damunt de l'enllaç, sense punxar-lo (s'accedeix amb el selector *a:hover*) i quan fem clic en l'enllaç (s'accedeix amb el selector *a:active*).

Per exemple, si volem que, en qualsevol estat, els enllaços es mostren de color blanc, hauríem d'afegir aquestes quatre regles al nostre CSS (en realitat bastaria amb les dues primeres, si no volem fer res en usar el ratolí en els enllaços):

```
a:link { color: white }  
a:visited { color: white }  
a:hover { color: white }  
a:active { color: white }
```

o bé una sola regla conjunta:

```
a:link, a:visited, a:hover, a:active { color: white }
```

encara que la primera opció és més versàtil si després volem afegir alguna opció extra en algun estat (per exemple, canviar la grandària de la lletra en passar el ratolí per damunt).

També podem, entre altres coses, actuar sobre el subratllat que hi ha en els enllaços, amb la propietat *text-decoration*. Si per exemple volem que els subratllats es lleven en passar el ratolí per damunt, faríem:

```
a:hover { color: white; text-decoration: none; }
```

Exercici 18

Continuarem modificant l'arxiu **harrypotter.html** que hem fet en exercicis anteriors. Afig un bloc *style* (si no ho has fet ja) en el *head* i afig aquests estils:

- Fes que tot el document (selector *body*) tinga tipus de lletra Arial i color de fons blau clar (en format hexadecimal, cerca un to que t'agradi).
- Fes que els enllaços (per a l'índex inicial) tinguin lletra de color blanc, sense subratllat, i que en passar el ratolí per damunt, o fer clic en l'enllaç, la lletra canvie a color taronja i subratllat.

Estils per a llistes

CSS també té una sèrie de propietats útils per a canviar l'aparença per defecte de les llistes. Destaquem les següents:

- **list-style-type:** serveix per a indicar l'estil de les llistes (tipus de vinyeta o número). Pot prendre els valors *disc* (cercles negres), *circle* (cercles blancs), *square* (quadrats), *decimal* (números), *lower-roman* i *upper-roman* (números romans en minúscules/majúscules), *lower-latin* i *upper-latin* (alfabet llatí en minúscules o majúscules, respectivament), o *none* (no usar vinyeta).

```
ol { list-style-type: lower-roman; }
```

- **list-style-image:** en el cas que vulguem usar una imatge com a vinyeta, podem utilitzar aquesta propietat. La imatge l'haurem de tindre guardada en alguna carpeta o subcarpeta de la nostra aplicació web, i per a carregar-la s'usa d'aquesta manera:

```
ul { list-style-image: url("imgs/llista.gif"); }
```

També podem usar les propietats de visibilitat (propietat *display* vista abans) per a fer que els elements de la llista es mostren en línia, un darrere l'altre, i no l'un davall de l'altre, utilitzant el valor *inline*. Això és útil per a fer menús horitzontals.

Exercici 19

Crea un document anomenat **menuLlistat.html**. En ell, utilitzant únicament CSS i llistes, genera una llista com aquesta. Fes que cada text siga un enllaç a la pàgina de Wikipedia del corresponent llenguatge de programació. L'enllaç haurà de ser de color negre sense subratllar, i quan se li passe el ratolí per damunt es posarà de color roig.

Java	C#	Python	PHP
------	----	--------	-----

Estils per a taules

Ja hem vist que les taules admeten una sèrie d'atributs per a modificar alguns aspectes, com la vora, amplària, marge, etc. Aquestes coses també es poden modificar des de les corresponents propietats CSS (*width*, *padding*, etc.), però en general el resultat no sol ser el mateix. Com l'ús de taules està desaconsellat tampoc aprofundirem molt en aquestes propietats, però vegem alguns exemples de com aconseguir aquests estils amb CSS.

Suposem que volem una taula del 50% d'amplària permesa, i borda d'1 píxel. Per a això, podem tindre un CSS com aquest:

```
.mitabla { width: 50%; border-style: solid; border-width: 1px; }
```

I després creem la taula així:

```
...  
<table class="mitabla">  
...
```

Normalment les vores no solen quedar del tot igual, i és possible que ens toque definir la vora de cada cel·la perquè es veja. També podem usar la propietat *border-collapse* de la taula per a ajuntar vores (de taula i de cada cel·la) i que siguin només un, i no es dupliquen:

```
.mitabla { width: 50%; border-style: solid; border-width: 1px;  
            border-collapse: collapse; }  
...
```

Més sobre selectors

Farem un volt més de rosca al que són els selectors CSS. En un apartat anterior hem vist que fonamentalment s'utilitzen quatre tipus: d'etiqueta, descendent, de classe i d'id . Però existeixen alguns altres que poden ser-nos útils. I a més, hem de tindre en compte la jerarquia i l'ordre a l'hora d'aplicar selectors, quan varis coincideixen per a un mateix element de la pàgina HTML.

Altres selectors avançats

Vamos a veure alguns selectors una mica més avançats, que pot vindre bé conèixer:

- El **selector de fills** és similar al descendent, però aplica l'estil només a etiquetes que penguin directament de la contenidora. Per exemple, per a aplicar un estil a negretes que estan en paràgrafs (sense etiquetes intermèdies), posarem:

```
p > strong { ... }
```

- El **selector adjacent** s'utilitza per a aplicar un estil a una etiqueta que siga germana adjacent d'una altra (és a dir, a continuació d'una altra). Per exemple, per a cursives que vagen darrere de negretes (però no dins d'elles) usem:

```
strong + em { ... }
```

- El **selector d'atributs** s'utilitza per a aplicar l'estil a les etiquetes que tinguen un atribut amb un cert valor. Posem l'etiqueta, i després entre claudàtors l'atribut i el valor que ha de tindre, entre cometes. Per exemple, si volem aplicar un estil a tots els enllaços que vagen a la pàgina de la Universitat d'Alacant, posaríem alguna cosa com:

```
a[href="http://www.ua.es"] { ... }
```

Conflictes entre selectors

Què ocorre quan un bloc de contingut HTML encaixa amb més d'un selector en la fulla d'estils CSS? Quin de tots s'aplica? Si els estils són complementaris (per exemple, en un se li canvia el color del text i en un altre la grandària), s'apliquen tots. Però si hi ha conflicte (per exemple, dos selectors volen canviar el color del text), és una situació complexa i, en general, s'aplica l'estil que més exactament s'ajusta. Per exemple, si tenim aquest paràgraf:

```
<p id="miID">Este és un paràgraf marcat amb un ID.</p>
```

D'aquests tres selectors, se li aplicaria el segon, perquè és el que més s'ajusta al que el paràgraf té.

```
p { color: red; }  
p#miID { color: green; }  
body { color: blue }
```

En qualsevol cas, convé deixar la fulla d'estils amb els estils més generals al principi (començant pel *body*), i amb els més específics al final (selectors de classe, o de ID, o de fills, etc.), de manera que primer es començaran a aplicar els estils generals, i després s'anirà afinant als específics en els elements que els tinguen.

Formularis

Fins ara, l'única forma que hem vist d'interactuar amb una web són els enllaços, fent clic en ells. Però existeix una altra opció més potent perquè l'usuari interactue amb una web: els **formularis**. Són una eina molt útil per a enviar informació a un servidor. Quan ens registrem en una web, emplenem una sèrie de dades (e-mail, contrasenya, etc.) mitjançant un formulari. Així doncs, un formulari és una sèrie d'elements (anomenats normalment "controls" o "camps") que col·loquem en una pàgina web perquè l'usuari pugui introduir informació. Veurem a continuació quins tipus d'elements pot tindre un formulari i com es col·loquen en la pàgina.

Estructura bàsica d'un formulari

Tot el contingut d'un formulari va tancat entre una parella d'etiquetes **form**. Aquesta etiqueta té, entre altres, els següents atributs:

- **action**: seria l'equivalent al *href* dels enllaços. Indica a quina pàgina volem anar quan emplenem el formulari i l'enviem. Podem utilitzar, com en els enllaços, una ruta relativa o absoluta a la pàgina.

El normal és que la pàgina a la qual anar en el *action* d'un formulari siga una pàgina escrita en llenguatge de servidor (PHP, JSP, ASP... segons el llenguatge que hàgem triat), ja que normalment el servidor ha de recollir la informació que envia l'usuari en el formulari, processar-la i fer alguna cosa amb ella. Per exemple, si ens estem registrant en la web, el servidor haurà de processar les nostres dades i donar-nos d'alta en la base de dades d'usuaris de la web.

- **method**: indica la forma en què s'envien les dades al servidor. Pot valdre *get* (les dades són visibles en la barra d'adreces del navegador quan els enviem) o *post* (les dades no són visibles des de la barra d'adreces, i per tant és una mica més segur).

```
<form action="mipagina.php" method="post">
... Contingut del formulari
</form>
```

Elements d'un formulari

Dins de les etiquetes *form* que engloben al formulari, podem posar diferents tipus de controls. Els més habituals són:

- **Quadres de text**: per a introduir en una línia una informació curta i concreta, com per exemple un e-mail o un password.
- **Àrees de text**: per a introduir textos llargs, de més d'una línia. Per exemple, el contingut d'una notícia.
- **Caselles de selecció**: caselles de verificació que es marquen i desmarquen
- **Botons de radi** o *radio buttons*: conjunt de botons o opcions entre els quals només pot haver-hi un seleccionat.

- **Llistes:** desplegable o fixes
- **Botons:** per a enviar el formulari, o esborrar les dades d'aquest, per exemple
- **Etiquetes:** donen informació d'altres controls, indicant per a què serveixen.

La major part d'aquests controls (quadres de text, caselles de selecció, botons de radi, botons normals) s'afigen amb l'etiqueta *input*. Altres controls necessiten d'altres etiquetes. Veurem totes elles a continuació.

L'etiqueta *input*

L'etiqueta *input* s'utilitza per a definir diferents tipus de controls, depenent del que valga el seu atribut *type*. Aquest atribut pot valdre:

- **text:** per a definir **quadres de text** (d'una sola línia).
- **password:** per a definir quadres de text emmascarats (per a contrasenyes)
- **file:** per a definir controls per a pujar **arxius** al servidor. Apareixerà un quadre de text i un botó perquè, en punxar en ell, puguem triar l'arxiu. Aquests controls no són de molta utilitat sense un llenguatge de servidor que els ajude a completar la tasca, com PHP, JSP o ASP.NET.
- **casella de selecció:** per a definir **caselles de verificació**
- **radio:** per a definir **botons de ràdio**
- **submit:** per a definir el **botó** que servirà per a enviar el formulari en prémer-lo
- **reset:** per al **botó** que servirà per a esborrar el contingut del formulari en prémer-lo
- **button:** per a definir un altre botó qualsevol, amb una altra funcionalitat. Aquesta última opció no és massa útil de moment, si no es combina amb llenguatges de client com Javascript.
- **hidden:** serveix per a definir camps ocults, que emmagatzemen informació juntament amb el formulari però no la mostren a l'usuari. Aparentment no tenen utilitat, però combinat amb llenguatges de servidor com PHP, poden ser de molta ajuda.

Una vegada triat el tipus de control, l'etiqueta *input* té una altra sèrie d'atributs que hem de tindre en compte:

- **name:** serveix per a donar un nom identificatiu al control. Ens serà molt útil per a identificar-ho després en els llenguatges de client o de servidor (Javascript, PHP...)
- **id:** l'identificador del control, igual que podem usar aquest atribut en altres etiquetes HTML. S'utilitza en CSS, i també per a algunes coses amb Javascript. Sovint s'utilitza al costat de *name*, donant als dos el mateix nom o valor.
- **value:** serveix per a donar un valor inicial amb el qual emplenar el control (fins que l'usuari li ho canvie)
- **size:** serveix per a indicar la grandària del control (en caràcters si és un quadre de text, o en píxels per a la resta)
- **maxlength:** serveix per a indicar el màxim nombre de caràcters que pot acceptar el control (en el cas de quadres de text).

- **checked**: serveix per a indicar, si és una *casella de selecció* o botó de radi, que està marcat. El seu valor sempre és *checked*, és a dir, s'usa així: *checked="checked"*
- **disabled**: serveix per a deshabilitar un control, i que l'usuari no pugui modificar el seu valor. S'usa sobretot per a donar informació a l'usuari, però sense que aquest la pugui modificar. El seu valor sempre és *disabled*, és a dir, s'usa de la següent forma: *disabled="disabled"*. La dada *disabled* no s'envia amb el formulari.
- **readonly**: similar a l'anterior, serveix perquè un control siga de només lectura i l'usuari no pugui modificar el seu valor. A diferència de l'anterior, el contingut no apareix atenuat, i sí que s'envia amb la resta del formulari.
- **alt**: igual que per a les imatges, serveix per a donar informació sobre per a què serveix aquest control en concret, de manera que en passar el ratolí per damunt se'ns mostri aquesta informació.

Vegem alguns exemples: un quadre de text que admet fins a 50 caràcters per a un e-mail, una casella de verificació per a veure si l'usuari està casat, dos botons de radi per a triar el seu idioma preferit, i un botó per a enviar el formulari.

```
<form action="mipagina.php" method="post">

  <p>Introdueix el teu e-mail:</p>
  <input type="text" name="email" size="20" maxlength="50" />
  <br />

  <input type="casella de selecció" name="casat" /> Casat
  <p>Tria el teu idioma preferit:</p>
  <input type="radio" name="idioma" value="angles" checked="checked"/>Anglès
  <br />

  <input type="radio" name="idioma" value="frances" />Francés
  <br />

  <input type="submit" value="Enviar dades" />
</form>
```

Això es veuria més o menys així en la pàgina:

Introdueix el teu e-mail:

☐ Casat

Tria el teu idioma preferit:

☒ Anglès

☐ Francés

Enviar Dades

Observa algunes coses interessants, com que els controls de tipus *radio* comparteixen el mateix *name* (perquè en prémer un se solten els altres), i que tenen un *value* que no es mostra en el formulari realment, i la utilitat del qual escapa a l'abast d'aquest tema.

Les etiquetes de formularis: *label*

Amb l'etiqueta **label** podem definir controls de tipus etiqueta, és a dir, controls que serveixen per a donar informació sobre altres controls. En realitat, en l'exemple anterior, no hauríem d'haver usat paràgrafs per a indicar que el primer quadre de text és per a escriure l'e-mail, o que els botons de ràdio són per a triar l'idioma preferit. Per a això estan les etiquetes.

L'etiqueta **label** té un atribut **for**, que serveix per a indicar el nom (atribut **name**) del control sobre el qual dona informació. Així, el formulari anterior es podria haver definit així:

```
<form action="mipagina.php" method="post">
  <label for="email">Introdueix el teu e-mail:</label>
  <br />
  <input type="text" name="email" size="20" maxlength="50" />
  <br />
  <input type="casella de selecció" name="casat" /> Casat
  <br />
  <label for="idioma">Tria el teu idioma preferit:</label>
  <br />
  <input type="radio" name="idioma" value="angles" checked="checked"/>Anglès
  <br />
  <input type="radio" name="idioma" value="frances" />Francès
  <br />
  <input type="submit" value="Enviar dades" />
</form>
```

Àrees de text

Per a introduir text que ocupe més d'una línia, no ens serveix l'etiqueta *input* amb el seu atribut *type="text"*. En el seu lloc, hem d'usar la parella d'etiquetes **textarea**. Aquesta etiqueta té tres atributs:

- **name** per a posar-li un nom al control, igual que féiem amb *input*
- **rows** per a indicar quantes files de text hi haurà
- **cols** per a indicar quantes columnes de text hi haurà

Si després s'ocuparen més files o columnes que les que indiquem, apareixerà una barra de desplaçament per a poder veure la resta del text escrit en l'àrea de text.

Ací tenim un exemple d'ús i com es veuria:

```
<textarea name="notícia" rows="5" cols="20">
Text de la notícia
</textarea>
```

Text de la notícia

Llistes

Podem definir dos tipus de llistes per a triar opcions en un formulari: llistes fixes (ocupen una altura determinada i deixen visibles X elements, encara que poden tindre més i veure'ls mitjançant barres de desplaçament) i llistes desplegable (només és visible un element, i els altres es mostren desplegant la llista). Ambdues es creen amb l'etiqueta **select**, que admet els següents atributs:

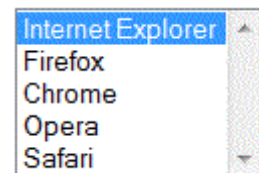
- **name**: per a donar-li un nom al control, igual que en *textarea* o *input*

- **size:** per a indicar el nombre de files visibles. Si no usem aquest atribut, o posem un valor d'1, crearem una llista desplegable. Si val més d'1, serà una llista fixa amb el nombre de files visibles que hàgem establert aquí. En aquest cas, podem utilitzar un atribut anomenat **multiple** (sense valor) que permet seleccionar diversos elements.

Internament, l'etiqueta *select* té una sèrie d'etiquetes **option**, una per a cada element que vulguem afegir a la llista. Aquests elements tenen un atribut **value** per a indicar el seu valor (que pot ser diferent del que després es mostrarà). A més, per a indicar si hi ha algun element preseleccionat per endavant, marcarem aqueix element amb l'atribut **selected="selected"**.

Vegem ací un exemple i com es veuria:

```
<select name="navegador" size="5">
  <option value="ie" selected="selected">
    Internet Explorer</option>
  <option value="firefox">Firefox</option>
  <option value="chrome">Chrome</option>
  <option value="opera">Opera</option>
  <option value="safari">Safari</option>
</select>
```



Exercici 20

Crea una pàgina anomenada **formDades.html** per a emplenar les teues dades personals. Afig un encapçalat 1 (*h1*) que diga "Emplena les teues dades", i davall un formulari com el que tens a continuació:

Rellena tus datos

Nombre

Apellidos

E-mail

DNI

Idioma materno

- ☐ Castellano
☐ Valenciano
☐ Otros

Subir mi foto Ningún archivo seleccionado

☐ Permitir que otros puedan ver mis datos

Algunes opcions avançades

Conjunts de camps o *fieldsets*

Podem agrupar conjunts de controls en una espècie de «marcs», per a poder dividir millor la informació del formulari. Aquests marcs es creen amb la parella d'etiquetes **fieldset**. Conté una subetiqueta **legend** que és el títol d'aqueix marc. Un exemple:

```
<fieldset>
  <legend>Dades personals</legend>
  <label for="nom">Nom</label>
  <br/>
  <input type="text" name="nom" />
  <br />
  <label for="cognoms">Cognoms</label>
  <br />
  <input type="text" name="cognoms" />
  <br />
  <label for="dni">DNI</label>
  <br />
  <input type="text" name="dni" size="10" maxlength="9" />
  <br />
</fieldset>
```

que es veuria així en el navegador:

Agrupar opcions en llistes

L'etiqueta **select** admet algunes opcions més per a agrupar els elements que conté en categories. Per a això s'utilitza l'etiqueta **optgroup**, amb un atribut *label* que dona nom a cada categoria. Vegem un exemple i com es veuria en el navegador:

```
<label for="programa">Programa seleccionat</label>
<br />
<select name="programa">
  <optgroup label="Sistemas Operativos">
    <option value="Windows">Windows</option>
    <option value="Mac">Mac</option>
    <option value="Linux" selected="selected">
      Linux</option>
    <option value="Un altre">Altre</option>
  </optgroup>
  <optgroup label="Navegadores">
    <option value="IE">Internet Explorer</option>
    <option value="Firefox">Firefox</option>
    ...
</select>
```

Exercici 21

Crea una pàgina anomenada **formProducte.html** amb el següent formulari (més l'encapçalat 1 amb el text "Informació sobre el producte", i un botó d'Enviar , centrat en la part inferior):

Información sobre el producto

Datos básicos:

Nombre

Descripción

Foto

☐ Añadir contador de visitas

Datos económicos:

Precio € Impuestos

Promoción

☒ Ninguno

☐ Transporte gratuito

☐ Descuento 5%