

Tema 6

Fulles d'estil CSS (II)

Objectius

- Aplicar les carauterístiques de les fulles d'estil en els documents HTML.
- Conèixer les tècniques bàsiques del disseny visual de pàgines i llocs web.
- Reconèixer els avantatges que aporta la utilització del model de caixes.
- Conèixer les carauterístiques avançades de les fulles d'estil i aplicar-les al disseny visual.
- Reconèixer els avantatges que aporta la utilització d'elements de disseny visual accessible i adaptatiu (*responsive*) a les carauterístiques de cada dispositiu.

Introducció

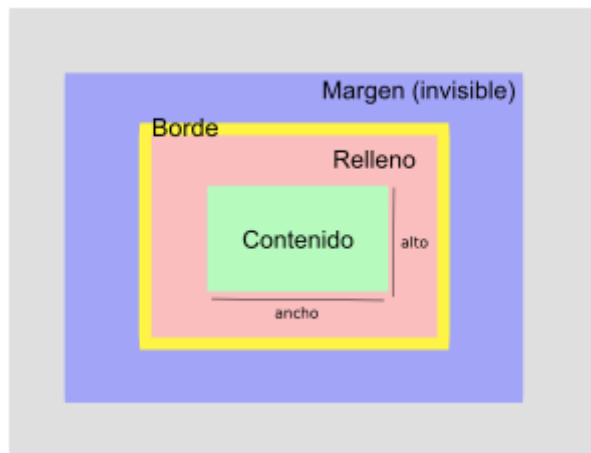
En el tema anterior es van estudiar les carauterístiques més bàsiques de les fulles d'estil en cascada. L'ús de les fulles d'estil permet millorar l'aparença de la presentació dels diferents elements que componen un document HTML. En aquest tema s'aborda l'estudi d'aspectes de disseny visual utilitzant per a això les carauterístiques avançades de les fulles d'estil en cascada. El disseny visual de les pàgines i llocs web té la seua importància en un entorn d'informació multimèdia, com és el servei Web. La utilització de les tècniques de disseny visual, que s'estudien en aquest tema, permet al desenvolupador de pàgines i lloc web obtenir resultats que utilitzen la potencialitat de les tècniques de desenvolupament Web. L'estudi de les fulles d'estil en cascada s'ha dividit en diversos temes a causa de la seua amplitud, facilitant-se d'aquesta manera el seu aprenentatge d'una forma més detallada i ordenada. Aquests temes s'han planificat de manera consecutiva dins de la periodificació de continguts del mòdul professional.

Índex

6.1 MODEL DE CAIXES	2
6.2 POSICIONAMENT	8
6.3 CONTROL DE LA VISUALITZACIÓ	16
6.4 ESTRUCTURA I LAYOUT	28
6.5 BONES PRÀCTIQUES	38
Glossari de termes	40

6.1 MODEL DE CAIXES

El disseny visual de pàgines Web es basa en la utilització de les fulles d'estil i, més concretament, en el denominat model de caixes. El model de caixes és un model de format que es basa en el fet que cada element HTML produeix una caixa o zona rectangular al seu voltant. Així, qualsevol element HTML (taula, fila, paràgraf, divisió, etc.) és en realitat una caixa amb la següent estructura:



Així, cada caixa té un **contingut** que pot ser un text en el cas d'un paràgraf, una imatge, el contingut d'una cel·la, altres capes, etc. Al voltant del contingut, hi ha un espai de farciment (*padding*), que és un espai entre el contingut i la vora. La **vora** (*border*), que pot estar visible o no, correspon amb una línia que tanca completament el contingut i el seu farciment. I, finalment, el **marge** (*margin*), que és l'espai lliure existent entre la caixa i allò que li envolta, és a dir, les altres caixes adjacents. L'aparença de la presentació visual de cadascuna de les parts que constitueixen una caixa pot establir-se mitjançant propietats d'estil:

- Farciment

Propietat	Descripció	Valors
Padding-top padding-right padding-bottom padding-left	Ample del farciment superior, dret, inferior i esquerre	[<longitud> <percentatge> auto]
padding	Grandàries per a farciments.	[<longitud> <percentatge> auto] {1,4}

- Marges

Propietat	Descripció	Valors
Margin-top margin-right margin-bottom margin-left	Grandària del marge superior, dret, inferior i esquerre	[<longitud> <percentatge> auto]
margin	Ample per a marges.	[<longitud> <percentatge> auto]{1,4}

- Vores

Propietat	Descripció	Valors
Border-top-width border-right-width border-bottom-width border-left-width	Amplària de la vora superior, dret, inferior o esquerre	[thin medium thick <longitud>]
border-width	Amplès de diverses vores individuals	[thin medium thick <longitud>] {1,4}
Border-top-color border-right-color border-bottom-color border-left-color	Color de la vora superior, dret, inferior o esquerre	[<color> transparent]
border-color	Colors de diverses vores individuals	[<color> transparent] {1,4}
Border-top-style border-right-style border-bottom-style border-left-style	Estil de la vora superior, dret, inferior o esquerre	[none hidden dotted dashed solid double groove ridge inset outset]
border-style	Estils de diverses vores individuals	[none hidden dotted dashed solid double groove ridge inset outset] {1,4}
Border-top border-right border-bottom border-left	Ample, estil i el color per a la vora superior, dret, inferior o esquerre	[<border-top-width> <border-top-style> <border-top-color>]
border	Ample, l'estil i el color per a les 4 vores	[<border-top-width> <border-top-style> <border-top-color>]

- Contingut

Propietat	Descripció	Valors
width	Ample	[<longitud> <percentatge> auto]
min-width	Ample mínim	[<longitud> <percentatge>]
max-width	Ample màxim	[<longitud> <percentatge> none]
height	Alt	[<longitud> <percentatge> auto]
min-height	Alt mínim	[<longitud> <percentatge>]
max-height	Alt màxim	[<longitud> <percentatge> none]
line-height	Altura entre les bases del text	[normal <número> <longitud> <percentatge>]
vertical-align	Alineació vertical del text	[baseline sub super top text-top middle bottom text-bottom <percentatge> <longitud>]
text-align	Alineació horitzontal del text	[left right center justify]

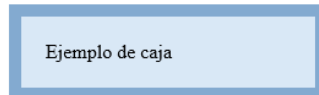
En les taules anteriors, s'inclouen algunes de les propietats d'estil emprades per a definir l'estil de la presentació quan s'empra el model de caixes, pot accedir-se a una referència més àmplia en: <http://www.w3c.es/divulgacion/guiasreferencia/css21/#modeloCajas>. En definir les dimensions de la caixa, cal tindre sempre en compte que les propietats **width** i **height** defineixen l'ample i l'alt del contingut, sense farciment, sense marges i sense bores. La dimensió total de la caixa es calcula sumant les dimensions del contingut, del farciment, de la vora i del marge.

Una caixa sol rebre altres noms com són: contenidor, bloc, capa, etc. Segons l'organisme W3C, una caixa és una zona rectangular constituïda per un contingut, un farciment, una vora i un marge que pot ser tractada de manera independent, com un element únic. De manera que pot ser alterada de diverses formes, per exemple: pot posicionar-se o fins i tot moure's en la finestra de visualització de la pàgina, pot especificar-se una imatge o un color per al fons, es pot escriure damunt o davall d'ella, pot fer-se transparent o opaca, etc. Per exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS - Exemple de caixa</title>
<meta charset="utf-8" />
<style type="text/css">
p {
width: 200px;
border: 10px solid rgb(125,165,205);
padding: 20px;
margin: 100px;
background-color: rgb(215,230,245);
}
</style>
</head>
<body>
<p> Exemple de caixa </p>
</body>
```

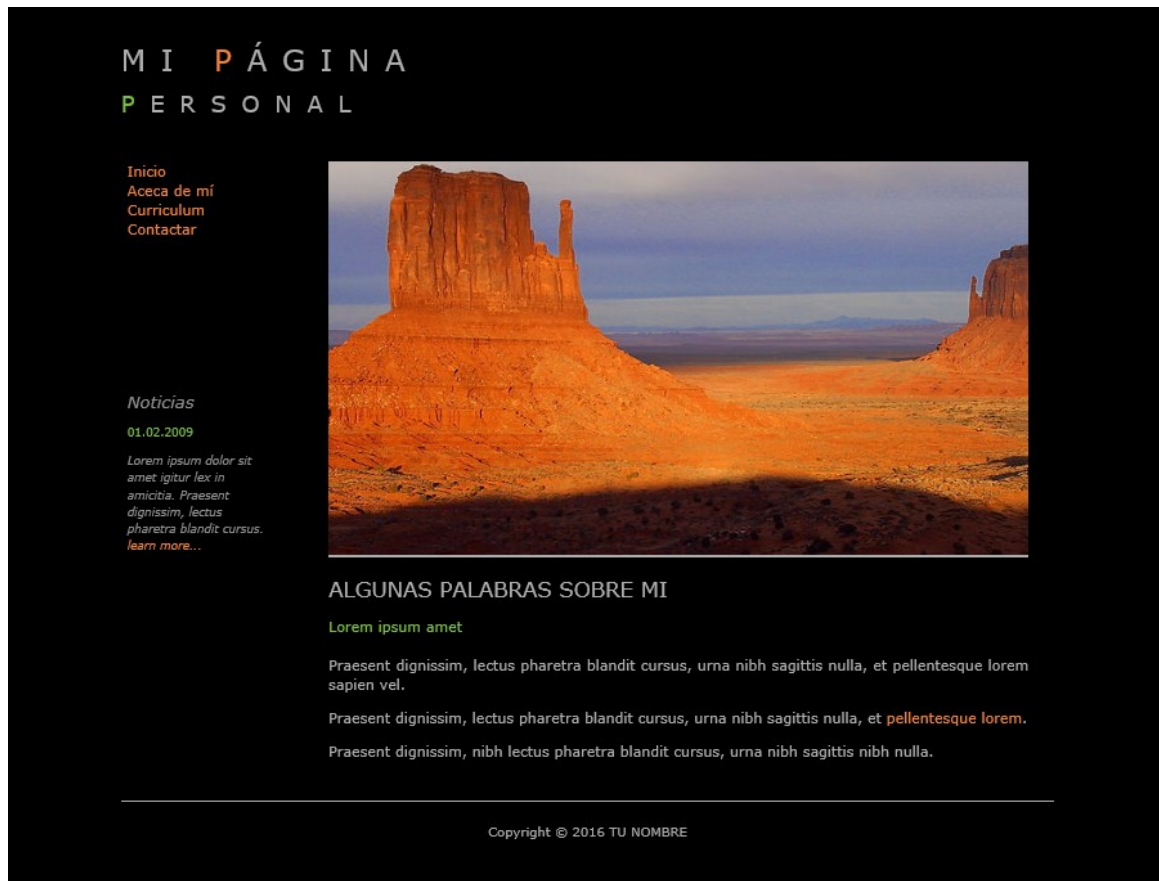
```
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador:



Estructura i composició de pàgines Web

La utilització del model de caixes ofereix múltiples avantatges. El principal avantatge és que permet establir l'estructura visual de les pàgines Web. En l'actualitat, el model de caixes és molt àmpliament utilitzat i constitueix una característica fonamental del disseny visual de les pàgines i dels llocs web. Aquest model de format visual s'utilitza per a definir la composició visual dels elements HTML que formen una pàgina Web. Per a això, se sol fraccionar la pàgina en seccions utilitzant divisions, mitjançant etiquetes <div>. A cada etiqueta <div> se li assigna un identificador, de manera que és possible referir-se a una divisió concreta des d'una fulla d'estil i així, poder aplicar-li les característiques d'estil específiques que siguin apropiades en cada cas: el color de fons, l'estil de les vores, el tipus de lletra, la posició en la pàgina, etc. Per exemple:



El resultat de la presentació de la pàgina Web anterior s'ha obtingut estructurant adequadament els continguts mitjançant l'ús de divisions. En el codi HTML corresponent, que es mostra a continuació, pot apreciar-se l'ús que es fa de les divisions: *contenedor*, *capçalera*, *menu*, *contingut* i *peu* per a estructurar els continguts sobre la pàgina Web.

```
<!DOCTYPE html>
<html>
<head>
<title>CSS - Pàgina personal</title>
<meta charset="utf-8" />
<link rel="stylesheet" type="text/css" href="./PaginaPersonalEstilo.css">
</head>
<body>
<div id="contenedor">
<div id="capçalera">
<h2>EL MEU <span class="orange">P</span>ÁGINA</h2>
<h3><span class="green">P</span>ERSONAL</h3>
</div>
<div id="menu">
<a class="orange" href="">Inici</a br> </>
<a class="orange" href="">Acéca de mí</a br> </>
<a class="orange" href="">Currículum</a br> </>
<a class="orange" href="">Contactar</a br> </>
<div class="noticies">
<h2 class="ramat">Notícies</h2>
<span class="green">01.02.2009</span>
```

```
<p class="ramat">
Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim,
lectus pharetra blandit cursus.
<a class="orange" href="">learn habite...</a>
</p>
</div>
</div>
<div id="contingut">

<h2>ALGUNES PARAULES SOBRE LA MEUA</h2>
<h4 class="green">Lorem ipsum amet</h4>
<p>Praesent dignissim, lectus pharetra blandit cursus, urna nibh sagittis
nulla, et pellentesque lorem sapien vel.</p>
<p>Praesent dignissim, lectus pharetra blandit cursus, urna nibh sagittis
nulla, et <a class="orange underline" href="">pellentesque lorem</a>.</p>
<p>Praesent dignissim, nibh lectus pharetra blandit cursus, urna nibh sagittis
nibh nulla.</p>
</div>
<div id="peu">
<p class="copyright">Copyright &#169; 2016 EL TEU NOM</p>
</div>
</div>
</body>
</html>
```

En l'arxiu de fulla d'estil externa CSS que s'utilitza en l'exemple anterior, que es mostra a continuació, pot apreciar-se l'aplicació de les propietats d'estil corresponents.

```
body{
background: rgb(0,0,0);
color: #a9a9a9;
font-family: Verdana,Arial,sans-serif;
font-size: 11px;
font-weight: 300;
line-height: 15px;
}
#contenedor {
width: 80%;
margin-top: 5px;
margin-left: auto;
margin-right: auto;
padding: 30px;
}
#capçalera {
font-size: 15px;
letter-spacing: 12px;
}
#menu {
float: left;
width: 15%;
height: 400px;
margin: 5px;
margin-top: 20px;
}
#contingut {
float: right;
width: 75%;
text-align: justify;
margin: 20px;
```

```
}
#peu {
  clear: both;
  float: none;
  padding: 5px;
  font-size: 0.9em;
  border-top: solid 1px;
  text-align: center;
}
.orange {
  color: #f47b33;
}
.green {
  color: #74be37;
}
.ramat {
  color: #898989;
  font-style: italic;
}
.noticies {
  text-align: left;
  font-size: 0.8em;
  line-height: 1.5em;
  margin-top: 120px;
}
a { text-decoration: none; }
a:hover { color: #fff; text-decoration: none; }
#imatge {
  width: 100%;
  max-width: 611px;
  max-height: 400px;
  border-bottom: solid 2px;
}
```

Definir l'estructura visual d'una pàgina Web és una tasca complexa, és necessari seguir una sèrie de passos. El primer pas és planificar l'estructura de la pàgina, per a això sol realitzar-se un esbós del resultat final que es pretén aconseguir. A partir de l'esbós, s'escriu el codi HTML de la pàgina, utilitzant etiquetes <div> per a dividir els elements de contingut en parts. El tercer pas, consisteix a afegir els continguts (informació i altres elements HTML) a cadascuna de les divisions. Finalment, es crea una fulla d'estil per a establir el posicionament cadascuna de les divisions i definir el format visual dels elements de la pàgina i poder aconseguir el resultat desitjat.

En els apartats següents s'estudien les propietats de posicionament i de control de la visualització de les caixes, que permeten definir l'estructura o composició visual dels elements de les pàgines i llocs web. En l'àmbit del disseny visual, l'acció de definir la composició o estructura visual d'una pàgina Web sol rebre el nom de maquetació.

6.2 POSICIONAMENT

Per regla general, el posicionament exauto dels elements d'HTML en una pàgina Web és desaconsellable. Això és degut a les carauterístiques de visualització diferents que presenten els diversos tipus de dispositius que, en l'actualitat, poden emprar-se per a visitar les pàgines i els llocs web. Els continguts que presenta qualsevol pàgina Web haurien de poder veure's d'una manera adequada, independentment dels dispositius emprats per a la seua consulta: monitor d'ordinador

de sobretaula, monitor d'ordinador portàtil, Tauleta, Telèfon intel·ligent, etc. En l'actualitat, les fulles d'estil en cascada CSS proporcionen diverses eines per a posicionar, de manera adequada en cada cas, qualsevol element en el seu document HTML. En general, per a posicionar els elements HTML en una pàgina sol recórrer-se a crear un "puzle" d'elements niats en divisions. Les propietats de posicionament s'inclouen en l'especificació CSS2.

Les fulles d'estil CSS permeten utilitzar cinc mètodes diferents per a establir el posicionament d'una caixa en la finestra de visualització del navegador: estàtic, relatiu, absolut, fix i flotant. Les propietats d'estil que poden intervenir per a definir el posicionament d'elements són les següents: position, top, bottom, right, left, float i clear.

Propietat	Descripció	Valors
position	Forma o esquema de posicionament de la caixa	[static relative absolute fixed]
Top right bottom left	Desplaçament vertical i horitzontal de la caixa (respecte al límit superior, dret, inferior o esquerre del contenidor)	[<longitud> <percentatge> auto]
float	Posicionament flotant	[left right none]
clear	Control de caixes adjacents amb posicionament flotant	[none left right both]

Posicionament estàtic

És el valor per defecte establert per a establir el posicionament de tots els elements HTML en la pàgina. Els elements posicionats de manera estàtica apareixen en pantalla segons el flux normal de representació d'HTML i, per tant, no es veuen afectats per les propietats top, bottom, left i right. A més, l'element afectat no pot posicionar-se ni reposicionar-se i la seua visibilitat no pot modificar-se. El posicionament estàtic s'especifica mitjançant la propietat:

```
position: static;
```

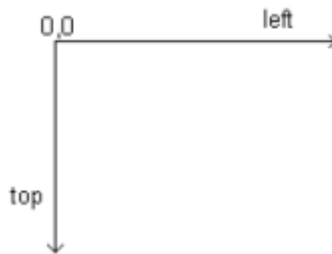
Posicionament relatiu

Consisteix a desplaçar la caixa un cert valor en píxels o en % respecte a la seua posició normal o estàtica. Per tant, l'element roman en el flux de les dades, encara que està, en una certa mesura, desplaçat en relació amb el que seria la seua posició estàtica o normal. El posicionament relatiu actua juntament amb les propietats: top, bottom, right i left. La posició relativa s'especifica mitjançant les coordenades (x,i) on:

- **x**, és la distància respecte a la vora esquerra de l'element pare o de la finestra del navegador (eix horitzontal). D'aquesta manera, la propietat left determina la distància entre la part esquerra de l'element i la vora esquerra de la pàgina i, right la distància entre la part dreta de l'element i la vora dreta de la pàgina.

- **i**, és la distància respecte a la vora superior de l'element pare o de la finestra del navegador (eix vertical). D'aquesta manera, la propietat **top** determina la distància entre la vora superior de l'element i la vora superior de la pàgina i, **bottom** la distància entre la vora inferior de l'element i la vora inferior de la pàgina.

La posició de coordenades (0,0) correspon amb la cantonada superior esquerra de l'element contenidor, és a dir, de l'element pare, o bé, de la finestra de visualització del navegador:



El posicionament relatiu s'especifica mitjançant la propietat:

position: **relative;**

Exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS - Posicionament Estàtic i Relatiu</title>
<meta charset="utf-8" />
<style type="text/css">
.normal {
width: 180px;
border: 2px solid gray;
text-align: center;
}
.relativa {
position: relative;
top: 30px;
left: 60px;
width: 280px;
border: 2px solid gray;
text-align: center;
}
</style>
</head>
<body>
<div class="normal">
Posició estàtica o normal
</div>
<div class="relativa">
Posició relativa, desplaçada mitjançant les propietats top i left
</div>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador:

Posición estática o normal

Posición relativa, desplazada mediante las
propiedades top y left

Posicionament absolut

La posició de la caixa vindrà donada per les coordenades que es definisquen. Així, la caixa no estarà dins del flux estàtic o normal del document, per la qual cosa es podrà solapar o superposar amb altres elements de la pàgina. La resta dels elements es comporten com si no existiren els elements posicionats de manera absoluta. L'element contenidor o pare de l'element que es posiciona de manera absoluta ha d'estar posicionat, encara que siga amb un posicionament relatiu. El posicionament absolut actua juntament amb les propietats: top, bottom, right i left.

El posicionament absolut s'especifica mitjançant la propietat:

```
position: absolute;
```

Exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>Posicionament absolut</title>
<meta charset="utf-8" />
<style type="text/css">
.absoluta {
position: absolute;
left: 75px;
top: 100px;
width: 180px;
border: 2px solid black;
text-align: center;
}
</style>
</head>
<body>
<div class="absoluta">
Divisió amb esquema de posicionament absoluta
</div>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador:

División con esquema de
posicionamiento absoluta

Quan s'utilitza el posicionament absolut es corre el risc que no es mostre la pàgina correctament si s'empren resolucions de pantalla diferents a la utilitzada durant el temps de disseny. Això és pel fet que els elements que s'hagen posionat de manera absoluta poden superposar-se sobre els elements que es posicionen seguint el flux normal de la pàgina. Atés que, en l'actualitat, s'empren multitud de dispositius per a utilitzar el servei d'informació Web, es recomana no utilitzar el posicionament absolut. Convé, a més, recordar que els navegadors Web agreguen, de manera predeterminada, un marge al cos de la pàgina i que aquest marge varia d'un navegador a un altre. Per això, és prudent, en cas d'usar posicionament absolut, especificar els marges de l'etiqueta <body>.

Posicionament fixed

Implica un posicionament absolut, però mantenint la posició de la caixa fins i tot quan l'usuari desplaça la pàgina mitjançant les barres de desplaçament. D'aquesta manera, la caixa estarà sempre visible. Resulta útil per a encapçalats o peus de pàgines, així com per als índexs de les pàgines que es mantindran inalterables en un document extens. El posicionament fix actua juntament amb les propietats: top, bottom, right i left.

El posicionament fix s'especifica mitjançant la propietat:

```
position: fixed;
```

Exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>Posicionament fix</title>
<meta charset="utf-8" />
<style type="text/css">
.fixa {
position: fixed;
left: 75px;
```

```
top: 100px;  
width: 180px;  
border: 2px solid black;  
text-align: center;  
}  
</style>  
</head>  
<body>  
<div class="fixa">  
Divisió amb esquema de posicionament fix  
</div>  
</body>  
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador:

División con esquema de
posicionamiento fijo

Posicionament flotant

El posicionament flotant és molt utilitzat en l'actualitat. Desplaça horitzontalment la caixa tot el que siga possible cap a l'esquerra o cap a la dreta dins contenidor en el qual es troba. La resta de les caixes ocupen el lloc deixat per la caixa flotant. Els elements que hi haguera abans de l'element float no es veuen afectats. Utilitza les propietats float i clear.

La propietat **float** permet situar una caixa el més a la dreta o el més a l'esquerra possible en la mateixa línia, dins del seu contenidor. Aquesta propietat pot adoptar els següents valors:

- right. Alinea a la dreta l'element indicat, espentant als altres elements cap a l'esquerra.
- left. Alinea a l'esquerra l'element indicat, espentant als altres elements cap a la dreta.
- none. No especifica res i deixa la gestió al navegador per a situar la caixa segons el flux normal o estàtic de presentació del document.

Amb el posicionament flotant, no es produeix superposició o solapament de caixes perquè la propietat float té en compte la resta de les caixes flotants. Una caixa flotant ha de tindre definida

la seua amplària, de manera implícita o explícita, ja que en cas contrari serà tractada com no flotant.

Exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>Posicionament flotant</title>
<meta charset="utf-8" />
<style type="text/css">
.dreta {
float: right;
}
.esquerra {
float: left;
}
</style>
</head>
<body>
<div>
<div class="dreta">

</div>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Set
senar risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies set, dolor. Cras elementum ultrices
diam. Maecenas ligula massa, varius a, semper congue, euismod senar,
el meu.
</p>
</div>
</body>
</html>
```

En el codi anterior pot apreciar-se que l'ús de la declaració `float: right;` força a la imatge a posicionar-se en la part dreta del contenidor. A més, el text del paràgraf es distribueix a l'esquerra. Aquest mateix exemple pot modificar-se per a aplicar a la imatge la classe d'estil denominada esquerra. L'exemple anterior, produeix el següent resultat en el navegador.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi.



La propietat **clear** anul·la l'efecte introduït per la propietat float i afecta als elements que hi haguera darrere de l'element "float". S'utilitza juntament amb la propietat float per a indicar si un element flotant permet altres elements flotants al costat d'ell. El valor assignat estableix el costat que no ha de ser adjacent a cap caixa flotant. Així, el valor right anul·la els elements flotants a la dreta. El valor left anul·la els elements flotants a l'esquerra. I, el valor both anul·la els elements flotants de tots dos costats. Per exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>Posicionament flotant. Ús de la propietat clear</title>
<meta charset="utf-8" />
<style type="text/css">
.dreta {
float: right;
}
.esquerra {
float: left;
}
.buidar {
clear: right;
}
</style>
</head>
<body>
<div>
<div class="dreta">

</div>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Set
senar risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies set, dolor. Cras elementum ultrices
diam. Maecenas ligula massa, varius a, semper congue, euismod senar,
el meu.
</p>
<p class="buidar">
Set senar risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies set, dolor. Cras elementum ultrices
```

diam. Maecenas ligula massa, varius a, semper congue, euismod
senar, el meu.

```
</p>  
</div>  
</body>  
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus.
Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,
dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a,
semper congue, euismod non, mi.



Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum
ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi.

Si en el codi HTML de l'exemple anterior, s'eliminara l'assignació de la classe d'estil *buidar* a l'últim
paràgraf, el resultat en la finestra de visualització del navegador seria el següent.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus.
Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,
dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a,
semper congue, euismod non, mi.

Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec,
ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa,
varius a, semper congue, euismod non, mi.



Es recomana realitzar exercicis per a poder comprovar el funcionament dels diferents tipus de
posicionament de les caixes i, d'aqueixa manera, tindre l'oportunitat d'experimentar i poder
comprendre el seu comportament.

6.3 CONTROL DE LA VISUALITZACIÓ

Les propietats de visualització permeten controlar la presentació de les caixes en la finestra del navegador. Aquestes propietats d'estil poden controlar efectes visuals com: la superposició, el desbordament, la retallada, la visibilitat i la visualització de les caixes. Les propietats corresponents als efectes visuals relacionades amb el control de la visualització de caixes són, respectivament, les següents: z-index, overflow, clip, visibility i display, respectivament.

Superposició

En el cas que alguna caixa se superpose o solape amb una altra, es pot determinar quin estarà davant utilitzant la propietat z-index. El valor d'aquesta propietat pren un valor sencer per a cada caixa. La caixa a la qual se li assigne un valor major serà la que apareixerà per davant de les altres en la finestra de visualització. La propietat z-index només funciona amb un posicionament absolut.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Superposició (z-index)</title>
<meta charset="utf-8" />
<style type="text/css">
.caixes {
position: absolute;
width: 180px;
height: 140px;
padding: 4px;
border: 2px solid gray;
}
.cajal {
left: 20px;
top: 20px;
background-color: rgb(125,165,205);
z-index: 1;
}

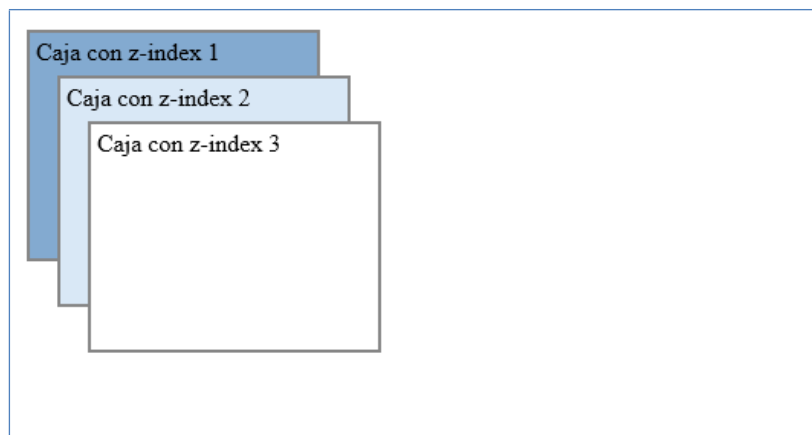
.caixa2 {
left: 40px;
top: 50px;
background-color: rgb(215,230,245);
z-index: 2;
}

.caixa3 {
left: 60px;
top: 80px;
background-color: white;
z-index : 3;
}
</style>
</head>
<body>
<div class="caixes cajal">
Caixa amb z-index 1
</div>
<div class="caixes caja2">
```

```
Caixa amb z-index 2
</div>
<div class="caixes caja3">
Caixa amb z-index 3
</div>
</body>
</html>
```

En el codi anterior, pot apreciar-se l'ús que es fa de la propietat `z-index`. A més, també pot apreciar-se l'ús de l'atribut `type` en l'etiqueta `<style>`. L'atribut `type` especifica el tipus de mitjà, anteriorment conegut com a tipus ACARONE, que afecta el contingut d'una etiqueta. Així, l'etiqueta `<style type="text/css">` identifica que el contingut entre les etiquetes `<style>` i `</style>` és codi escrit mitjançant llenguatge d'estil CSS. En l'especificació HTML5, el valor per defecte de l'atribut `type` de la etiqueta `<style>` és "text/css", per la qual cosa ja no és necessari definir-lo. D'altra banda, i com a recordatori, també pot apreciar-se l'ús que es fa de l'atribut `lang` de l'etiqueta `<html>`. Aquest atribut especifica l'idioma dels valors dels atributs i del text contingut en un element. Així, l'etiqueta `<html lang="es">` especifica que l'idioma dels continguts dels elements de la pàgina i dels valors dels seus atributs és espanyol. L'objectiu de l'atribut `lang` és permetre als navegadors representar el contingut de la pàgina de forma més significativa, segons la pràctica cultural acceptada per a un idioma donat. La funcionalitat que aporta aquest atribut depèn de les prestacions i configuració del navegador utilitzat per l'usuari.

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.



Desbordament

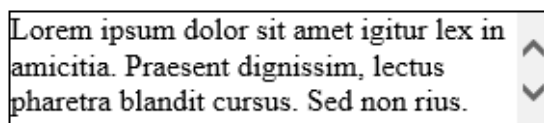
La propietat `overflow` determina les accions que ha de fer el navegador quan un element és més gran que el seu contenidor. Ha de recordar-se que el contenidor d'un element és l'element pare que el conté. Els possibles valors d'aquesta propietat són:

- `hidden`. Oculta la part que es desborda, sense possibilitat de visualitzar o accedir a ella.
- `scroll`. Oculta la part que es desborda, encara que queda accessible mitjançant les barres de desplaçament.

- visible. La part que es desborda es mostra, ignorant l'especificació de l'element pare que el conté.
- auto. Es deixa la gestió del desbordament al navegador.

```
<!DOCTYPE html>
<html>
<head>
<title>Desbordament (overflow)</title>
  <meta charset="utf-8" />
  <style type="text/css">
    .caixa {
      width: 280px;
      height: 60px;
      border: 1px solid black;
      overflow: auto;
    }
  </style>
</head>
<body>
  <div class="caixa">
    Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim,
    lectus pharetra blandit cursus. Set senar rius. Suspenguera lectus tortor,
    digniessim sit amet, adipiscing nec, utricies set, dolor. Gras elementum
    ultrices diam.
  </div>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.



Retallada

La propietat clip determina la part visible d'un element, generalment una imatge. De manera que la imatge sencera s'inclourà en el document, encara que només una part rectangular serà visible. Es recomane incloure l'element que es desitja retallar dins d'una etiqueta <div>. Aquesta propietat només funciona amb un posicionament absolut. Les coordenades del rectangle retallat venen donades per les cantonades superior-esquerra, superior-dreta, inferior-dreta i inferior-esquerra respecte de la imatge inicial.

```
<!DOCTYPE html>
<html>
<head>
<title>Retallada (clip)</title>
<meta charset="utf-8" />
<style>
.retallada {
position: absolute;
clip: rect(0px 170px 215px 0px);
}
</style>
</head>
<body>
<div class="retallada">
<p></p>

</div>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.



Visibilitat

La propietat `visibility` determina si un element està visible o ocult. El valor `visible` mostra l'element, mentre que el valor `hidden` oculta l'element.

```
<!DOCTYPE html>
<html>
<head>
<title>Visibilitat (visibility)</title>
<meta charset="utf-8" />
<style>
.ocult {
visibility: hidden;
}
</style>
</head>
```

```
<body>
<p>
Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim,
lectus pharetra blandit cursus.
</p>
<div class="ocult">

</div>
<p>
Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim,
lectus pharetra blandit cursus. Set senar rius.
</p>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim, lectus pharetra blandit cursus.

Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim, lectus pharetra blandit cursus. Sed non rius.

Si en el codi de l'exemple anterior s'elimina l'assignació a la classe d'estil ocult en l'etiqueta <div> corresponent, es produirà el següent resultat.

Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim, lectus pharetra blandit cursus.



Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim, lectus pharetra blandit cursus. Sed non rius.

Visualització

La propietat d'estil més característica que pot aplicar-se per a controlar la visualització dels elements en la pàgina és la propietat **display**. Es tracta d'una propietat molt potent l'ús de la qual està àmpliament estès. Permet controlar gran diversitat de característiques de presentació dels elements. Els valors més representatius de la propietat display són: none, inline, block, run-in, list-item, inline-block, table, inline-table, table-row-group, table-row, table-column-group, table-column, table-cell i table-caption. Alguns d'aquests valors s'estudien a continuació.

El valor **none** de la propietat display, defineix un element que no es mostrarà. L'element afectat es retira del document i de la visualització de la pàgina en el navegador. Per exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>Visualització (display) - none</title>
<meta charset="utf-8" />
<style>
.nomostrar {
display: none;
}
</style>
</head>
<body>
<p>
Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim,
lectus pharetra blandit cursus.
</p>
<div class="nomostrar">

</div>
<p>
Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim,
lectus pharetra blandit cursus. Set senar rius.
</p>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim, lectus pharetra blandit cursus.

Lorem ipsum dolor sit amet igitur lex in amicitia. Praesent dignissim, lectus pharetra blandit cursus. Sed non rius.

En l'exemple anterior, pot apreciar-se la diferència existent respecte a l'ús de la declaració visibility: hidden; estudiada anteriorment. A més, pot apreciar-se l'efecte que produiria l'eliminació de l'assignació a la classe d'estil nomostrar en l'etiqueta <div> corresponent.

Ja s'han estudiat, els conceptes d'element de bloc i element de línia. Els elements de bloc creen blocs de contingut vertical en el context visual, generalment usant salts de línia abans i després del seu contingut. Els elements de línia es presenten horitzontalment, uns darrere d'uns altres. La propietat d'estil display permet redefinir un element de línia com un element de bloc i al revés. Per a això, s'utilitzen els valors **inline** i **block** de la propietat d'estil display. Per exemple :

```
<!DOCTYPE html>
<html>
<head>
<title>Visualització (display) - block/inline</title>
<meta charset="utf-8" />
<style>
ul {
list-style: none;
padding: 5px;
}
li {
display: inline;
border: 1px solid gray;
background-color: rgb(215,230,245);
text-align:center;
margin-left: 3px;
margin-top: 6px;
padding-right: 15px;
padding-left: 15px;
width: 80px;
}
</style>
</head>
<body>
<ul>
<li>Opció 1</li>
<li>Opció 2</li>
<li>Opció 3</li>
<li>Opció 4</li>
</ul>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

Opción 1 Opción 2 Opción 3 Opción 4

Els elements de les llistes són elements de bloc, per la qual cosa es presenten els uns davall dels altres. En l'exemple anterior, es presenten en una única línia horitzontal en aplicar la declaració d'estil display: inline;. Pot comprovar-se l'efecte que es produiria en eliminar aquesta declaració.

L'ús de l'element `<table>` d'HTML presenta alguns problemes en la construcció de pàgines Web. Aquests problemes són causats per la seua falta d'adaptació a la construcció de pàgines Web que complisquen amb les normes bàsiques d'Accessibilitat Web i que s'adapten a les carauterístiques de visualització d'alguns tipus de dispositius. Per aquests motius, sol emprar-se la propietat d'estil `display` per a crear taules en pàgines Web. Les taules creades amb la propietat `display` permeten presentar informació en format tabular d'una forma que és compatible amb les normes d'Accessibilitat Web actuals i que queda adaptada a les carauterístiques específiques de visualització de diversos tipus de dispositius. Per tant, la creació de taules mitjançant la propietat d'estil `display` permet crear taules en pàgines Web accessibles i adaptatives (*responsive*).

S'utilitzen diversos valors de la propietat d'estil `display` per a construir i mostrar una taula, emprant per a això divisions. Alguns d'aquests valors són els següents: **table**, **inline-table**, **table-row**, **table-row-group**, **table-column**, **table-column-group**, **table-cell** i **table-caption**.

- **table**. Serveix per a crear una taula que ocupa l'ample del seu contenidor.
- **Inline-table**. Serveix per a dissenyar una taula de tipus línia.
- **table-row**. Crea una fila de la taula. Fa que qualsevol element es mostre com una fila d'una taula.
- **table-row-group**. Reagrupa una o diverses files de la taula.
- **table-column**. Mostra una columna de la taula.
- **table-column-group**. Reagrupa una o diverses columnes de la taula.
- **table-cell**. Crea i mostra una cel·la de la taula.
- **table-caption**. Crea i mostra la llegenda de la taula. Per defecte, apareix en la part superior de la taula, però la seua posició es pot modificar amb la propietat `caption-side`.

A continuació, s'inclou un exemple de codi per a la presentació d'informació en format tabular emprant la propietat d'estil `display`.

```
<!DOCTYPE html>
<html>
<head>
<title>Visualització - Display Table 1</title>
<meta charset="utf-8" />
<style>
.contenedor{
padding-top: 20px;
font-family: Arial;
}
.taula{
display: table;
width: 50%;
margin-left: auto;
margin-right: auto;
```



```
border: 1px solid;
}
.fila{
display: table-row;
}
.cel·la{
display: table-cell;
padding: 5px;
}
.titule{
display: table-caption;
text-align: center;
padding-bottom: 5px;
}
.agruparcolumnas{
display: table-column-group;
}
</style>
</head>
<body>
<div class="contenedor">
<div class="taula">
<div class="titule">CLIENTS</div>
<div class="fila" style="background:#00ffff">
<div class="cel·la">Nom</div>
<div class="cel·la">Direcció</div>
<div class="cel·la">Ciutat</div>
<div class="cel·la">Telefono</div>
</div>
<div class="fila">
<div class="cel·la">Francisco Campos LLanda</div>
<div class="cel·la">C/ La Granota, 4</div>
<div class="cel·la">Almansa (Albacete)</div>
<div class="cel·la">678 33 22 11</div>
</div>
<div class="fila">
<div class="cel·la">María Fernández García</div>
<div class="cel·la">C/ Major, 32</div>
<div class="cel·la">Aspe (Alacant)</div>
<div class="cel·la">643 77 88 99</div>
</div>
<div class="fila">
<div class="cel·la">Jordi Moll Torrens</div>
<div class="cel·la">C/ Baix, 2</div>
<div class="cel·la">Xaló (Alacant)</div>
<div class="cel·la">677 78 78 00</div>
</div>
</div>
</div>
</body>
</html>
```

L'exemple anterior produeix el següent resultat de representació d'informació en format tabular en la finestra de visualització del navegador.

CLIENTES

Nombre	Dirección	Ciudad	Telefono
Francisco Campos LLanda	C/ La Rana, 4	Almansa (Albacete)	678 33 22 11
María Fernández García	C/ Mayor, 32	Aspe (Alicante)	643 77 88 99
Jordi Moll Torrens	C/ Baix, 2	Xaló (Alacant)	677 78 78 00

La construcció de taules mitjançant la propietat d'estil display també se sol aplicar per a resoldre la composició o estructura visual i el posicionament d'elements complexos, com, per exemple, els controls i elements que formen un formulari. Per exemple :

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<meta charset="utf-8" />
<style>
.cos {
display: table;
margin-left: auto;
margin-right: auto;
padding: 10px;
width: 60%;
}
.línia {
display: table-row;
height: 25px;
}
.textos {
display: table-cell;
text-align: right;
width: 40%;
padding-right: 10px;
}
.controls {
display: table-cell;
text-align: left;
width: 50%;
}
.boton {
text-align: center;
}
</style>
</head>
<body>
<div style="text-align: center;">
<h2>DADES DE PRODUCTES</h2>
</div>
<form id="form1" method="post" action="http://server/maneja.aspx">
<div class="cos">
```

```
<div class="línia">
<div class="textos">
Id. de Producte
</div>
<div class="controls">
<input type="text" name="Codigo" size="10"/>
</div>
</div>
<div class="línia">
<div class="textos">
Descripció
</div>
<div class="controls">
<input type="text" name="Descripcion" size="40" />
</div>
</div>
<div class="línia">
<div class="textos">
Preu
</div>
<div class="controls">
<input type="text" name="Preu" />
</div>
</div>
<div class="boton">
<br />
<input type="submit" name="Codigo" value="Enviar" />
</div>
</form>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

DATOS DE PRODUCTOS

Id. de Producto	<input type="text"/>
Descripción	<input type="text"/>
Precio	<input type="text"/>
<input type="button" value="Enviar"/>	

La composició i el posicionament dels controls i elements que formen un formulari també sol resoldre's emprant un posicionament flotant, mitjançant l'ús de la propietat d'estil float. El següent codi produeix una presentació idèntica a la de l'exemple anterior.

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<meta charset="utf-8" />
<style>
.cos {
display: table;
margin-left: auto;
margin-right: auto;
padding: 10px;
width: 60%;
}
.línia
{
height: 25px;
}
.textos
{
text-align: right;
width: 40%;
position: relative;
float: left;
}
.controls
{
text-align: left;
width: 58%;
float: right;
}
.final
{
clear: both;
}
.boton {
text-align: center;
}
</style>
</head>
<body>
<div style="text-align: center;">
<h2>DADES DE PRODUCTES</h2>
</div>
<form id="form1" method="post" action="http://server/maneja.aspx">
<div class="cos">
<div class="línia">
<div class="textos">
Id. de Producte
</div>
<div class="controls">
<input type="text" name="Codigo" size="10"/>
</div>
</div>
<div class="línia">
<div class="textos">
```

```
Descripció
</div>
<div class="controls">
<input type="text" name="Descripcion" size="40" />
</div>
</div>
<div class="línia">
<div class="textos">
Preu
</div>
<div class="controls">
<input type="text" name="Preu" />
</div>
</div>
<div class="final">

</div>
<div class="boton">
<br />
<input type="submit" name="Codigo" value="Enviar" />
</div>
</form>
</body>
</html>
```

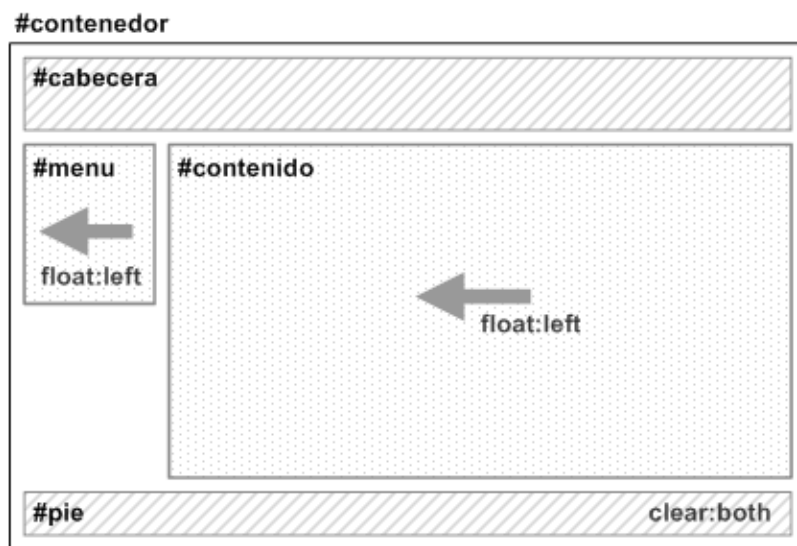
6.4 ESTRUCTURA I LAYOUT

S'utilitza el terme layout per a referir-se a la composició o estructura visual general d'una pàgina Web. La forma habitual de crear el layout d'una pàgina és utilitzant divisions, mitjançant l'ús de l'etiqueta <div>. L'acció de definir el layout d'una pàgina Web es coneix com a maquetació.

Es poden classificar els dissenys del layout de les pàgines Web segons diversos criteris: el nombre de columnes, la posició dels elements de navegació, l'ample fix o variable dels elements, etc. S'utilitza el terme **layout fluid o líquid** (*fluïu layout* o *liquid layout*) per a referir-se a aquella mena de layout que utilitza un ample variable, mitjançant valors de percentatge, per a definir l'ample de les divisions que ho componen. Així, una pàgina web posseeix un disseny fluid quan la seua grandària s'ajusta a la dimensió horitzontal de la pantalla de manera automàtica, sense que hi haja necessitat d'utilitzar una barra de desplaçament horitzontal (*scroll*). D'aquesta manera, el disseny s'expandeix a l'ample disponible de la finestra de visualització o de la pantalla perquè la grandària de les divisions que el formen (elements <div>) és un percentatge del total disponible (100%). L'objectiu d'aquesta mena de disseny de layout és aconseguir que es produïska un ajust automàtic del contingut de la pàgina Web a les mesures i resolució del dispositiu, de la mateixa forma que un líquid s'adapta a la forma del recipient o contenidor on es troba. El problema sorgeix en pantalles molt grans amb resolucions altes, perquè solen aparèixer grans espais en blanc. I també solen aparèixer problemes de visualització en pantalles xicotetes amb resolucions baixes inferiors a 1.024 px d'ample, per exemple, en *tauletes* en orientació vertical (*portrait*) i en *telèfons intel·ligents* en orientació horitzontal (*landscape*), perquè les imatges es miniaturitzen i els textos es tornen il·legibles. Per a evitar aquests problemes se sol fixar un ample mínim a les divisions que componen el layout, la qual cosa fa que apareguen les barres de desplaçament horitzontal quan siga necessari.

Disseny fluid a dues columnes amb capçalera i peu de pàgina

Per a obtenir un disseny de layout fluid a dues columnes amb capçalera i peu de pàgina, tal com es mostra següent figura, s'utilitza la propietat **float**. Aquesta propietat s'aplicarà als elements posicionats, com l'element menú i l'element contingut. A més, s'aplicarà la propietat **clear** a l'element peu, per a evitar els solapaments ocasionats pels elements posicionats amb float. I s'aplicarà una amplària relativa en % als elements de la pàgina perquè s'adapten o fluïsqüen sobre la finestra de visualització del navegador per a diversos tipus de resolucions.



S'utilitzaran, per tant, com a mínim les següents declaracions d'estil CSS:

```
#menu {  
  float: left;  
  width: 15%;  
}  
#contingut {  
  float: left;  
  width: 85%;  
}  
#peu {  
  clear: both;  
}
```

Les anteriors declaracions s'aplicaran als elements `<div>` següents en el codi HTML:

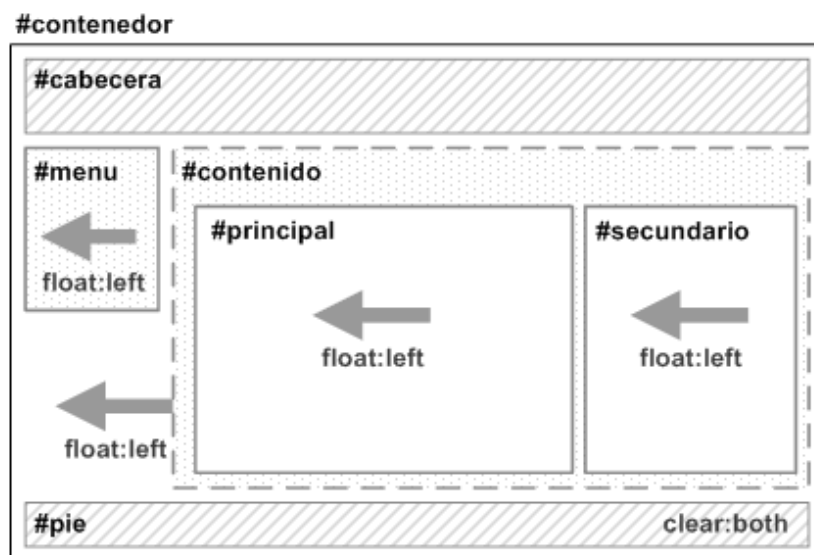
```
<body>  
<div id="contenedor">  
  <div id="capçalera">  
  </div>  
  
  <div id="menu">  
  </div>
```

```
<div id="contingut">
</div>

<div id="peu">
</div>
</div>
</body>
```

Disseny fluid a tres columnes amb capçalera i peu de pàgina

En la següent il·lustració es mostra un cas de layout a 3 columnes amb capçalera i peu de pàgina.



El codi CSS mínim per a definir l'estructura de la pàgina, sense aplicar cap estil addicional, és:

```
#menu {
float: left;
width: 15%;
}
#contingut {
float: left;
width: 85%;
}
#contingut #principal {
float: left;
width: 80%;
}
#contingut #secundari {
float: left;
width: 20%;
}

#peu {
clear: both;
}
```

I el codi HTML per a definir l'estructura de la pàgina és el següent:

```
<body>
<div id="contenedor">
<div id="capçalera">
</div>

<div id="menu">
</div>

<div id="contingut">
<div id="principal">
</div>

<div id="secundari">
</div>
</div>

<div id="peu">
</div>
</div>
</body>
```

Exemple. A continuació, s'inclou un exemple de codi complet per a obtenir un disseny fluid d'una pàgina Web amb layout a dues columnes amb capçalera i peu de pàgina.

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<meta charset="utf-8" />
<style>
body {
font-family: Arial;
}
a:link {
color: #069;
text-decoration: none;
}
a:hover {
text-decoration: underline;
}
#contendor {
width: 100%;
margin: 0px;
padding: 0;
min-width: 500px;
overflow: auto;
}
#menu {
float: left;
width: 18%;
margin: 1%;
background-color: #ddd;
}
#contingut {
float: right;
width: 77%;
```



```
margin: 1%;
overflow: hidden;
text-align: justify;
}
#peu {
font-size: 80%;
color: #aaa;
background-color: #eee;
height: 100px;
overflow: auto;
border-top: 1px solid #000;
padding-top: 5px;
text-align: center;
clear: both;
}
ul {
list-style: none;
margin: 0px;
padding-left: 5px;
}
#enllaços li{
display: inline;
}
#enllaços a {
text-align: center;
margin: 1px;
padding: 5px;
width: 80px;
font-weight: bold;
background-color: #eee;
border-top-right-radius: 8px;
}
#enllaços a:hover {
color: white;
font-weight: bold;
background-color: #069;
text-decoration: underline;
border-top-right-radius: 8px;
}
#opcions li {
border-top: 1px solid #ccc;
font-size: 90%;
margin-bottom: 5px;
}
#barra li {
border-left: 5px solid rgb(192,192,192);
font-size: 80%;
padding-left: 15px;
padding-top: 5px;
}
</style>
</head>
<body>
<div id="capçalera">
<h1>Lorem ipsum dolor sit amet</h1>
</div>
<div id="enllaços">
<ul>
<li><a href="#">Opció 1</a></li>
<li><a href="#">Opció 2</a></li>
```

```
<li><a href="#">Opció 3</a></li>
<li><a href="#">Opció 4</a></li>
</ul>
</div>
<hr />
<div id="contendor">
<div id="menu">
<div id="opcions">
<p>Opcions</p>
<ul>
<li><a href="#">Opció 1</a></li>
<li><a href="#">Opció 2</a></li>
<li><a href="#">Opció 3</a></li>
<li><a href="#">Opció 4</a></li>
</ul>
</div>
<br /> <br /> <br />
<div id="barra">
<p>Enllaços d'interés</p>
<ul>
<li><a href="#">Pàgina 1</a></li>
<li><a href="#">Pàgina 2</a></li>
<li><a href="#">Pàgina 3</a></li>
<li><a href="#">Pàgina 4</a></li>
</ul>
</div>
<br />
</div>
<div id="contingut">
<h3>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</h3>
<p>
Maecenas senar tortor. Vestibulum davant ipsum primis in faucibus orci
luctus et ultrices posuere cubilia Curae; Set fermentum lorem a velit .
</p>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin sodales,
enim in volutpat vehicula, llig turpis vehicula magna, ut rutrum arcu lorem
ac pede.
</p>
<p>
Curabitur rutrum eros a risus . Cum sociis natoque penatibus et magnis dis
parturient muntanyes, nascetur ridiculus mus. In molestie suscipit allibere.
Cras sem. Nunc senar tellus et urna mattis tempor. Nulla nec tellus a quam.
hendrerit venenatis. Suspendisse pellentesque odi et est. Morbi set nisl
set dui consequat sodales. Donec porta porta ligula. Nam erat massa, blandit
in, dapibus sit amet, vestibulum et, augue. Suspendisse ac risus. Duis
semper fringilla sem. Praesent augue arcu, scelerisque nec, ornara
malesuada, posuere a, neque. Nullam nulla nisi, ultrices quis, adipiscing
senar, varius ut, dui. Nulla viverra pellentesque sem.
</p>
</div>
<div id="peu">
La meua pàgina Web - 2016
</div>
</div>
</body>
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

Lorem ipsum dolor sit amet

[Opción 1](#) [Opción 2](#) [Opción 3](#) [Opción 4](#)

Opciones

[Opción 1](#)
[Opción 2](#)
[Opción 3](#)
[Opción 4](#)

Enlaces de interés

[Página 1](#)
[Página 2](#)
[Página 3](#)
[Página 4](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Maecenas non tortor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed fermentum lorem a velit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin sodales, enim in volutpat vehicula, leo turpis vehicula magna, ut rutrum arcu lorem ac pede.

Curabitur rutrum eros a risus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In molestie suscipit libero. Cras sem. Nunc non tellus et urna mattis tempor. Nulla nec tellus a quam hendrerit venenatis. Suspendisse pellentesque odio et est. Morbi sed nisl sed dui consequat sodales. Donec porta porta ligula. Nam erat massa, blandit in, dapibus sit amet, vestibulum et, augue. Suspendisse ac risus. Duis semper fringilla sem. Praesent augue arcu, scelerisque nec, ornare malesuada, posuere a, neque. Nullam nulla nisi, ultrices quis, adipiscing non, varius ut, dui. Nulla viverra pellentesque sem.

Mi página Web - 2016

El codi de l'exemple anterior crea una pàgina amb amplària variable que s'adapta a les dimensions de la finestra de visualització del navegador. Pot provar-se a restaurar la finestra del navegador i modificar la seua grandària per a comprovar la seua adaptació a les dimensions. Per a definir una pàgina amb amplàries fixa, bastaria amb substituir les amplàries en percentatges per amplàries en nombre de píxels .

Disseny de layout amb taules

El disseny de layout de les pàgines Web també pot resoldre's mitjançant l'ús de taules, aplicant la propietat d'estil **display** als diversos elements que la componen.



Tradicionalment, els documents en paper han utilitzat les denominades quadrícules de disseny per a situar amb harmonia els diferents blocs de lectura. Les quadrícules estan formades per un conjunt de files i columnes que permeten organitzar el contingut i millorar la comprensió del lector. El disseny del layout de pàgines Web amb la propietat d'estil display i els valors table, table-row i table-cell, proporciona els següents avantatges als dissenyadors Web:

- L'altura de totes les cel·les d'una fila és sempre la mateixa, la qual cosa proporciona coherència visual a la composició. No obstant això, si s'utilitzen altres mètodes de disseny de layout, llavors l'altura de les divisions de les files depèn del contingut que incloguen.
- La propietat vertical-align gestiona perfectament l'alineació vertical sobre la cel·la.
- Poden establir-se vores en un sol costat de les cel·les. O també, pot aplicar-se estils diferents a cadascun de les vores: superior, inferior, dreta i esquerra.
- Poden utilitzar-se unitats diferents, fixes en píxels i relatives en percentatge, per a establir l'amplada i l'altura de les cel·les.
- No es produeixen problemes de flotació dels elements adjacents, ja que totes les cel·les, i per tant tots els elements de la composició, estan el flux normal de la pàgina.
- Es genera un codi net, semàntic, lleuger i clar.

Exemple. A continuació, s'inclou un exemple de codi complet per a obtenir un disseny de pàgina Web mitjançant taula amb layout a tres columnes amb capçalera i peu de pàgina, emprant per a això la propietat d'estil display i els valors: table, table-row i table-cell.

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Estructura i Layout - Display Table 1</title>
  <meta charset="utf-8" />
<style>
#contenedor{
width: 90%;
margin: 10px auto;
font-family: Arial;
}
#capçalera{
padding-top: 30px;
}
#cos{
display: table;
border-spacing: 5px;
background: #ccc;
border: 1px solid #333;
margin-top: 20px;
}
#continguts{
display: table-row;
background: #fcfcfc;
}
#secundari{
width: 20%;
display: table-cell;
border: 1px solid #FFF;
padding: .5em;
}
#principal{
width: 60%;
display: table-cell;
border: 1px solid #FFF;
padding: .5em;
}
#lateral{
width: 20%;
display: table-cell;
border: 1px solid #FFF;
padding: .5em;
}
#peu{
padding-top: 50px;
text-align: center;
}
ul {
list-style: none;
margin: 0px;
padding-left: 5px;
min-width: 500px;
}
li{
display: inline;
}
#menu a {
text-align: center;
margin: 1px;
padding: 5px;
width: 90px;
```

```
font-weight: bold;
background-color: #ccc;
text-decoration: none;
}
#menu a:hover {
color: white;
background-color: #069;
text-decoration: none;
}
</style>
</head>
<body>
<div id="contenedor">
<div id="capçalera">
<h1>LOREM IPSUM DOLOR</h1>
</div>
<div id="menu">
<ul>
<li><a href="#">Opció 1</a></li>
<li><a href="#">Opció 2</a></li>
<li><a href="#">Opció 3</a></li>
<li><a href="#">Opció 4</a></li>
</ul>
</div>
<div id="cos">
<div id="continguts">

<div id="secundari">
<h3>Curabitur rutrum eros a risus .</h3>
<p>
Cum sociis natoque penatibus et magnis dis parturient muntanyes,
nascetur ridiculus mus. In molestie suscipit allibere.
</p>
<p>
Cras sem. Nunc senar tellus et urna mattis tempor. Nulla nec tellus a
quam hendrerit venenatis.
</p>
<p>Suspendisse pellentesque odi et est.</p>
</div>

<div id="principal">
<h3>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</h3>
<p>
Maecenas senar tortor. Vestibulum davant ipsum primis in faucibus orci
luctus et ultrices posuere cubilia Curae; Set fermentum lorem a
velit.
</p>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin
sodales, enim in volutpat vehicula, llig turpis vehicula magna, ut
rutrum arcu lorem ac pede.</p>
<p>
Curabitur rutrum eros a risus . Cum sociis natoque penatibus et
magnis dis parturient muntanyes, nascetur ridiculus mus. In molestie
suscipit allibere. Cras sem. Nunc senar tellus et urna mattis tempor.
Nulla nec tellus a quam hendrerit venenatis. Suspendisse
pellentesque odi et est. Morbi set nisl set dui consequat sodales.
Donec porta porta ligula. Nam erat massa, blandit in, dapibus sit
amet, vestibulum et, augue. Suspendisse ac risus. Duis semper
fringilla sem. Praesent augue arcu, scelerisque nec, ornara
```

malesuada, posuere a, neque. Nullam nulla nisi, ultrices quis, adipiscing senar, varius ut, dui. Nulla viverra pellentesque sem.

```
</p>
</div>
```

```
<div id="lateral">
```

```
<p>
```

Nam erat massa, blandit in, dapibus sit amet, vestibulum et, augue.

Suspendisse ac risus.

```

```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div id="peu">
```

La meua pàgina Web - 2016

```
</div>
```

```
</div>
```


```
</body>
```

```
</html>
```

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

LOREM IPSUM DOLOR

[Opción 1](#)
[Opción 2](#)
[Opción 3](#)
[Opción 4](#)

<p>Curabitur rutrum eros a risus.</p> <p>Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In molestie suscipit libero.</p> <p>Cras sem. Nunc non tellus et urna mattis tempor. Nulla nec tellus a quam hendrerit venenatis.</p> <p>Suspendisse pellentesque odio et est.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p> <p>Maecenas non tortor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed fermentum lorem a velit.</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin sodales, enim in volutpat vehicula, leo turpis vehicula magna, ut rutrum arcu lorem ac pede.</p> <p>Curabitur rutrum eros a risus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In molestie suscipit libero. Cras sem. Nunc non tellus et urna mattis tempor. Nulla nec tellus a quam hendrerit venenatis. Suspendisse pellentesque odio et est. Morbi sed nisl sed dui consequat sodales. Donec porta porta ligula. Nam erat massa, blandit in, dapibus sit amet, vestibulum et, augue. Suspendisse ac risus. Duis semper fringilla sem. Praesent augue arcu, scelerisque nec, ornare malesuada, posuere a, neque. Nullam nulla nisi, ultrices quis, adipiscing non, varius ut, dui. Nulla viverra pellentesque sem.</p>	<p>Nam erat massa, blandit in, dapibus sit amet, vestibulum et, augue. Suspendisse ac risus.</p> 
--	--	--

Mi página Web - 2016

6.5 BONES PRÀCTIQUES

A continuació, s'indiquen algunes recomanacions o bones pràctiques quant a l'ús de les fulles d'estil aplicades al disseny visual de pàgines Web:

- **Inicialitzar els valors de les propietats d'estil.** Una fulla d'estil s'aplica a les pàgines Web a les quals s'ha associat. Quan una pàgina no inclou la inicialització dels valors de les propietats d'estil de manera expressa, l'aspecte amb el qual es mostra en el navegador depèn dels valors predeterminats de cada navegador.
- **Provar en els navegadors més utilitzats.** Una pràctica imprescindible dels dissenyadors Web és provar el seu treball en diversos navegadors diferents. D'aquesta manera, el dissenyador pot descobrir els errors propis de disseny i també les diferències de visualització existents en els diferents navegadors, a causa de la fulla d'estils predeterminada del navegador.
- **Tècniques de depuració d'errors.** Solen cometre's errors en escriure codi CSS per a especificar les declaracions d'estil. En primer lloc, quan no s'està segur sobre si totes les regles CSS estan ben escrites, el millor és comprovar sintàcticament la fulla d'estils utilitzant un validador de CSS, per exemple: <http://jigsaw.w3.org/css-validator/>. Una vegada descartats els errors de sintaxis, el següent problema és detectar per què una regla CSS no s'aplica correctament a un element. Una estratègia útil consisteix a afegir alguna propietat que siga visualment significativa per a comprovar si el selector és el correcte. Per exemple, augmentar molt la seua grandària de lletra, canviar el color del fons, etc.
- **Utilitzar el mètode de disseny de millora progressiva.** La millora progressiva proposa que el disseny web es realitze en dos passos. En primer lloc, el disseny Web ha de permetre l'accés complet i correcte a tota la informació de la pàgina, per la qual cosa el disseny i la composició de la pàgina incorpora només les carauterístiques més bàsiques. I, en segon lloc, després de complir el requisit anterior, es van afegint progressivament les millores i carauterístiques més avançades i específiques del disseny, al mateix temps que es van comprovant els resultats obtinguts en diversos navegadors.
- **Tabular el codi.** Aquesta tècnica d'escriptura de codi facilita significativament la seua lectura, per la qual cosa tabular el codi CCS que especifica les propietats i regles d'estil és una de les pràctiques recomanades pels dissenyadors Web professionals. No és recomanable inserir tabuladors en el codi, sinó que és millor inserir 2 o 4 espais en blanc.
- **Millora del rendiment.** Diverses estratègies, que s'enumeren a continuació, són útils per a millorar el rendiment de l'aplicació de les fulles d'estil en cascada en les pàgines Web: enllaçar fulles d'estils externes en comptes d'incloure els estils en la pròpia pàgina, enllaçar les fulles d'estils mitjançant <link> en comptes de @ import, reduir el nombre d'arxius CSS externs de la pàgina i combinar, si és possible, tots els arxius CSS externs individuals en un únic arxiu.

Glossari de termes

Accessibilitat Web. Un lloc web accessible es carautoritza per ser transformable, comprensible i navegable. El terme transformable s'aplica al fet que la informació i els serveis del lloc web han de ser accessibles per a totes les persones i han de poder ser utilitzats amb tots els dispositius de navegació. El terme comprensible es refereix al fet que el lloc web ha d'incloure continguts clars i simples. El terme navegable significa que el lloc web ha d'incloure mecanismes senzills de navegació.

Disseny Web responsive o adaptatiu. El disseny web adaptatiu, o adaptable, conegut per les sigles RWD de l'anglès, *Responsive Web Design*, és una filosofia de disseny i desenvolupament de llocs web l'objectiu dels quals és adaptar l'aparença de les pàgines Web al dispositiu que s'estiga utilitzant per a visualitzar-les. De manera que la presentació de la informació siga independent del dispositiu i, per tant, la visualització de la informació siga adequada. Hui dia les pàgines Web es presenten en multitud de dispositius com a ordenadors, tauletes, telèfons mòbils, llibres electrònics, ordinadors portàtils, etc. A més, encara dins de cada tipus, cada dispositiu té les seues carautòriques concretes: grandària de pantalla, resolució, potència de CPU, sistema operatiu o capacitat de memòria entre altres. Aquesta tecnologia pretén que, un únic disseny Web s'obtinga una visualització adequada en qualsevol dispositiu. El dissenyador i autor nord-americà Ethan Marcotte va crear i va difondre aquesta tècnica a partir d'una sèrie d'articles en *A List Apart*, que és una publicació en línia especialitzada en disseny i desenvolupament Web. Aquesta idea inicial la va estendre en el seu llibre *Responsive Web Design*, posteriorment. Actualment, existeix una varietat de frameworks de disseny Web que poden ser utilitzats per a establir les carautòriques de la presentació visual de la pàgina incorporant disseny *responsive*. Un dels frameworks més coneguts i empleats actualment és Bootstrap.

Framework. Traduït al castellà com a Entorn de treball o Marco de treball. És un conjunt estandarditzat de conceptes, pràctiques i criteris per a enfocar un tipus de problemàtica particular que serveix com a referència, per a enfrontar i resoldre nous problemes d'índole similar.

Landscape. Orientació horitzontal de pàgines. La pàgina es presenta apaïsada.

Portrait. Orientació vertical de pàgines. La pàgina gira pel costat més llarg.