

HTML5 i CSS3

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/>

Algunes característiques d'HTML5

Tot el que hem vist en els apartats anteriors és compatible amb la versió HTML 4. No obstant això, amb l'estàndard HTML 5 s'han introduït molts canvis importants en l'estructura del document i la sintaxi d'algunes etiquetes, encara que es manté la compatibilitat amb les versions anteriors. Veurem ara alguns dels aspectes més importants que incorpora HTML5.

Canvis en les primeres línies del document

El nou *doctype*

Comentàvem quan véiem l'estructura bàsica d'un document HTML que en algunes versions (sobretot en documents XHTML) s'inclouïa abans de l'etiqueta *html* una etiqueta anomenada *DOCTYPE*, que especificava el tipus de document HTML que s'estava definint (versió, etc.), perquè el navegador el tinguera present. Aquesta etiqueta era molt difícil de memoritzar, i el normal era copiar-la i pegar-la d'un altre document que la tinguera, o usar un editor que l'emplenara automàticament en generar nous documents HTML. Vegem un exemple d'aquesta etiqueta, per a indicar que estem utilitzant HTML 4 estricta (més rigorós amb els errors de sintaxis):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Amb HTML5, aquesta etiqueta s'ha simplificat molt, i n'hi ha prou amb posar:

```
<!doctype html>
```

Observem com la paraula clau *doctype* passa a estar en minúscules, com la resta d'etiquetes, i es redueixen tots els seus atributs a indicar simplement que és un document de tipus HTML. Amb això és més senzill incloure aquesta etiqueta en els nostres documents. El no incloure-la no afectarà al fet que es veja correctament en els navegadors, però sí que pot afectar al fet que siga un document vàlid per a segons quin validadors i versions.

Indicar el joc de caràcters

Recordem també que, per a indicar el joc de caràcters utilitzat (una cosa molt útil per a després poder posar, per exemple, accents i enes palatals en documents en espanyol), utilitzàvem al principi del *head* una etiqueta *meta* amb una sintaxi com (per a un joc de caràcters ISO-8859-1, que és el que correspon a l'alfabet llatí):

```
<fique http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
```

Això també s'ha simplificat en HTML5, i n'hi ha prou amb posar:

```
<meta charset="iso-8859-1"/>
```

Compatibilitat amb alguns navegadors antics

Veurem a continuació que HTML5 incorpora algunes etiquetes que no existien en versions anteriors, com *section*, *article*, etc. En general, això el podria suportar qualsevol navegador anterior, sempre que en el CSS indicàrem quin estil tindran aquestes etiquetes,

però no ocorre així amb Internet Explorer, per a versions anteriors a la 9. Per a solucionar això, es pot afegir aquest text al final del *head*:

```
<!--[if lt IE 9]>  
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>  
<![endif]-->
```

El que estem fent és incloure un fitxer de script (veurem què és això en un altre tema), que "actualitza" Internet Explorer perquè sàpia reconèixer les noves etiquetes.

Canvis en l'estructura de la pàgina

Fins ara, podíem veure l'etiqueta *bodi* com una espècie de foli en blanc, de manera que tot el que posàrem dins d'ella (paràgrafs, llistes, etc.) s'anava col·locant, una cosa després d'una altra, amb els estils i característiques que definírem en el CSS. Amb HTML5, s'estructura la pàgina en una sèrie de compartiments, i definim cadascun d'aqueixos compartiments amb una etiqueta nova, com veurem a continuació.

L'etiqueta *header*

L'etiqueta **header** s'utilitza per a definir un títol o element introductori de qualsevol apartat de la pàgina. Típicament les pàgines tindran un *header* al principi, on posarem el títol de la pàgina amb algun logo, i el menú de navegació (enllaços). Però podem tindre altres *headers* en altres apartats de la pàgina que precisen de la seua pròpia introducció.

L'etiqueta *section*

L'etiqueta **section** s'utilitza per a definir cadascuna de les unitats de contingut de la pàgina. Per exemple, cadascuna dels apartats dividits en pestanyes d'una web, o les diferents seccions d'un periòdic en línia (esports, economia, etc.)

L'etiqueta *article*

L'etiqueta **article** s'utilitza per a definir apartat autònoms en un document. És una cosa similar a l'etiqueta *section*, amb el que és fàcil confondre el seu ús, però podem diferenciar-los aplicant la regla de la sindicació: si el que escriurem és un contingut que es pot publicar per si mateix en un altre lloc web, o difondre per si mateix mitjançant RSS (sense necessitat d'incloure contingut d'altres parts de la pàgina, o d'eliminar contingut que sobre), llavors és un article. Si no, és una secció. Exemples d'ús de article podrien ser notícies, entrades d'un blog, o missatges en un fòrum.

L'etiqueta *nav*

L'etiqueta **nav** s'utilitza per a definir grups de vincles (enllaços) que ens faciliten navegar pel lloc web. S'utilitza per a definir el menú principal de la pàgina, o altres submenús addicionals que puguem existir.

L'etiqueta *aside*

L'etiqueta *aside* s'utilitza per a definir blocs d'informació accessòria o tangencial, relacionada amb un altre contingut de la pàgina. Per exemple, si estem escrivint una notícia sobre alguna llei que ha aprovat el govern, podríem usar l'etiqueta *aside* per a reproduir alguna frase literal que haja dit algun ministre. També podem usar-ho per a incloure blocs de publicitat, o llistes secundàries d'enllaços.

L'etiqueta *footer*

L'etiqueta **footer** s'utilitza per a definir el peu per a una secció de contingut (amb el que podem tindre diversos *footer*, un per a cada secció, article o *aside*). Normalment conté informació de l'autoria de la secció, llistes d'enllaços relacionats (normes d'ús, política de privacitat, mapa del lloc, etc.). No és necessari que aparega al final de la secció (per exemple, en un blog, el nom de l'autor d'una entrada podria aparéixer al principi, sota el títol).

Exemple d'ús

Posant junt tot l'anterior, i utilitzant altres elements que ja coneixem d'HTML 4, podríem definir una pàgina amb la següent estructura:

```
<!doctype html>
<html>
<head>
  <meta charset="iso-8859-1" />
  <title>Periódico digital</title>
  <!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>
<body>
  <header>
    
    <nav>
      <ul>
        <li><a href="deportes.html">Deportes</a></li>
        <li><a href="economia.html">Economía</a></li>
      </ul>
    </nav>
  </header>
  <div id="principal">
    <div id="col1">
      <article>
        <h1>Se acaba la crisis</h1>
        <p>blablablabla...</p>
      </article>
      <article>
        <h1>Fabrican la vacuna para el ébola</h1>
        <p>blablablabla...</p>
        <h2>Podría suponer un Premio Nobel</h2>
        <p>blablablabla...</p>
      </article>
    </div>
    <div id="col2">
      <aside>
        
      </aside>
      <article>
        <h1>El próximo mundial se jugará en Siberia</h1>
        <p>blablablabla...</p>
      </article>
    </div>
  </div>
  <footer>
    <section id="autores">
```

```
<section>
  <h1>Ignacio Iborra</h1>
  <p>Profesor de Informàtica</p>
</section>
<section>
  <h1>Pepito Pérez</h1>
  <p>Diseñador web</p>
</section>
</section>
<section id="copyright">
  <p>&copy; 2014</p>
</section>
</footer>
</body>
</html>
```

Si carreguem aquesta pàgina sense CSS associat, es veuria una cosa així:

Listín Diario

- [Deportes](#)
- [Economía](#)

Se acaba la crisis

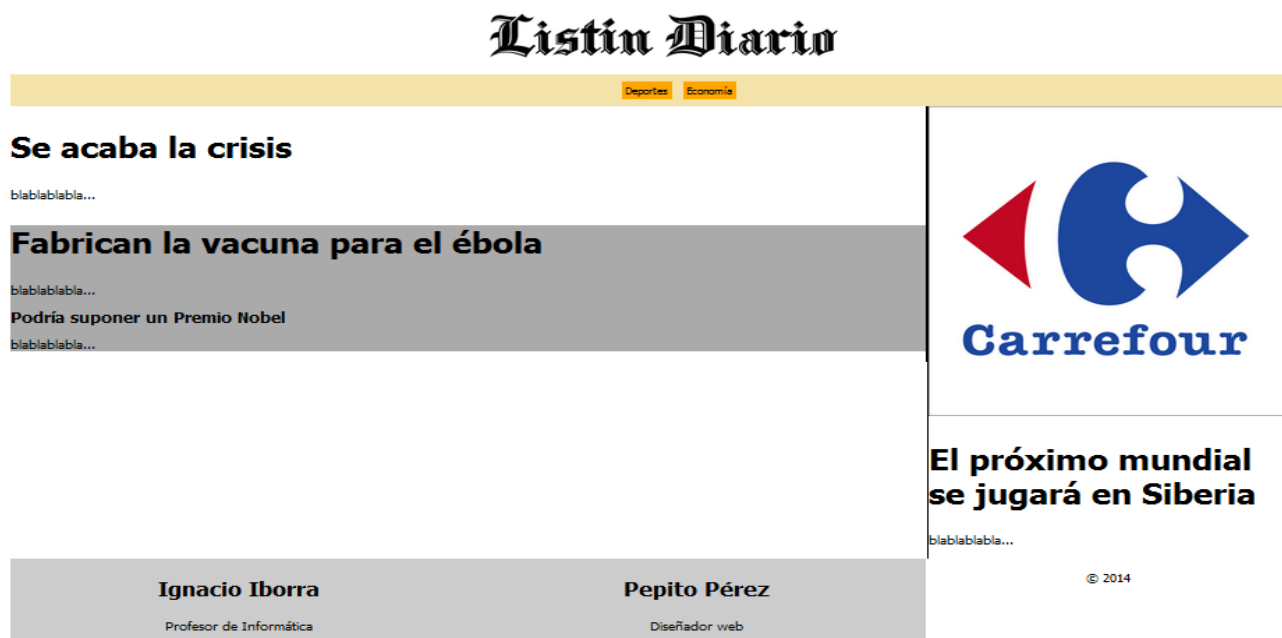
blablablabla...

Fabrican la vacuna para el ébola

blablablabla...

Podría suoner un Premio Nobel

Que es podria veure en el navegador (utilitzant CSS apropiadament) més o menys així:



Observa diversos aspectes:

- Per a facilitar la col·locació de les "caixes", tanquem cada bloc en un *div*. Hi ha un *div* "principal" que tancaria tot el contingut de notícies (entre el *header* i el *footer*, i dins hi ha dos *div*: un per a la columna 1 i un altre per a la columna 2. En cadascun d'aquests *div*, definim les seccions, articles o *asides* que considerem, i dins de cadascun el seu contingut i apartats.
- Podem tindre niades unes certes etiquetes en unes altres. Per exemple, un *section* dins d'un altre, o dins d'un *article*, o un *article* dins d'un *section*, o un *nav* en un *header*, o dins d'un *section*... hi ha diverses combinacions possibles.

Exercici 1

Crea una carpeta anomenada *periodic*. Dins, crea un document anomenat **portada.html** que continga més o menys el contingut de l'exemple anterior (pots canviar el contingut de les notícies, o el bàner de publicitat, logo del periòdic... però mantindre la mateixa estructura). Després, crea en la mateixa carpeta un document anomenat **estils.css** que, mitjançant estils, intente deixar-ho el més semblant possible a l'exemple. Enllaça el document CSS amb l'HTML mitjançant una etiqueta *link* en el *head* d'aquest últim.

Noves opcions per a formularis

Veurem a continuació algunes noves opcions incorporades en HTML5 que podem utilitzar en els nostres formularis, per a facilitar-nos tant la seua creació com l'enviament de dades correctes per part de l'usuari.

L'atribut *required*

Sovint ens haurem trobat amb formularis on un o diversos camps són obligatoris, és a dir, hem d'emplenar-los amb dades, de manera que si no ho fem, el formulari "es queixa" i ens marca (normalment en roig) el que no hem emplenat. Aquestes coses, fins a HTML4,

s'havien de fer amb Javascript si no volíem que fora el servidor el que rebera dades incorrectes i haguera de notificar d'això de nou al client, perdent temps i recursos.

L'atribut **required** s'aplica des d'HTML 5 en els camps d'un formulari que siga obligatori emplenar, de manera que el formulari directament no s'envia si estan buits. Per exemple, en el següent formulari és obligatori emplenar el camp *login*:

```
...  
<form action="login.php" method="post">  
  <label for="login">Login:</label>  
  <input type="text" name="login" required aria-required="true" />  
  <br />  
  <label for="login">Password:</label>  
  <input type="text" name="password" />  
  <br />  
  <input type="submit" value="Enviar" />  
</form>
```

Si no ho emplenem, el navegador ens posicionarà sobre aqueix quadre de text amb alguna mena de missatge (depenent del navegador) indicant que hem d'emplenar-lo. L'atribut *aria-required* és opcional, però fa que el nostre formulari siga més accessible per als lectors de pantalla.

L'atribut *pattern*

Aquest atribut permet especificar un patró d'entrada, de manera que perquè la dada que introdueix l'usuari es considere vàlid, ha d'ajustar-se a aqueix patró. Per exemple, per a un camp que demane un número de telèfon podem esperar que l'usuari escriga entre 9 i 11 dígits:

```
<input type="text" name="telefono" pattern="[0-9]{9,11}" />
```

És habitual també indicar en aqueix camp del formulari quin tipus d'informació s'espera. La sintaxi que s'utilitza per a definir diferents tipus de patrons és la mateixa que s'usa en Javascript, per exemple, però no la veurem amb detall en aquest tema. Simplement veurem alguns casos útils:

- Els claudàtors especifiquen un rang de valors. Per exemple, `[0-9]` indica qualsevol dígit, i `[a-zA-Z]` qualsevol lletra.
- Les claus indiquen el nombre d'ocurrències, ben exacte (un sol número entre claus) o un número mínim i màxim d'ocurrències, separats per comes.
- El símbol `+` indica una o més vegades. Per exemple, `u` o més dígits `([0-9]+)`. Per part seua, el símbol `*` indica zero o més vegades.
- El símbol `\` s'utilitza per a representar alguns símbols especials, com per exemple el punt (`\.`)

Els atributs *disabled* i *readonly*

Aquests atributs ja existien en versions anteriors, però ara han ampliat el rang. No sols podem posar-los en objectes *input*, sinó en quasi qualsevol control. Si ho posem per exemple en un *fieldset*, fem que tots els controls que conté queden desactivats o siguen de només lectura, respectivament.

Nous tipus d'input.

L'atribut *type* del control *input* ens servia per a definir diferents tipus de controls: quadres de text (*text*), quadres de password (*password*), botons (*button*), botó de reinicialitza (*reset*), caselles de selecció (*casella de selecció*), etc. Amb HTML5, aquesta llista de possibles opcions s'amplia, i podem tindre també quadres específics per a definir un e-mail (*type="email"*), o una data (*type="date"*), o una hora (*type="time"*), o tot junt (*type="datetime"*), o un número (*type="number"*), o un color (*type="color"*), entre altres opcions. Tots aquests nous tipus tenen validació nativa en HTML5 (és a dir, que en indicar aqueix tipus específic, el propi HTML5 s'encarrega de verificar si el que ha escrit l'usuari és un e-mail, data, hora, número o color vàlid, respectivament).

A més, en alguns d'aquests tipus, com en els tipus *file* i *email*, podem afegir l'atribut **multiple** (que fins ara s'usava en llistes fixes, per a indicar que es podia seleccionar més d'un valor). En aquest cas, permet afegir en el formulari més d'un arxiu adjunt, o més d'un e-mail, respectivament.

```
<input type="email" name="email" multiple />
```

Exercici 2

Realitzarem algunes modificacions al formulari *formDades.html* del tema anterior (*Exercici 20*). Crea una còpia d'aquest arxiu i canvia-la de nom a *formDadesHTML5.html*. Afig els següents canvis:

- Que el camp "email" siga de tipus e-mail.
- Fes que tant el nom, com l'e-mail siguen requerits.
- El DNI ha de constar de 8 dígits.

Àudio i vídeo

Aquesta potser és una de les novetats més potents que han aparegut en HTML5: la possibilitat d'afegir contingut multimèdia a una pàgina web i poder-ho veure o sentir sense necessitat de llibreries externes.

Vídeo

Pel que fa al vídeo, tenim l'etiqueta **vídeo**, amb un atribut **src** (similar al de les imatges) per a indicar l'arxiu de vídeo que volem mostrar. Alguns dels formats de vídeo més suportats són MPEG-4, OGG i WebM. Tenim els atributs de *width* i **height** per a donar una grandària al marc de reproducció. També tenim un atribut **controls** per a proporcionar els controls per a manejar el vídeo (reproduir, pausar, etc.), i altres atributs que no veurem ací. Vegem un exemple:

```
<vídeo src="vídeos/pelicula.mp4" width="500" height="300" controls></vídeo>
```

Cal remarcar que el vídeo se centrarà automàticament vertical i horitzontalment, en el cas que les dimensions no coincidisquen amb el *width* o el *height* indicats. Per exemple, si el vídeo anterior tinguera 250 píxels d'alt, se centraria verticalment deixant 25 píxels en negre per dalt i per baix.

Altra alternativa a això és utilitzar una etiqueta **source** dins de vídeo, per a indicar l'arxiu a reproduir, i el tipus d'arxiu que és, perquè el navegador pugui detectar si ho pot reproduir

o no, abans de carregar-ho. En aquest cas, no hi ha atribut `src`, i podem donar diversos formats del mateix vídeo perquè el navegador tinga on triar.

```
<video width="500" height="300" controls>
  <source src="videos/pelicula.mp4" type="video/mp4" />
  <source src="videos/peligula.ogv" type="video/ogg" />
</video>
```

Àudio

De manera similar a l'anterior, disposem d'una etiqueta **àudio** per a reproduir arxius d'àudio en una pàgina web. També compta amb un atribut **src** per a indicar l'arxiu d'àudio a reproduir, i un atribut **controls** per als controls de reproducció, entre altres atributs. Entre els formats més suportats estan MP3 i OGG/Vorbis.

```
<audio src="musica/cancion.mp3" controls></audio>
```

També tenim l'alternativa d'usar l'etiqueta *source*, com en el vídeo:

```
<audio controls>
  <source="musica/cancion.mp3" type="audio/mpeg" />
  <source="musica/cancion.ogg" type="audio/ogg" />
</audio>
```

Altres funcionalitats d'HTML5

Una vegada vista la nova estructura d'un document web amb HTML5, i alguns dels canvis més importants, veurem algunes altres noves etiquetes i funcionalitats que podem usar en les nostres pàgines web.

Els elements *figure* i *figcaption*

Aquests elements s'utilitzen per a fer anotacions en unes certes figures del document (imatges, diagrames, fragments de codi...) de manera que, si queden separats del lloc al qual pertanyen, l'anotació permeti associar-los al seu contingut. Imaginem per exemple una notícia, amb una imatge relacionada amb ella. Si, per motius de disseny, la imatge queda fora de la notícia, amb l'anotació podrem saber que pertany a aqueixa notícia. A més, es pot enllaçar el contingut de la notícia amb la figura en si.

```
<h1>Elvis no ha mort</h1>

<p>Fonts fiables informen d'haver vist al Rei a les platges de Miami, envoltat
d'alguns dels seus afins, com pot veure's en la <a href="#fig1">imagen</a>.</p>
...
<figure id="fig1">
  <figcaption>Elvis Presley en les platges de Miami</figcaption>
  
</figure>
```

Podem veure com l'etiqueta **figure** serveix per a englobar en un conjunt l'element que es vol anotar (en aquest cas, una imatge) i l'anotació que es fa. A tot això se li posa un nom (en l'exemple, *fig1*), amb el qual es pot enllaçar des d'un altre contingut.

L'element *details*

Aquest element, juntament amb el seu subetiqueta **summary**, s'utilitza per a crear zones expandibles, és a dir, zones d'informació que apareixen inicialment "plegades", i que fent clic sobre elles amplien la informació que contenen. Per exemple:

```
<details>
  <summary>Enllaços d'interés</summary>
  <ul>
    <li><a href="http://www.ua.es">Universitat d'Alacant</a></li>
    <li><a href="http://www.upv.es">Univ. Polit. de València</a></li>
  </ul>
</details>
```

En el navegador només es mostraria:

► Enllaços d'interés

i en punxar sobre això, apareixeria la llista d'enllaços davall.

Opcions extra per a llistes ordenades

Quan creguem una llista ordenada (etiqueta *ol*), podem utilitzar dos atributs extra: l'atribut **reversed** (per a indicar que la numeració vaja de major a menor) i l'atribut **start** per a indicar des de quin número començar (si no s'indica, es comença o acaba en 1, depenent de si és ordre ascendent o descendent).

Un exemple:

```
<ol start="7" reversed>
  <li>U</li>
  <li>Dos</li>
  <li>Tres</li>
</ol>
```

que es veuria així:

7. U
6. Dos
5. Tres

Altres etiquetes

Existeixen altres etiquetes que poden resultar molt útils per a alguns casos, com les etiquetes *progress* i *ficar* per a definir indicadors de progrés per a alguna acció que hàgem realitzat en la pàgina, o l'etiqueta *estafe* per a anotar una data que escrivim a mà en algun document HTML en un format de data que un cercador puga interpretar millor (per si algú busca alguna data donada en un document). Aquestes, i moltes altres etiquetes, no les veurem en aquest tema per a simplificar-lo una mica, i per estar una mica allunyades del propòsit d'aquest.

Algunes característiques de CSS3

Una vegada vistes algunes de les noves característiques que incorpora HTML5, veurem en aquest apartat algunes de les novetats més importants que inclou CSS3 a l'hora de dotar a les nostres aplicacions web d'estils amb cada vegada més possibilitats.

Novetats en l'ús de colors

Com ja hem vist anteriorment, fins a CSS2 teníem bàsicament tres maneres d'especificar el color d'un element en una pàgina: amb un nom de color simple (*xarxa*, *yellow*...), amb el seu codi de colors en RGB decimal (tres xifres de 0 a 255 per a cada component roig, verd i blau, per exemple: *rgb(12, 116, 52)*), i amb el seu codi de colors RGB en hexadecimal (seguint la mateixa filosofia de l'anterior, però amb format hexadecimal, per exemple: *#FF2E34*).

CSS3 incorpora novetats interessants en aquest conjunt de formats de color, com és la inclusió del nivell de transparència o opacitat d'un element, o l'ús de degradats.

Definir la transparència

Amb CSS3 podem definir el grau de transparència o opacitat que tindrà el color d'un element (tant color de text com color de fons), afegint un quart element als tres colors RGB, passant a ser un format **RGBA**. Aquest quart element és un nombre decimal que va des del 0 (totalment transparent) fins a l'1 (totalment opac). Si decidim usar la transparència, llavors haurem d'especificar el color d'aquesta manera:

```
h1{ background-color: rgba(12, 116, 52, 0.5); }
```

on els tres primers números expressen la quantitat de roig, verd i blau, com abans, i el quart número és la transparència.

Una alternativa diferent és utilitzar la propietat **opacity** en el CSS, amb valors també entre 0 i 1:

```
p { background-color: rgb(12, 116, 52); opacity: 0.5; }
```

La diferència amb l'anterior és que, usant *opacity*, l'opacitat afecta a l'element i a tots els subelements que continga, incloent el text (que podria ser llavors parcialment transparent sense nosaltres voler-ho).

Ús de degradats lineals



Un degradat lineal és un efecte pel qual passem gradualment d'un color a un altre, seguint un eix horitzontal, vertical o en qualsevol altra direcció. Amb CSS 3, podem aconseguir aquest efecte amb la funció **linear-gradient**. Entre parèntesi (sense deixar espai abans del parèntesi), indicarem certs

valores:

- La direcció del degradat, que podem indicar-la de tres formes diferents:
 - Indicant cap a quin costat anar: *to top*, *to bottom*, *to left* o *to right*
 - Indicant des de quin extrem partir: *top left*, *top right*, *bottom left* o *bottom right*

- Indicant un angle en graus (*deg*), on *0deg* apunta cap amunt i *90deg* cap a la dreta.
- Dos o més colors (separats per comes), que seran els colors cap als quals s'anirà canviant gradualment (d'un al següent). Al costat de cada color, es pot indicar un percentatge, que expressa en quin punt del recorregut es tindrà aqueix color

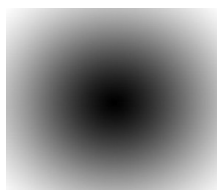
Per exemple, el següent degradat va cap a la dreta, part del blanc, passa per un roig a mig camí, i acaba en un negre.

```
div { background-image: linear-gradient(90deg, #FFFFFF, #FF0000 50%, #000000); }
```



Existeixen moltes altres variants d'ús de degradats lineals, i algunes incompatibilitats amb alguns formats depenent del navegador, però aquest exemple ens pot servir de mostra del que es pot fer.

Ús de degradats radials



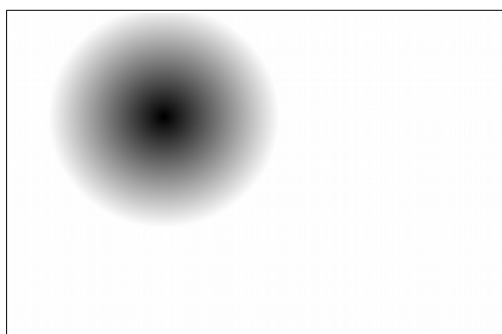
Un degradat radial és un altre tipus de degradat on es passa d'un color en un punt concret d'un determinat element (per defecte, el centre del mateix) i es va canviant gradualment cap a un altre color, en totes direccions a partir d'aqueix punt. Aquest degradat s'obté amb la funció **radial-gradient**. Els paràmetres que admet són:

- La posició del punt d'inici del degradat. Si no s'indica res, es pren el centre de l'element (el que podem indicar amb *center center*), però podem moure-ho a una altra posició, indicant les seues coordenades X e Y separades per un espai, o bé un costat i alineació (*left top*, *right center*, *center bottom*...)
- La forma del centre de difusió. Podem utilitzar les paraules *circle* o *ellipse*, seguida d'una altra que indique la seua grandària: *closest-side* (fins al costat més pròxim), *closest-corner* (fins a la cantonada més pròxima), *farthest-side*, *farthest-corner* (fins al costat/cantó més llunyà)
- Una llista d'almenys dos colors: un que serà el del punt de partida i l'altre al qual es va canviant. Igual que ocorria amb el degradat lineal, podem especificar més de 2 colors, indicant per als intermedis el percentatge de trajecte en el qual s'ha de tindre aqueix color.

El següent exemple crea un degradat radial que comença a 50 px a la dreta i 30 px per davall de l'extrem superior. Té forma circular fins al costat més pròxim, i passa del negre al blanc.

```
div { background-image: radial-gradient(50px 30px, circle closest-side, black, white); }
```

i obtindríem alguna cosa com això:



Models de color HSL

A més del format RGB, amb CSS3 també podem indicar colors en format HSL, on també s'especifiquen tres valors per a un color:

- El **to** (*Hue*), un valor en graus entre 0 i 359, sent per exemple el 0 el roig, 60 groc, etc.
- La **saturació**, en percentatge, indicant com de complet és el to anterior. Un valor de 100% indica que és el to en tota la seua plenitud, i un 0% deixarà una ombra grisa, ignorant el to
- La **lluminositat** també en percentatge, sent 100% completament blanc i 0% completament negre. El normal és estar entorn del 50%.

```
h1 { background-color: hsl(120, 80%, 50%); }
```

- Addicionalment, podem indicar un quart paràmetre, que és la transparència, i funciona igual que en el cas anterior. En aquest cas, estem amb el model HSLA

```
h1 { background-color: hsla (120, 80%, 50%, 0.5); }
```

Cantonades arredonides

Un dels principals elements de disseny que va sorgir amb les aplicacions mòbils va ser l'ús de caixes o icones amb les vores arredonides. Fins a la seua generalització en CSS, només era possible aconseguir aquest efecte amb alguna propietat CSS que només funcionava en uns certs navegadors, com la propietat *-moz-border-radius* en Firefox, o realitzant vertaderes obres d'art amb píxels de color i CSS per a aconseguir "dibuixar" a mà les cantonades.

Afortunadament, amb CSS3 tot això s'ha normalitzat, utilitzant la propietat **border-radius**, i indicant el radi de la vora (normalment en píxels):

```
div { border-radius: 10px; }
```

Alternativament, també podem usar les propietats **border-top-left-radius**, **border-top-right-radius**, **border-bottom-left-radius** i **border-bottom-right-radius** per a indicar radis separats per a cada cantonada, o bé indicar una llista de quatre radis diferents en *border-radius*, separats per espais, que s'apliquen respectivament a les cantonades superior esquerra i dreta, i inferior dreta i esquerra:

```
div { border-top-left-radius: 10px; border-top-right-radius: 15px;
      border-bottom-left-radius: 25px; border-bottom-right-radius: 20px; }
```

o bé

```
div { border-radius: 10px 15px 20px 25px; }
```

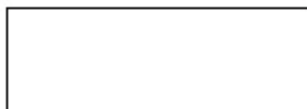
Existeixen altres combinacions possibles, però són menys habituals i no les veurem ací.

Ús d'ombres

Una altra característica de CSS3 és la possibilitat d'usar ombres en quasi qualsevol component. La forma ràpida de fer-ho és utilitzar la propietat **box-shadow**, que permet especificar el color de l'ombra, gruix, desenfocament, etc. El seu ús bàsic és com en el següent exemple:

```
p.ombra { box-shadow: 5px 10px 0px 0px rgba (100, 120, 145, 1); }
```

- El primer valor és el desplaçament **horitzontal** de l'ombra. Un valor positiu deixarà una ombra per la dreta, i un negatiu una ombra per l'esquerra, de tants píxels com s'indique.
- El segon valor és el desplaçament **vertical** de l'ombra. Un valor positiu deixarà una ombra per davall, i un negatiu una ombra per damunt, de tants píxels com s'indique.
- El tercer valor és el **desenfocament**. Només admet valors positius, i com més gran siga més desenfocament hi haurà. Un valor de 0 indica que no hi ha desenfocament.
- El quart valor és la **distància de propagació** de l'ombra. Un valor positiu fa que l'ombra es propague en totes direccions, sent llavors més gran que l'element al qual s'aplica, i un negatiu que es contraga (sent més xicoteta que l'element al qual s'aplica). Un valor de 0 indica que no hi ha propagació. Per exemple, aquesta caixa té una distància de propagació negativa, per la qual cosa l'ombra (que només està aplicada per davall) és més xicoteta que la longitud de la caixa.

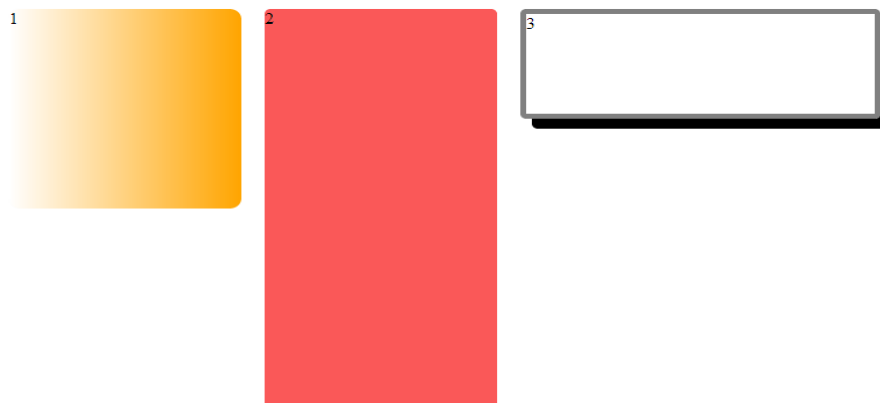


- El cinqué valor és el **color** de l'ombra. Podem usar valors amb o sense transparència.

Anàlogament, existeix també una propietat anomenada **text-shadow** per a donar ombra al text d'un element. Té els mateixos valors que els explicats abans per a box-shadow, excepte la distància de propagació, que no existeix.

Exercici 3

Crea un document anomenat **estilsCSS3.html**. En ell, crea una etiqueta *style* i deixa embeguts dins els estils per a fer alguna cosa com això:



- La primera caixa (1) ha de tindre una amplària i altura de 200px. Vores arrodonides de 10px de radi i degradat de blanc a taronja, d'esquerra a dreta.
- La segona caixa (2) ha de tindre una amplària de 200px i altura de 400px respectivament. Ha de tindre vores arrodonides amb 5px de radi, i color roig clar de fons.
- La tercera caixa (3) ha de tindre amplària de 300px, i altura de 100px. Ha de tindre una vora arrodonida de color gris de 5px de gruix, i ombra de 10px per la dreta i per baix, de color negre, sense propagació.
- Totes les caixes han de tindre 10px de marge a cada costat, i posicionament flotant a l'esquerra.

Prova de canviar la grandària de la finestra per a veure com es recol·loquen .

Transformacions i transicions

Vejam alguns dels efectes de transformació de la forma d'alguns elements, o de moviment d'elements per la pàgina, que incorpora CSS3.

Transformacions

La propietat **transform** ens permet realitzar una transformació sobre un element. Aquesta transformació pot ser de diversos tipus:

- Un simple **desplaçament**, amb la funció **translate**. Aquest desplaçament només pot fer-se en relació a la posició actual de l'element. Podem utilitzar dos arguments: el desplaçament en X (des de l'extrem esquerre de l'element) i en Y (des de l'extrem superior). Tots dos admeten valors negatius (per a moure's cap a l'esquerra i cap amunt, respectivament). Per exemple, la següent regla desplaça una caixa 100 píxels a la dreta i 20 píxels cap avall:

```
div { transform: translate (100px, 20px); }
```

Si només volem moure en una direcció, tenim les funcions **translatex** i **translatey**, a les quals només hem d'indicar-los el moviment horitzontal i vertical, respectivament.

- Un **escalat** (canvi de grandària), amb la funció **scale**. Li indiquem amb dos paràmetres l'escalat en X e Y (sent 1 la grandària real, 0.5 la meitat de la grandària real, 2 el doble de la grandària real, etc.). Si només indiquem un valor, s'emprarà tant per a l'escalat en X com per a l'escalat en Y. Per exemple, aquest estil escala les imatges a la meitat de la seua grandària en pantalla:

```
img { transform: scale(0.5, 0.5); }
```

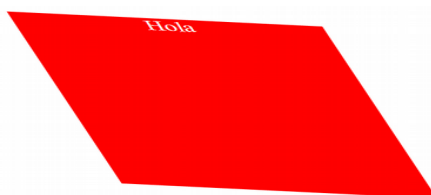
Igual que en el cas anterior, també tenim les funcions **scalex** i **scaley** per a escalar només en una dimensió.

- Una **rotació** al voltant del punt d'origen de l'element (per defecte, el seu centre). Indiquem l'angle de rotació en graus (graus positius giren cap a la dreta, en el sentit horari)

```
div { transform: rotate (45deg); }
```

- Una **inclinació** al llarg dels eixos X e Y, amb la funció **skew**. La inclinació respecte a cada eix s'indica en graus. Amb això s'aconsegueix un element inclinat, com amb perspectiva.

```
div { transform: skew (20deg, 5deg); }
```



Si volem aplicar més d'una transformació, es posen una després d'una altra separades per espais:

```
div { transform: translateX(50px) scale(1.5, 0.25); }
```

Transicions

Una transició és un efecte pel qual un element passa d'un estat o posició inicial a un altre, però no de manera immediata (com succeeix en les transformacions) sinó gradualment al llarg d'uns segons. Per exemple, podem fer canvis graduals de color, o de posició d'un element. Per a això, utilitzarem la propietat **transition**, a la qual li donarem els següents valors (separats cadascun per espais):

- La propietat CSS de l'element sobre la qual s'ha d'actuar en la transició (per exemple: *color*, *width*, *height*, *background-color*...).
- La duració de la transició, en segons (s) o mil·lisegons (ms).
- La funció de transició (opcional), que pot valdre *linear* (tota la transició té el mateix ritme), *ease* (comença lenta, després va ràpida i acaba lenta), *ease-in* (començament lent), *ease-out* (final lent), etc.
- On retarde (opcional) a l'inici de la transició, és a dir, un temps d'espera en segons o mil·lisegons, abans de començar.

Si volem aplicar una transició sobre més d'una propietat, es posa una coma i es repeteixen aquests valors per a la següent propietat CSS. Per exemple, el següent CSS canvia la posició (X) i el color d'una caixa quan passem el ratolí per damunt, en una animació de 2 segons per a la posició i 1 segon per al color, amb final lent.

```
#caixa1 {
  position: relative;
  left: 0px;
  background-color: red;
```



```

    transition: background-color 1s ease-out, left 2s ease-out;
}
#caixa1:hover {
    background-color: blue;
    left: 100px;
}

```

Notar com necessitem definir dues regles: una amb les condicions inicials de l'element, i una altra amb les condicions que canvien en passar el ratolí per damunt (:hover). En les condicions inicials, indiquem le(s) transició(ns) que volem, que seran les que canvien fins a les indicades en la regla :hover, i també indiquem els valors inicials que tindran les propietats que volem canviar (posicions inicials, colors inicials...). En la regla :hover simplement indiquem els valors finals que vulguem que tinguin les propietats a canviar (en el cas anterior, l'increment en X i el color de fons final). En el cas de voler canviar la posició (*left* o *top*), és important definir el tipus de posicionament de l'objecte (relatiu, en el cas anterior).

També podríem fer que les transicions comencen en passar el ratolí per damunt d'un altre element que continga a l'actual, o que estiga connectat amb ell d'alguna manera. Per exemple, si volem aplicar-les en entrar amb el ratolí en qualsevol lloc de la pàgina, en lloc d'usar el selector #caja1:hover anterior, usaríem este:

```

html:hover #caixa1 {
    background-color: blue;
    left: 100px;
}

```

Animacions

A més de tot això, CSS3 també permet realitzar animacions sobre els elements. Les animacions són més complexes que les transicions perquè en aquestes només hi ha un estat inicial i un final, mentre que en una animació hi ha molts estats intermedis. No obstant això, no tractarem animacions en aquest tema degut a la seua dificultat i al fet que s'allunyen del propòsit essencial.

Exercici 4

Afig una transició a la caixa 2 de l'exercici anterior perquè, en passar el ratolí per damunt, canvie el seu color del roig clar al blau clar, en 2 segons. Afig una altra transició a la caixa 3 perquè, en passar el ratolí per damunt, es desplaci 20 píxels cap avall, en 1 segon. Fes que la caixa 1 baixi 200 píxels quan posem el ratolí dins de la pàgina.

Incorporar noves fonts

Ja hem vist que amb la propietat *font-family* podem indicar el tipus de lletra que volem que tinga la web, i que fins i tot podem indicar diverses (separades per comes), per si el navegador no detecta que el sistema tinga instal·lada alguna d'elles, poder triar la següent. No obstant això, aquest mètode és dependent dels tipus de lletra que tinga instal·lats el client en el seu sistema, i és possible que la web es veja amb tipus de lletra diferents depenent del sistema.

Per a evitar este problema podem utilitzar tipus de lletra "estàndard" que normalment són presents en tots els sistemes operatius (com per exemple *Arial*, *Courier New*, *Verdana*...)

o bé incloure en el CSS i en la nostra web el tipus de lletra concret que vulguem utilitzar, perquè el navegador el localitzi en carregar la pàgina i pugui mostrar-la independentment del sistema operatiu sobre el qual s'estiga executant.

Per a aquesta última opció, tenim la directiva **@font-face**, que ens permet importar els arxius d'una determinada font en el nostre CSS, per a poder utilitzar aqueixa font en les nostres pàgines. Entre claus hem d'indicar, almenys, dues propietats:

- El **nom** que li donarem a aqueixa font. Pot ser el que nosaltres vulguem, i s'indica en la propietat **font-family**.
- La **ruta a l'arxiu** de la font, que prèviament ens haurem hagut de descarregar i col·locar en alguna carpeta de la nostra web. Aquesta ruta s'indica en la propietat **src**. Per exemple, si ens hem descarregat la font *miFuente.otf*, i la tenim en la carpeta *fuentes*, podríem tindre en conjunt una mica com això:

```
@font-face
{
    font-family: "miFuente";
    src:url("fuentes/miFuente.otf") format ("otf");
}
```

En el cas dels arxius de font, existeixen diferents **formats d'arxiu**:

- Arxius **EOT**, propietari per a Internet Explorer
- Arxius **OTF** (*OpenType Format*) i **WOFF** (*Web Open Font Format*), reconeguts per quasi tots els navegadors
- Arxius **TTF** (*TrueType Font*), que sovint no són gratuïts
- Format **SVG**, que en realitat són imatges vectorials, però fins fa poc era l'únic sistema suportat per dispositius iOS (iPhone, iPod Touch...)

Hauríem d'obtenir la font en algun format que siga compatible per tots els navegadors (normalment, OTF o WOFF), o bé descarregar un "pack" que continga la font en cadascun d'aquests formats, i afegir-los tots en la propietat **src**, separats per comes. Quedaria una cosa així:

```
@font-face
{
    font-family: "miFuente";
    src:url("fuentes/miFuente.eot") format ("eot"),
    src:url("fuentes/miFuente.eot?ieantiguo") format ("eot"),
    src:url("fuentes/miFuente.woff") format ("woff"),
    src:url("fuentes/miFuente.otf") format ("otf"),
    src:url("fuentes/miFuente.ttf") format ("ttf"),
    src:url("fuentes/miFuente.svg#webfontFHzvtkso") format ("svg");
}
```

Observa que el format EOT està duplicat. Açò és així perquè la primera línia serveix per a versions d'Internet Explorer 9 o posteriors, i la segona per a versions anteriors. L'afegitó de *"?ieantiguo"* és simplement un paràmetre que se li posa de per riure, perquè Internet Explorer ometa la resta de formats que venen després.

Normalment, serà necessari descarregar les versions en negreta (*bold*) i en cursiva (*italic*) de cada font, que venen en arxius separats de la font normal. Després, haurem d'afegir l'arxiu corresponent amb un nou **@font-face**, amb el mateix nom que l'original, i amb una

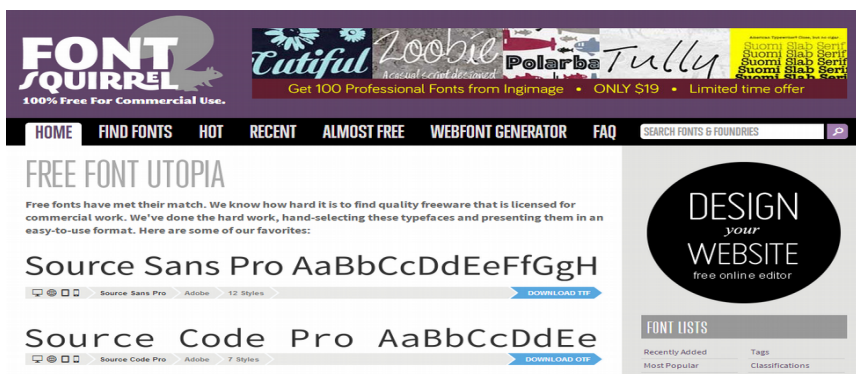
propietat més, per a indicar en cada cas que és negreta o cursiva. Ací veiem un exemple de cadascuna:

```
@font-face
{
  font-family: "miFuente";
  src:url("fuentes/miFuente.otf") format ("otf");
}

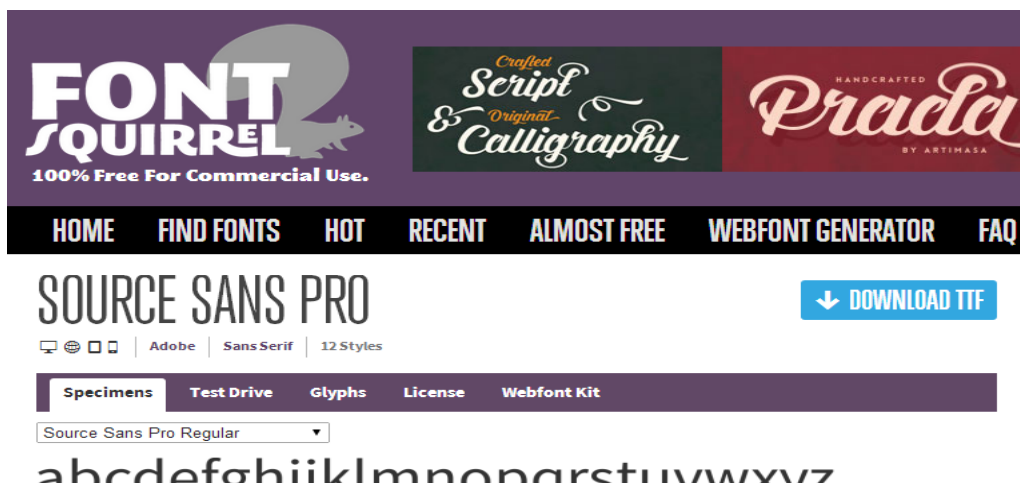
@font-face
{
  font-family: "miFuente";
  src:url("fuentes/miFuente_Bold.otf") format ("otf");
  font-weight: bold;
}
.
@font-face
{
  font-family: "miFuente";
  src:url("fuentes/miFuente_Italic.otf") format ("otf");
  font-style: italic;
}
```

Obtindre fonts

Podem obtenir fonts diferents a les habituals en diverses pàgines d'Internet. Per posar un exemple, parlarem ací de FontSquirrel (www.fontsquirrel.com):



En aquesta web, podem buscar diferents fonts per categories (afegides recentment, les més populars, etc.). Al costat de cada font, en blau, apareixerà un enllaç per a descarregar-nos l'arxiu (en format OTF o TTF, segons la font). A més, punxant sobre la font, accedirem a una altra pàgina amb més opcions:



En la llista de l'esquerra, podem carregar la font en les seues versions normal (*regular*) negreta i/o cursiva, i en el botó blau de la dreta, descarregar l'arxiu TTF/OTF d'aqueixa versió de la font. A més, en el menú violeta, per a algunes fonts, tindrem l'opció *WebFont Kit* que ens descarregarà un paquet amb la font en diferents formats d'arxiu.

Una vegada descarregats els arxius per a la(s) font(s) i format(s) que vulguem, haurem de copiar-los en l'aplicació web (en la carpeta arrel, o en alguna subcarpeta). Normalment amb el *WebFont Kit* ve ja un CSS de prova per a col·locar la font en les nostres pàgines, que podem aprofitar.

Aplicar la font

Una vegada tinguem la font col·locada en la nostra aplicació web, i els *@font-face* llocs en el nostre CSS. Si volem aplicar aqueixa font als paràgrafs (per exemple), bastarà amb usar la propietat *font-family* que ja coneixem per a text, i posar-li el mateix nom que li hàgem posat en el *font-family* del *@font-face*.

```
@font-face
{
  font-family: "miFuente";
  ...
}
...
p { font-family: "miFuente"; }
```

Exercici 5

Crea una pàgina anomenada **miFuente.html** , i crea en ella elements de diversos tipus (h1, paràgrafs, llistes, la qual cosa vulgues). Cerca alguna font que t'agrade, descàrrega l'arxiu (o arxius) que necessites i aplica-la a aquesta pàgina.

Una altra manera d'incrustar fonts en la pàgina és important-les. Per a això, Google ens ofereix la possibilitat d'importar directament la font a la fulla d'estils.

Per a això accedim a la web de Google <https://fonts.google.com> i observem les diferents famílies de fonts que conté. Seleccionem (+) la font que desitgem i obtenim el codi import necessari per al nostre CSS (p.e. `@import 'https://fonts.googleapis.com/css?family=open+Sans';`) i que col·locarem al principi del nostre fitxer d'estil.