

## DISSENY D'INTERFÍCIES WEB

## 2. Creació i aplicació d'estils a la interfície web. CSS3

Per crear el contingut i donar estructura i semàntica a una web s'utilitza el llenguatge HTML. Tot i això, no tan sols és important el contingut, sinó que també ho són, i molt, el disseny i la presentació de la informació. És a dir, la utilització del color, de la tipografia, de la imatge i dels elements interactius que fan la interfície agradable per a l'usuari que accedeix al web.

Així, per millorar l'aspecte i el manteniment d'un lloc web és indispensable la utilització d'estils en llenguatge CSS, Cascading Style Sheets. Aquest llenguatge controla la presentació dels documents HTML de tal manera que permet separar la presentació del contingut de la pàgina web i li aporta un format al document en HTML molt més ric en disseny. Actualment, CSS està suportat per tots els navegadors web.

### 2.1. Fulls d'estils: CSS3

Els fulls d'estils en cascada o CSS corresponen a un conjunt de regles usades per definir i crear la presentació d'un document escrit en HTML. L'organisme encarregat de regular la seva especificació és el World Wide Web Consortium (W3C)

Podeu consultar l'estàndard CSS3 a: [www.w3.org/TR/CSS](http://www.w3.org/TR/CSS).

L'estàndard actual de W3C per a l'aplicació d'estils és CSS3, i és totalment compatible amb les seves versions anteriors.



Logotip del CSS3

El llenguatge CSS indica al navegador com s'ha de visualitzar el contingut de la pàgina web.

Els avantatges que ofereix la utilització de fulls d'estil són:

- Separar l'estructura de les pàgines i el seu contingut (en HTML), del format del

## DISSENY D'INTERFÍCIES WEB

- Unificar el disseny de les pàgines web del lloc web, per així definir un estil una sola vegada i poder aplicar-lo tantes vegades com es vulgui.
- Reutilització dels mateixos fulls d'estil per a diferents documents HTML.
- Estàndard de W3C, que ofereix compatibilitat amb versions anteriors i amb els diferents navegadors.

### 2.1.1. Incloure estils CSS en un document HTML

Per crear un document web amb estils CSS es parteix d'un document HTML amb els continguts etiquetats per designar la funció de cada element dins del document, és a dir, els encapçalaments, els paràgrafs, les imatges, els enllaços i altres elements. Seguidament, s'utilitza el llenguatge CSS per definir l'aspecte de cada element HTML, com el color, la mida, el tipus de lletra i la posició.

De fet, els estils es poden crear amb el mateix editor que s'utilitza per crear el document HTML, com pot ser Brackets o Codepen.

A més a més, els estils CSS d'un document web es poden definir en un document de text extern amb l'extensió .css, o bé també es poden definir al mateix document HTML.

Enllaçar un full d'estils extern

Un **full d'estils extern** és un document de text amb extensió .css en el qual hi ha definits els estils d'un document web.

---

### Exemple de contingut d'un document CSS

```
/* Comentari del document CSS */  
  
h1 {  
  color: #FFFFFF;  
  background-color: #999999;  
}  
  
p {  
  font-family: Georgia, Helvetica;  
}
```

---

Per enllaçar un document CSS a una pàgina web s'utilitza l'etiqueta `<link>`, la qual s'especifica en la capçalera, `<head>`, de la pàgina web.

És aconsellable utilitzar un full d'estils extern, ja que permet la seva reutilització i facilita el manteniment del lloc web. Pel fet de tenir els estils centralitzats en un o uns quants documents, qualsevol modificació resulta més senzilla.

---

### Exemple d'enllaç d'un full d'estils extern

## DISSENY D'INTERFÍCIES WEB

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

Els atributs de l'element `link` més rellevants es poden consultar en la [taula 2.1](#).

**Taula: 2.1.** Atributs de l'element `link`

Atribut	Descripció
<code>href</code>	Especifica la ruta del document .css.
<code>rel</code>	Defineix la relació entre els documents .css i .html. Pot ser <code>stylesheet</code> o bé <code>alternate</code> .
<code>type</code>	Indica al navegador el tipus de recurs. En aquest cas: <code>text/css</code> .
<code>title</code>	Especifica un títol al recurs.
<code>media</code>	Especifica els medis en els quals s'aplica en full d'estils.

### Incrustar un full d'estils a HTML

Un full d'estils pot incrustar-se en el document HTML amb l'element `<style>`, el qual es sol definir a la capçalera (`<head>`) del document web.

### Exemple d'estils incrustats

```
<head>
...
<style type="text/css" media="screen">
  body { background: url(foo.gif) red; color: black; }
  p em { background-color: yellow; color: black; }
  .nota { margin-left: 5em; margin-right: 5em; }
</style>
</head>
```

A l'exemple anterior es pot veure com l'element `<style>` també admet atributs com `type` i `media`.

S'aconsella fer servir un **estil incrustat** quan un únic document té un únic estil. Però si els mateixos estils s'utilitzen en diferents pàgines web, llavors seria més apropiat utilitzar un full d'estils extern.

### Importar un full d'estils

Un full d'estils extern pot ser importat amb la regla `@import` de CSS, que es pot especificar tant a l'inici d'un fitxer .css com al principi de l'element `<style>`.

## DISSENY D'INTERFÍCIES WEB

### Exemple d'importació de fulls d'estils

---

```
<style type="text/css" media="screen, projection">
  @import url(http://elmeudomini.com/css/estils.css);
  @import url(/css/estils2.css);
  p { background-color: yellow; color: black; }
</style>
```

---

#### Estils en línia

Per especificar estils en línia s'utilitza l'atribut `style`, que pren per valor qualsevol nombre de propietats CSS separades per ”;”.

---

### Exemple d'estils en línia

---

```
<p style="color: red; font-family: 'New Century Schoolbook', serif">
Aquest paràgraf mostra el text vermell i amb la font New Century Sci
</p>
```

---

Els estils en línia perden molts dels avantatges dels fulls d'estil en barrejar el contingut amb la presentació. A més a més, els estils en línia s'apliquen a tots els tipus dispositius de sortida (ordinadors, tauletes, mòbils...), la qual cosa provoca que es desquadi un disseny adaptatiu. Així doncs, és per aquests motius que s'haurien d'usar amb moderació.

#### 2.1.2. Característiques de CSS: cascada i herència

Dues de les característiques que fan que els fulls d'estil tinguin un ampli ventall de possibilitats són la cascada i l'herència. La **cascada** fa referència a la possible combinació de diferents fulls d'estil, mentre que l'**herència** es refereix a la capacitat que tenen els elements del document HTML d'heretar propietats dels seus elements antecessors.

##### Fulls d'estil en cascada

A l'hora de visualitzar una pàgina web, el navegador ha d'interpretar els diferents estils definits per a cada element HTML. Aquests poden estar especificats en diferents llocs:

- Estils predeterminats del navegador.
- Estils especificats per l'usuari final.
- Estils relacionats amb el document especificats pel seu autor. Aquests es poden definir en tres llocs:
  - En un fitxer extern.
  - Al principi del document, a través de l'element `<style>`.

## DISSENY D'INTERFÍCIES WEB

Així, hi ha tres fonts principals d'informació d'estils formant una cascada, de tal manera que els estils de l'usuari modifiquen o són prioritaris als que defineixi per defecte el navegador, i els estils de l'autor del document seran prioritaris als anteriors.

De la mateixa manera, si la definició d'un estil d'un autor entra en conflicte amb la definició d'un altre, s'aplicarà l'últim que estigui definit. És a dir, un estil definit a través de l'atribut `style` serà prioritari respecte als definits a través de l'element `<style>`.

---

### Exemple d'aplicació d'estils en cascada

```
<!doctype html>
<html>
<head>
...
<style>
  p {
    color: red;
  }
</style>
</head>
<body>
  <h1>Encapçalament</h1>
  <p style="color: blue">Un paràgraf</p>
  <p>Un altre paràgraf</p>
</body>
</html>
```

En podeu veure el resultat en la [figura 2.1](#).

**Figura 2.1.** Figura d'estil en cascada



Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-mo9/pen/pyrpER](https://codepen.io/ioc-daw-mo9/pen/pyrpER).

Si us hi fixeu, el text del primer paràgraf és blau, tal com s'indica a l'atribut `style`, enlloc de vermell definit a l'estil per als paràgrafs al `<head>` de la pàgina. Com veieu, doncs, el segon paràgraf ja apareix de color vermell.

L'herència als fulls d'estil

Atès que tots els elements d'una pàgina `HTML`, amb excepció de l'element arrel `<html>`, estan continguts en un altre element, s'ha de tenir en compte que tot ele-

## DISSENY D'INTERFÍCIES WEB

Malgrat tot, s'han de tenir en compte les següents consideracions:

- No totes les propietats s'hereten. Aquesta característica es pot consultar a: [www.w3.org/TR/css-2010/#properties](http://www.w3.org/TR/css-2010/#properties).
- Si es vol forçar l'herència d'una propietat d'un element que per defecte no hereta es pot utilitzar el valor `inherit`.
- Si s'especifica un valor a una propietat, aquest valor preval sobre el valor heretat.

---

### Exemple d'utilització de l'herència del llenguatge CSS

```
<!doctype html>
<html>
<head>
...
<style>
body {
  color: gray;
}

p {
  font-size: 15px;
}

strong {
  font-style: italic;
  font-size: 20px;
}

div {
  border-style: solid;
  border-width: 1px;
  border-color: red;
  background-color: rgba(255, 0, 0, 0.2);
  margin: 5px;
  width: 300px;
  padding: 5px;
}
</style>
</head>
<body>
<p>Un paràgraf amb un <strong>text molt emfatitzat</strong></p>

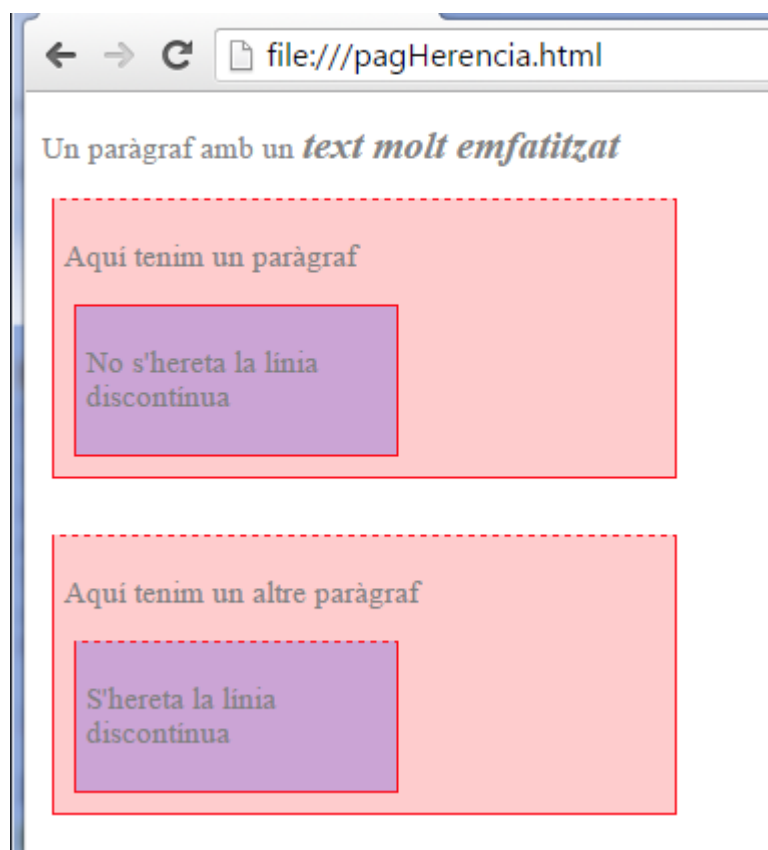
<div style="border-top-style: dashed;">
  <p> Aquí tenim un paràgraf</p>
  <div style="background-color:rgba(0,0,255,0.2);width:50%;">
    <p>No s'hereta la línia discontinua</p>
  </div>
</div>
<br />
<div style="border-top-style: dashed;">
  <p>Aquí tenim un altre paràgraf</p>
  <div style="border-top-style: inherit;background-color:rgba(0,0,255,0.2)">
    <p>S'hereta la línia discontinua</p>
  </div>
</div>
</body>
</html>
```

---

Ho podeu veure en la [figura 2.2](#).

**Figura 2.2.** Herència a CSS

## DISSENY D'INTERFÍCIES WEB



Podem veure el codi en el següent enllaç: [codepen.io/ioc-daw-m09/pen/yOopMb](https://codepen.io/ioc-daw-m09/pen/yOopMb).

Fixeu-vos com el color del text és gris, ja que el paràgraf, en estar contingut dins de l'element `<body>`, hereta els seus estils. Alhora, l'element `<strong>` hereta els estils dels elements `<p>` i `<body>`.

Fixeu-vos també com en la primera caixa blava que es troba a dins de la vermella no hereta l'atribut `border-top-style`, i com a la segona caixa blava s'ha forçat l'herència de la vermella, que la conté amb el valor `inherit`.

### 2.1.3. Format d'una regla CSS

Per definir un estil específic, CSS fa servir una regla que consisteix en un selector i un bloc on es declaren les diferents propietats que ha de tenir l'estil.

**selector { propietat: valor; propietat: valor; ... }**

On *selector* fa referència a l'element que s'aplica l'estil.

On les diferents propietats es declaren en el bloc separades per “;”.

### Exemple de definició d'un estil

## DISSENY D'INTERFÍCIES WEB

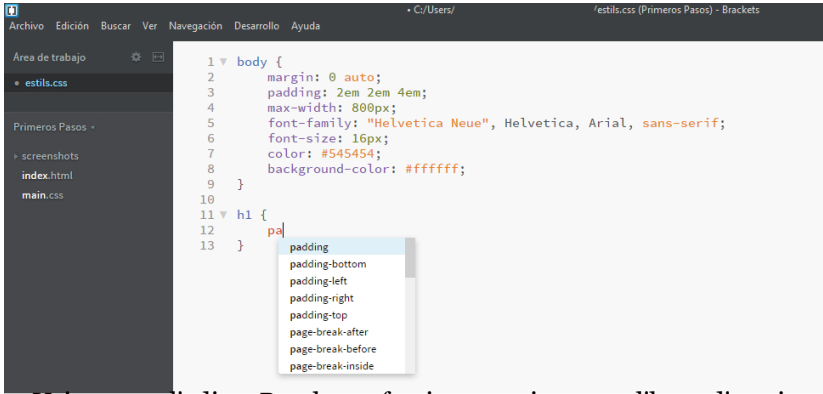
L'estil definit en l'exemple anterior estableix que les capçaleres h1 tindran el següent aspecte:

- La mida de la font serà de 10 píxels.
- El color de la font serà blau.
- El text es veurà centrat.

Per definir els estils es pot utilitzar qualsevol editor de text, per exemple el mateix que s'empri per crear els documents HTML, com poden ser Brackets, Codepen, Sublime, etc.

Vegeu en la [figura 2.3](#) com es poden escriure els estils en l'editor Brackets.

**Figura 2.3.** Creació d'estils amb l'editor Brackets



Veiem com l'editor Brackets ofereix suggeriments a l'hora d'escriure una propietat CSS.

### 2.1.4. Selectors

#### Selectors W3C

Podeu consultar els diferents tipus de selectors a: [goo.gl/9mY7Fw](https://goo.gl/9mY7Fw).

La funció dels selectors és indicar a quins elements s'aplicaran els estils definits. Els diferents tipus de selectors es presenten en la [taula 2.2](#).

**Taula: 2.2.** Selectors bàsics

Sintaxi	Descripció
*	Selector universal. Permet aplicar un estil a qualsevol tipus.
nomElement	Selector de tipus. Permet aplicar un estil a un element <u>HTML</u> concret.
.nomClasse	Selector de classe. Permet aplicar un estil als elements que utilitzin aquesta classe.
#idNom	Selector ID. Permet aplicar un estil a un element que tingui un codi d'identificació únic.



## DISSENY D'INTERFÍCIES WEB

Així, per exemple, es poden tenir els següents estils definits en un full d'estils:

---

### Exemples d'estils amb diferents selectors

---

```
* {
  font-size: 15px;
  font-family: Arial;
}

h1.ressaltat {
  font-weight: bolder;
  font-size: 25px;
  color: white;
  background-color: #ff80c0;
}

p,h2 {
  text-align:center;
}

.textBlau {
  color: blue;
}

#estil1 {
  font-size: larger
}

a[target] {
  background-color: yellow;
}
```

---

---

### Utilització dels estils anteriors

---

```
<h1 class="ressaltat">Títol ressaltat</h1>

<h2> Els subtítols apareixen centrats</h2>

<p> Els paràgrafs apareixen centrats igual que els subtítols.</p>

<p class="textBlau">Aquest paràgraf és blau. </p>

<p id="estil1">Aquest estil només s'utilitza una vegada en el docume

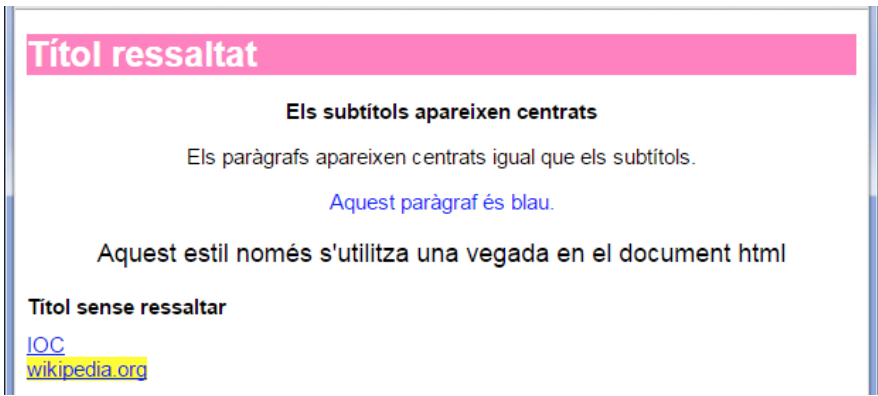
<h1>Títol sense ressaltar</h1>
<a href="http://ioc.xtex.cat">IOC</a>
<br/>
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>
```

---

El resultat es pot veure en la [figura 2.4](#).

**Figura 2.4.** Estils selectors

DISSENY D'INTERFÍCIES WEB



Podeu provar aquest exemple en l'enllaç següent: [codepen.io/ioc-daw-mo9/pen/pyrBaP](https://codepen.io/ioc-daw-mo9/pen/pyrBaP).

Algunes consideracions de l'exemple anterior són:

- L'atribut `class` s'utilitza per aplicar un estil de classe a l'element.
- L'atribut `id` s'utilitza per aplicar un estil `id`, únic per a un element.
- Per defecte, el text de tots els elements serà Arial de 15 px.
- Els *links* que tinguin l'atribut `target` apareixen ressaltats.
- Si es vol definir el mateix estil per a diferents selectors es poden agrupar separats per una coma.

En la [taula 2.3](#) es poden consultar els diferents tipus de selectors d'atribut, ja que en aquests s'hi poden especificar quins valors haurien de tenir.

Taula: 2.3. Selectors d'atribut

Sintaxi	Descripció
[ atribut ]	Selecciona l'element amb aquest atribut.
[ atribut="valor" ]	El valor de l'atribut és valor.
[ atribut^="text" ]	El valor de l'atribut comença per text.
[ atribut\$="text" ]	El valor de l'atribut acaba amb text.
[ atribut*="text" ]	El valor de l'atribut conté el text.
[ atribut~="valor" ]	El valor de l'atribut és una llista de paraules separades per espais, una de les quals és exactament valor.
[ atribut ="valor" ]	El valor de l'atribut és una llista de paraules separades per guions, començant per valor.

Selecciona l'element amb aquest atribut.

Combinacions de selectors

Els selectors es poden combinar per afegir especificacions a les regles `CSS`. En la [taula 2.4](#) es poden visualitzar algunes de les possibles combinacions dels selectors.

Taula: 2.4. Combinacions de selectors

Sin-	Descripció
------	------------

## DISSENY D'INTERFÍCIES WEB

element específic (A).

- A ~ B** Selectors general de germans. Permet aplicar un estil a l'element (B) que sigui germà de l'element (A).
  - A > B** Selectors de fills. Permet aplicar un estil al primer element fill (B) d'un element específic (A).
  - A B** Selectors descendents. Permet aplicar un estil a un element determinat que es trobi a dins d'un element específic.
- 

### Exemple de definició d'estils combinats.

```
/* selector general de germans */
h1 ~ h2 {
  color: red;
}

/* selector de germans adjacents */
h1 + h2 {
  color: blue;
}

/* selector de descendents*/
div h2 {
  color: green;
}

/*selector de fills*/
div > p {
  color: orange;
  font-weight: bold;
}
```

---

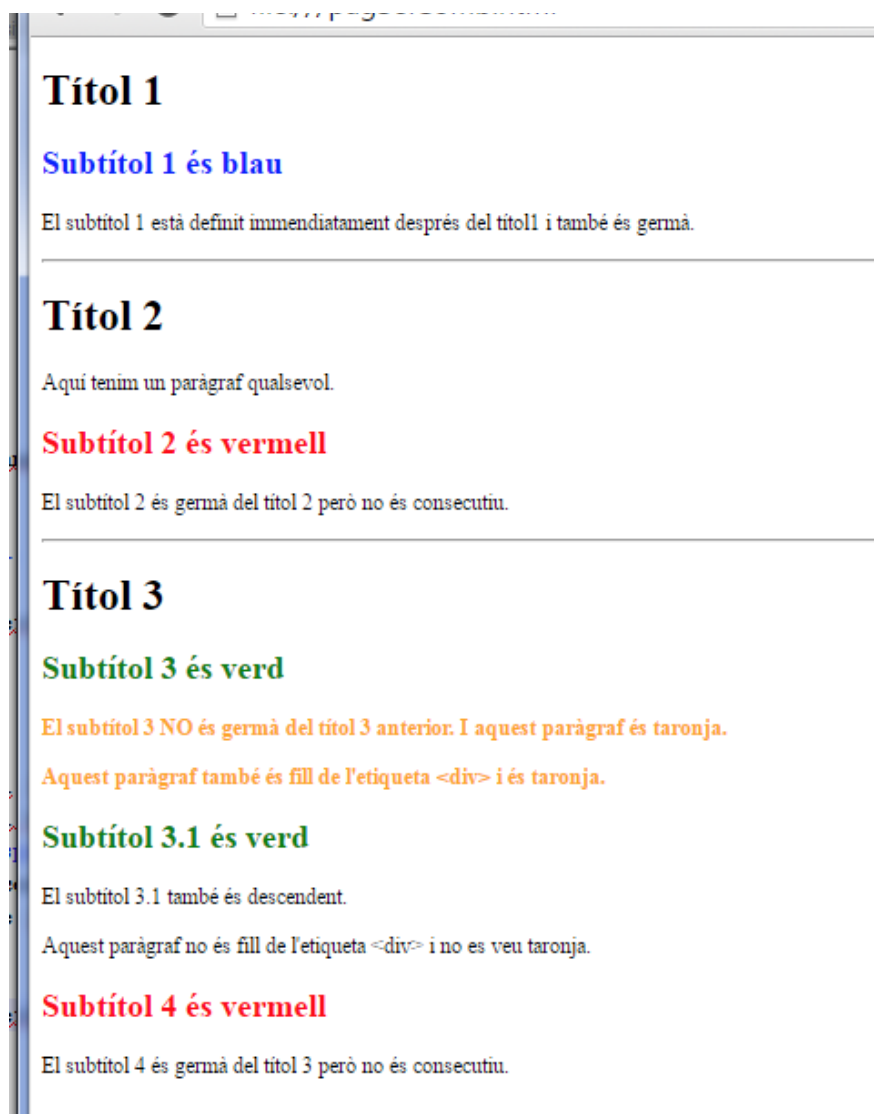
### Exemple d'utilització dels estils anteriors:

```
<h1>Títol 1</h1>
<h2>Subtítol 1 és blau</h2>
<p>El subtítol 1 està definit immediatament després del títol1 i tan</p>
<hr/>
<h1>Títol 2</h1>
<p>Aquí tenim un paràgraf qualsevol.</p>
<h2>Subtítol 2 és vermell</h2>
<p>El subtítol 2 és germà del títol 2 però no és consecutiu.</p>
<hr/>
<h1>Títol 3</h1>
<div>
  <h2>Subtítol 3 és verd</h2>
  <p>El subtítol 3 NO és germà del títol 3 anterior. I aquest paràgr<
  <p> Aquest paràgraf també és fill de l'etiqueta <div> i és f<
  <span><h2>Subtítol 3.1 és verd</h2></span>
  <span><p>El subtítol 3.1 també és descendent.</p></span>
  <span><p>Aquest paràgraf no és fill de l'etiqueta <div> i no<
</div>
<h2>Subtítol 4 és vermell</h2>
<p>El subtítol 4 és germà del títol 3 però no és consecutiu.</p>
```

---

**Figura 2.5.** Combinació de selectors

## DISSENY D'INTERFÍCIES WEB



Podeu provar aquest exemple en l'enllaç: [codepen.io/ioc-daw-mo9/pen/BKdeKW](https://codepen.io/ioc-daw-mo9/pen/BKdeKW).

Fixeu-vos en els següents punts:

- El primer subtítol h2 és germà adjacent de h1 (immediatament després del títol h1) i, per tant, es veu de color blau enlloc de vermell. En canvi, el subtítol 2 es veu vermell perquè no està definit immediatament després del títol h1.
- Els subtítols h2 descendents de h1 apareixen de color verd, tal com es veu en el subtítol 3 i 3.1.
- Els paràgrafs que siguin fills directes de l'element `div` es veuen de color taronja.

### Pseudoclasses

Les pseudoclasses s'afegeixen al selector de manera que es pugui aplicar un format determinat a l'element seleccionat.

La manera de definir estils utilitzant pseudoclasses és:

## DISSENY D'INTERFÍCIES WEB

```
    ...  
}
```

En la [taula 2.5](#) s'especifiquen les pseudoclasses per als enllaços.

**Taula: 2.5.** Pseudoclasses d'acció d'usuari i d'enllaç

Pseudo-classe	Descripció
:link	Permet definir l'estil dels enllaços quan encara no s'han visitat.
:visited	Permet definir l'estil dels enllaços visitats.
:active	Permet definir l'estil dels elements quan s'activen (quan es prem el ratolí damunt seu).
:hover	Permet definir l'estil dels elements quan es passa el ratolí damunt seu).
:focus	Permet definir l'estil dels elements quan reben el focus.

### Exemple d'estils per als enllaços

```
/* Els enllaços (visitats i no visitats) són de color blau amb el f  
a:link,  
a:visited {  
    color: blue;  
    background-color: gray;  
    text-decoration: none;  
}  
  
/* Quan es passa per damunt tindran el fons blau i text blanc */  
a:hover {  
    color: white;  
    background-color: blue;  
}  
  
/* Quan es fa clic abans de soltar es faran més grans */  
a:active {  
    font-size: larger;  
    color: white;  
    background-color: blue;  
}  
  
/* Quan se situa en un control de formulari <input> tindrà  
una vora vermella puntejada, i un farciment de 2px */  
input:focus {  
    border: red 2px dotted;  
    padding: 2px;  
}
```

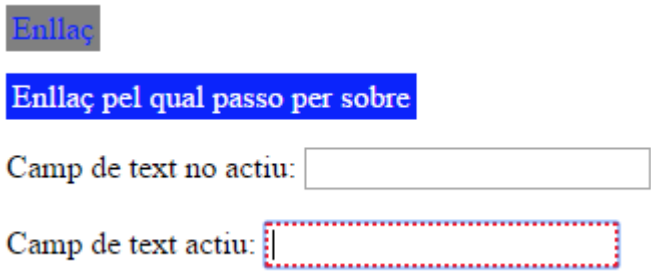
### Utilització d'estils per als enllaços

```
<p><a href="http://fsf.org">Enllaç</a></p>  
  
<p><a href="http://gnu.org">Enllaç pel qual passo per sobre</a></p>  
  
<p>Camp de text no actiu: <input type="text" name="input" /></p>
```

## DISSENY D'INTERFÍCIES WEB

El resultat que s'obté es pot veure en la [figura 2.6](#).

**Figura 2.6.** Estils dels enllaços



El DOM (Document Object Model) representa el document creat a partir de les etiquetes HTML i els estils CSS que el navegador mostrarà per la pantalla.

Podeu provar aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/MyEWvW](https://codepen.io/ioc-daw-m09/pen/MyEWvW).

Existeixen altres pseudoclasses que apliquen els estils fent referència a l'estructura del DOM. Vegeu-les en la [taula 2.6](#).

**Taula: 2.6.** Pseudoclasses estructurals

Pseudoclasse	Descripció
:root	Representa un element que es troba a l'arrel del document.
:nth-child (num)	Selecciona l'element indicat però amb la condició que sigui el fill enèsim del seu pare.
:nth-last-child (num)	Idèntic a l'anterior, però el número indicat es comença a comptar des de l'últim fill.
:nth-of-type (num)	Selecciona l'element indicat però amb la condició que sigui l'enèsim element fill d'aquest tipus.
:nth-last-of-type (num)	Idèntic a l'anterior, però el nombre indicat es comença a comptar des de l'últim fill.
:first-child	Selecciona un element que sigui el primer fill d'un altre element.
:last-child	Selecciona l'element indicat, però amb la condició que siguin l'últim fill del seu element pare.
:first-of-type	Fa referència al primer element d'aquest tipus en l'element pare.
:last-of-type	Fa referència a l'últim element d'aquest tipus en l'element pare.
:only-of-type	Fa referència a aquell element que és l'únic fill del seu pare.
:empty	Selecciona l'element indicat però amb la condició que no tingui cap fill i tampoc pot tenir cap contingut de text.

En la [taula 2.7](#) hi trobareu altres pseudoclasses.

**Taula: 2.7.** Altres pseudoclasses

## DISSENY D'INTERFÍCIES WEB

:lang	D'idioma. Permet aplicar estils en funció de l'estil especificat.
:not	De negació. Selecciona tots els elements que no compleixen amb la condició d'un selector.

---

---

### Exemple d'estils fent referència a l'estructura DOM

```
/*Selecciona els elements li descendents de les llistes de tipus es
ul.especial li {
  display: inline;
  padding: 5px;
}

/* selecciona el primer fill de la llista*/
ul.especial li:first-child {
  color: blue;
}

/*selecciona els elements imparells de la llista */
ul.especial li:nth-child(odd) {
  background-color: grey;
}

/*selecciona els elements imparells de la llista */
ul.especial li:nth-child(even) {
  background-color: pink;
}

/*selecciona el quart fill de la llista*/
ul.especial li:nth-of-type(4) {
  color: white;
}

/*selecciona els paràgrafs que no tenen germans paràgrafs*/
p:only-of-type {
  background-color: grey;
  color: #ffa0ff;
  font-weight: bold;
  font-size: 25px;
}

/*Els paràgrafs en anglès apareixen en itàlic */
p:lang(en){
  font-style:italic;
}

:root {
  font-family:arial;
}
```

---

---

### Exemple d'utilització d'estils que fan referència a l'estructura DOM

```
<div>
  <p>Menú especial:</p>

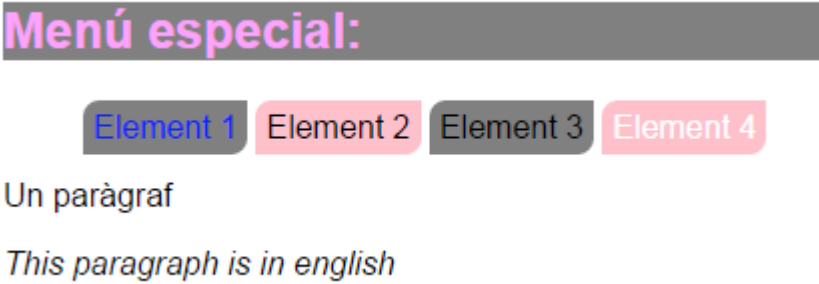
  <ul class="especial">
    <li> Element 1</li>
    <li> Element 2</li>
    <li> Element 3</li>
    <li> Element 4</li>
  </ul>

  <div>
    <p>Un paràgraf</p>
    <p lang="en">This paragraph is in english.</p>
  </div>
```

## DISSENY D'INTERFÍCIES WEB

El resultat es pot veure en la [figura 2.7](#).

**Figura 2.7.** Visualització de l'aplicació d'estils d'estructura DOM



Podeu provar l'anterior exemple en l'enllaç següent: [codepen.io/ioc-daw-mo9/pen/RajWyj](https://codepen.io/ioc-daw-mo9/pen/RajWyj).

Podeu consultar les pseudoclasses que defineix CSS a: [developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes](https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes).

### Pseudoelements

A diferència de les pseudoclasses, els pseudoelements no descriuen l'estat d'un element, sinó que s'afegeixen a un selector per tal de definir estils a una part concreta del document HTML.

Els pseudoelements solen anar precedits per ":", però per compatibilitat amb versions anteriors també poden anar precedits per ":".

La manera de definir estils utilitzant pseudoelements és:

```
selector::pseudoElement{
  propietat: valor;
  ...
}
```

Podem consultar els pseudoelements que defineix CSS a [developer.mozilla.org/es/docs/Web/CSS/Pseudoelements](https://developer.mozilla.org/es/docs/Web/CSS/Pseudoelements). En la [taula 2.8](#) podem veure la descripció d'alguns d'ells.

**Taula: 2.8.** Pseudoelements

Sintaxi	Descripció
::after	Permet introduir contingut al final de l'element. Cal afegir la propietat content amb el valor desitjat.
::before	Permet introduir contingut a l'inici de l'element. Cal afegir la propietat content amb el valor desitjat.



## DISSENY D'INTERFÍCIES WEB

<code>::first-line</code>	Permet definir l'estil de la primera línia de l'element.
<code>::selection</code>	Permet definir l'estil d'allò que l'usuari ha seleccionat amb el ratolí.

A continuació, podeu veure un parell d'exemples:

### Exemple d'estils CSS utilitzant pseudoelements

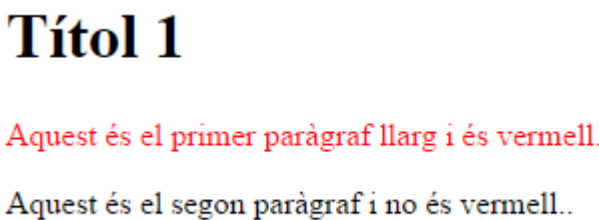
```
div::first-line { color: red; }  
h1::before { content: "Títol "; }  
p::after { content: "."; }  
p::first-letter { text-transform: uppercase; }
```

### Exemple d'utilització d'estils utilitzant pseudoelements

```
<h1>1</h1>  
<div>  
  <p>aquest és el primer paràgraf llarg i és vermell</p>  
  <p>aquest és el segon paràgraf i no és vermell.</p>  
</div>
```

El resultat que obtindreu és el que es pot veure en la [figura 2.8](#).

**Figura 2.8.** Visualització de l'aplicació d'estils utilitzant pseudoelements



**Títol 1**

Aquest és el primer paràgraf llarg i és vermell.

Aquest és el segon paràgraf i no és vermell..

Aquest exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-mo9/pen/YqExdd?editors=1100](https://codepen.io/ioc-daw-mo9/pen/YqExdd?editors=1100).

### 2.1.5. Unitats de mesura CSS

Algunes propietats CSS indiquen la mida de la lletra, l'amplada, els marges, etc. CSS classifica les unitats de mesura en dos tipus: **absolutes** i **relatives**.

En la [taula 2.9](#) es poden visualitzar les unitats de mesura absolutes.

**Taula: 2.9.** Unitats de mesura absolutes

Unitat	Descripció
--------	------------

## DISSENY D'INTERFÍCIES WEB

<b>in</b>	Polzades, <i>inches</i> . Equival a 2,54 cm
<b>pt</b>	Punts. Mesura que equival a 1/72 d'una polzada
<b>pc</b>	Piques. Mesura que equival a 12 pt

En la [taula 2.10](#) es poden visualitzar les unitats de mesura relatives.

**Taula: 2.10.** Unitats de mesura relatives

Unitat	Descripció
<b>em</b>	Mesura relativa a la mida del tipus de lletra del seu contenidor.
<b>rem</b>	<i>Root em</i> . Mesura relativa a la mida del tipus de lletra general. Té com a referència la unitat que fa servir l'element arrel ( <i>root</i> ), i no l'element contenidor.
<b>px</b>	Píxels. Mesura relativa a la resolució de la pantalla.
<b>ex</b>	Mesura relativa a l'alçada de la lletra x. Canvia si es canvia la font que fem servir.
<b>ch</b>	Mesura relativa a l'alçada del caràcter numèric "o". Canvia si es canvia la font que fem servir.
<b>%</b>	Percentatge.

També es disposa d'unitats relatives al *viewport*, és a dir, a l'amplitud de la finestra del navegador. En la [taula 2.11.18](#) es veuen aquestes propietats.

**Taula: 2.11.** Unitats de mesura relatives al 'viewport'

Unitat	Descripció
<b>vw</b>	<i>Viewport width</i> . 1 vw equival a (1/100) de l'amplada de la finestra, és a dir, a l'1%. Per reflectir un valor del 15% hauríem de posar 15 vw.
<b>vh</b>	<i>Viewport height</i> . 1 vw equival a (1/100) de l'alçada de la finestra, és a dir, a l'1%. Per reflectir un valor del 15% hauríem de posar 15 vw.
<b>vmin</b>	Funcionament similar als anteriors, però hi ha una avaluació prèvia de quin eix, <i>x</i> o <i>y</i> (amplada, alçada), és menor, i selecciona com a referent l'eix que ho sigui.
<b>vmax</b>	Idèntic a l'anterior, però agafant com a referent l'eix que sigui més gran dels dos.

A continuació podeu veure alguns exemple:

### Estils amb diferents tipus d'unitats de mesura

```
.mida1 {font-size: 16pt; }  
.mida2 {font-size: 16px; }  
.mida3 {font-size: 1.5em; }  
.mida4 {font-size: 1.5rem; }  
.mida5 {font-size: 1.5vw; }
```

## DISSENY D'INTERFÍCIES WEB

### Utilització d'estils amb diferents tipus d'unitats de mesura

---

```
<p class="mida1">Mida del text en punts (absoluta)</p>
<p class="mida2">Mida del text en píxels (relatiu a la resolució)</p>
<p class="mida3">Mida del text en em (relatiu al contenidor) </p>
<p class="mida4">paràgraf mida en rem (relatiu a l'element root)</p>
<p class="mida5">Mida del text en vw (relativa al viewport)</p>
<hr />
<div class="mida3">
  <p class="mida1">Mida del text en punts (absoluta)</p>
  <p class="mida2">Mida del text en píxels (relatiu a la resolució)</p>
  <p class="mida3">Mida del text en em (relatiu al contenidor) </p>
  <p class="mida4">Mida del text en rem (relatiu a l'element root)</p>
  <p class="mida5">Mida del text en vw (relativa al viewport)</p>
</div>
```

---

El resultat que obtindreu es pot veure en la [figura 2.9](#).

**Figura 2.9.** Visualització de l'aplicació d'estils amb diferents tipus de mesura

Mida del text en punts (absoluta)

Mida del text en píxels (relatiu a la resolució)

Mida del text en em (relatiu al contenidor)

paràgraf mida en rem (relatiu a l'element root)

Mida del text en vw (relativa al viewport)

---

Mida del text en punts (absoluta)

Mida del text en píxels (relatiu a la resolució)

### Mida del text en em (relatiu al contenidor)

Mida del text en rem (relatiu a l'element root)

Mida del text en vw (relativa al viewport)

---

L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-mo9/pen/KzyXdW](https://codepen.io/ioc-daw-mo9/pen/KzyXdW).

Encara que sovint s'utilitza la unitat de mesura píxel, una bona opció és “rem”, ja que especifica les mides relatives a l'element *root*, a diferència de “em”, la mida del qual pot costar de controlar perquè pot variar en funció d'on es trobi.

També podeu comprovar que si modifiquem la mida de la finestra del navegador, el text expressat en vw es redimensiona.

#### 2.1.6. Propietats CSS

Hi ha múltiples atributs o propietats que es poden definir amb CSS. Per conèixer l'ampli ventall que ofereix CSS es poden consultar les següents webs:

- Developer Mozilla: [developer.mozilla.org/es/docs/Web/CSS/Referencia\\_CSS](https://developer.mozilla.org/es/docs/Web/CSS/Referencia_CSS)
- W3C: [www.w3.org/community/webed/wiki/CSS/Properties](https://www.w3.org/community/webed/wiki/CSS/Properties)

## DISSENY D'INTERFÍCIES WEB

fix. Aquests prefixos es faran servir quan així sigui necessari i es poden veure en la [taula 2.12](#).

**Taula: 2.12.** Prefixos per als navegadors

Prefix	Navegador
-moz-	Prefix per al navegador Firefox.
-ms-	Prefix per al navegador Internet Explorer.
-webkit-	Prefix per als navegadors Chrome i Safari.
-o-	Prefix per al navegador Opera.

Tot i això, cal mencionar que no tots els navegadors es comporten de la mateixa manera davant del mateix full d'estil, i això es deu al fet que alguns no compleixen els estàndards establerts. Per això es pot recórrer a pàgines com [caniuse.com](http://caniuse.com) per veure si determinades característiques ja han estat implementades, i per quins navegadors.

### 'Shorthand properties'

Les propietats abreujades permeten establir els valors de diverses propietats CSS simultàniament. Aquestes propietats poden fer referència als marges, les vores, els farciments i també al fons del document web.

En la [taula 2.13](#) es pot consultar un recull de les propietats referents als continguts de la pàgina web.

**Taula: 2.13.** Propietats per a la font, els paràgrafs, els fons i les taules

	Propietats
<b>Font</b>	color, font-size, font-family, font-weight, font-style
<b>Paràgrafs</b>	line-height, text-decoration, text-align, text-indent, text-transform, text-shadow, text-overflow, text-wrap, list-style
<b>Fons</b>	background-color, background, background-image, background-origin, background-repeat, background-position, background-attachment
<b>Taules</b>	border-spacing, border-collapse, caption-side, empty-cells

## Estils amb CSS

```
/*Gradient de 180 graus que va de blau a blanc*/
.gradient1 {
  background: linear-gradient(180deg, blue, white);
}

/*Gradient de 90 graus que va de granat a taronja a taronja més cla:
```

## DISSENY D'INTERFÍCIES WEB

```
/*Gradient radial circular que va de negre a blanc */
.gradient3 {
  background: radial-gradient(circle, black, white);
}

/*Gradient el·líptic que va de negre a blanc*/
.gradient4 {
  background: radial-gradient(ellipse, black, white);
}

/*Els paràgrafs apareixen indentats i tenen una ombra d'un píxel, s'
p {
  text-indent: 50px;
  text-shadow: 1px 2px #ccc;
  padding: 50px;
}

/* Els títols tenen un farciment de 5px*/
h1{
  padding:5px;
}
```

### Utilització d'estils CSS

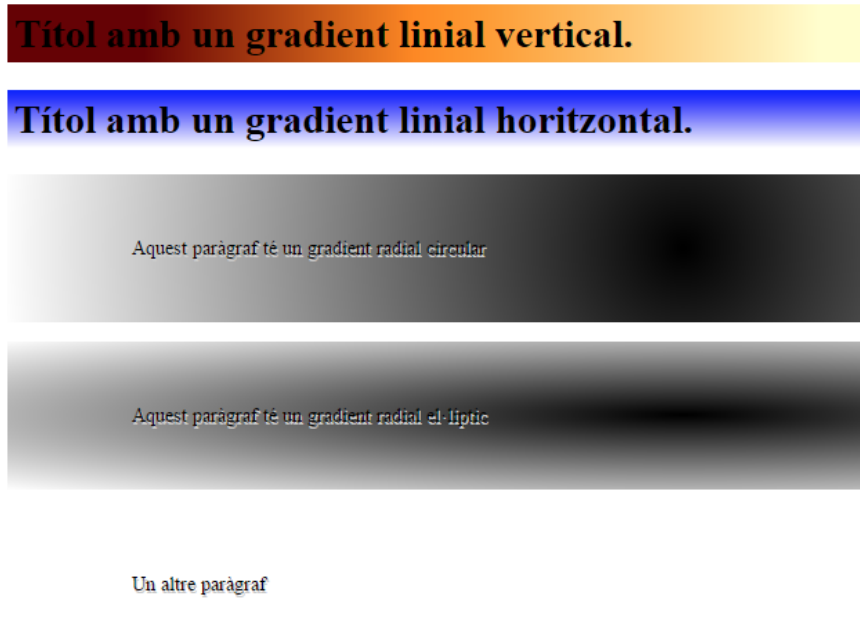
```
<h1 class="gradient2">Títol amb un gradient lineal vertical.</h1>
<h1 class="gradient1">Títol amb un gradient lineal horitzontal.</h1>

<p class="gradient3">Aquest paràgraf té un gradient radial circular.

<p class="gradient4">Aquest paràgraf té un gradient radial el·líptic
<p>Un altre paràgraf</p>
```

El resultat que obtindreu és el que es pot veure en la [figura 2.10](#).

**Figura 2.10.** Visualització d'estils de gradient



Podeu provar l'exemple anterior en l'enllaç següent: [codepen.io/ioc-daw-mo9/pen/NNwaoa?editors=1100](https://codepen.io/ioc-daw-mo9/pen/NNwaoa?editors=1100).

A més a més, cal tenir en compte que a pesar que els navegadors presenten algunes configuracions similars, com ara el tipus de lletra serif de color negre, per exemple.

## DISSENY D'INTERFÍCIES WEB

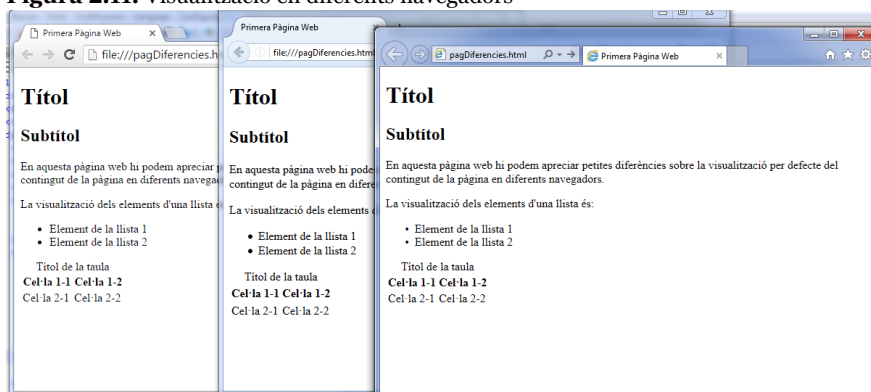
left o padding-left) o bé la interliniat (line-height).

### Exemple de visualització d'una pàgina web en diferents navegadors

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Primera Pàgina Web</title>
</head>
<body>
  <h1>Títol</h1>
  <h2>Subtítol</h2>
  <p>En aquesta pàgina web hi podem apreciar petites diferències
  <p>La visualització dels elements d'una llista és:
  <ul>
    <li>Element de la llista 1</li>
    <li>Element de la llista 2</li>
  </ul>
  <table>
    <caption>Títol de la taula</caption>
    <tr>
      <th>Cel·la 1-1</th>
      <th>Cel·la 1-2</th>
    </tr>
    <tr>
      <td>Cel·la 2-1</td>
      <td>Cel·la 2-2</td>
    </tr>
  </table>
</body>
</html>
```

En la [figura 2.11](#), d'esquerra a dreta, es poden veure les diferències de visualització de la mateixa pàgina web en els navegadors Chrome, Firefox i Internet Explorer.

**Figura 2.11.** Visualització en diferents navegadors



#### 2.1.7. El model caixa

##### Element "<div>"

Aquest element crea una caixa que per defecte ocupa tota l'amplada del navegador. S'utilitza per definir l'estil d'una secció d'una pàgina web.

## DISSENY D'INTERFÍCIES WEB

Aquest element crea una caixa que per defecte només ocupa l'amplada necessària. S'utilitza per definir l'estil d'una porció de contingut d'una secció d'una pàgina web.

Els navegadors, de manera automàtica, creen i col·loquen cada element HTML en una estructura que té format de caixa. Així, per defecte, la caixa dels elements HTML de tipus *block* ocupa tota l'amplada de la finestra, i la dels elements *inline* només agafen l'amplada necessària.

En principi, la caixa de cada element no és visible, ja que no mostra cap color de fons ni vora. No obstant això, es pot modificar amb CSS per canviar les seves dimensions, els colors, la vora i la posició.

Es poden modificar les propietats de la caixa de qualsevol element HTML, però generalment s'utilitza l'element `<div>` per organitzar i donar estructura al disseny de les pàgines web.

### Dimensions de la caixa

Les parts que componen una caixa en el seu ordre de visualització des del punt de vista de l'usuari són:

- **Contingut**, *content*: es referix al contingut de l'element; per exemple, el text, una imatge, etc.
- **Farciment**, *padding*: espai lliure opcional existent entre el contingut i la vora.
- **Vora**, *border*: línia que envolta el contingut i el farciment.
- **Imatge de fons**, *background image*: imatge que es mostra per darrere del contingut i el farciment.
- **Color de fons**, *background color*: color que es mostra darrere del contingut i el farciment.
- **Marge**, *margin*: separació opcional existent entre la caixa i la resta de caixes adjacents.

El farciment i el marge que defineix el model de caixa són transparents, per tant, es visualitza la imatge o el color de fons que s'hi hagi definit.

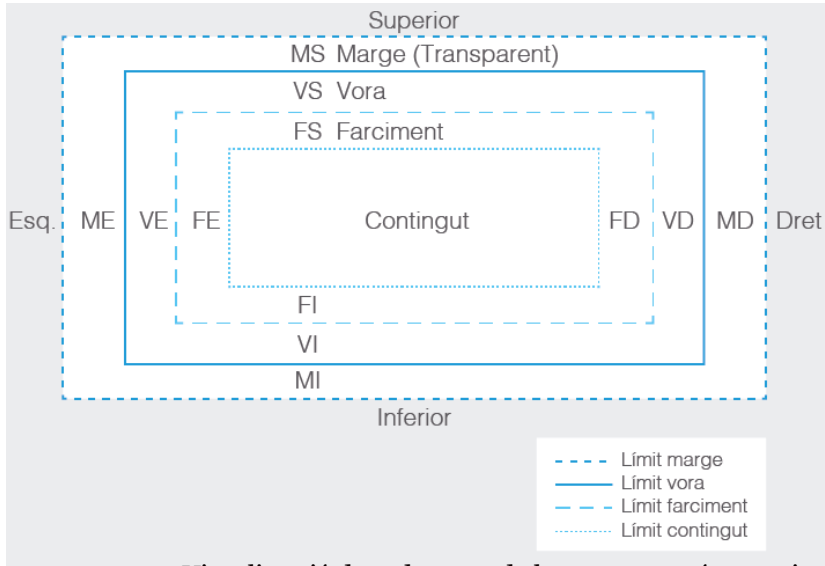
### Imatge i color de fons del model de caixa

Si en una caixa es defineixen el color i la imatge de fons, la imatge té més prioritat i es visualitza. Però si la imatge no cobreix totalment la caixa de l'element també es visualitza el color de fons.

Així doncs, cada caixa té una àrea de contingut, i opcionalment l'envolten unes àrees

DISSENY D'INTERFÍCIES WEB

Figura 2.12. El model de caixa



Visualització de cadascuna de les parts que té una caixa

Propietats per a la caixa

Podem consultar en la [taula 2.14](#) un recull de les propietats per a la caixa.

Taula: 2.14. Propietats per a la caixa

Propietat	Propietats
Marge de la caixa	margin-left, margin-right, margin-top, margin-bottom, margin
Fons	background-color, background-image
Amplada de la vora	border-top-width, border-right-width, border-bottom-width, border-left-width, border-width
Color de la vora	border-top-color, border-right-color, border-bottom-color, border-left-color, border-color
Estil de la vora	border-top-style, border-right-style, border-bottom-style, border-left-style, border-style
Farciment de la caixa	padding-left, padding-right, padding-top, padding-bottom, padding

La propietat `box-sizing` s'utilitza per modificar l'alçada i l'amplada de la caixa dels elements `HTML`, ja que per defecte es calcula sense tenir en compte el farciment i l'amplada de la vora.

Així doncs, la propietat `box-sizing` pot prendre per valor `content-box`, el qual és el valor per defecte de les caixes, i no es té en compte la mida del farciment i de la vora per calcular la mida de la caixa. O bé el valor `border-box`, on sí que es té en compte la mida del farciment i de la vora per calcular la mida de la caixa.

Estils CSS per a la caixa



## DISSENY D'INTERFÍCIES WEB

```

border: 1px solid blue;
}

.div2 {
width: 300px;
height: 100px;
padding: 50px;
border: 1px solid red;
}

.div3 {
width: 300px;
height: 100px;
border: 1px solid blue;
box-sizing: border-box;
}

.div4 {
width: 300px;
height: 100px;
padding: 50px;
border: 1px solid red;
box-sizing: border-box;
}

```

---



---



---

### Utilització dels estils anteriors

```

<h2>Caixes sense box-sizing</h2>
<div class="div1">
  <p>Aquesta caixa és més petita </p>
  <p>(amplada 300px i alçada 100px).</p>
</div>
<br>
<div class="div2">
  <p>Aquesta caixa és més gran</p>
  <p>(amplada 300px i alçada 100px).</p>
</div>

<h2>Caixes amb box-sizing</h2>
<div class="div3">
  <p>Les dues caixes tenen la mateixa mida</p>
  <p>(amplada 300px i alçada 100px).</p>
</div>
<br>
<div class="div4">Fantàstic!!</div>

```

---

El resultat que s'obté es pot veure en la [figura 2.13](#).

**Figura 2.13.** Model de caixa

## DISSENY D'INTERFÍCIES WEB

Aquesta caixa és més petita  
(amplada 300px i alçada 100px).

Aquesta caixa és més gran  
(amplada 300px i alçada 100px).

### Caixes amb box-sizing

Les dues caixes tenen la mateixa mida  
(amplada 300px i alçada 100px).

Fantàstic!!

---

L'exemple anterior el podeu provar en següent enllaç: [codepen.io/ioc-daw-m09/pen/BKmmrx](https://codepen.io/ioc-daw-m09/pen/BKmmrx).

### Posicionament

Els navegadors creen i posicionen de manera automàtica totes les caixes dels elements que formen cada pàgina HTML seguint el flux normal d'aquesta. No obstant això, CSS permet modificar la posició on es mostra la caixa, per així poder aconseguir un disseny determinat.

Per indicar els diferents tipus de posicionament de la caixa s'utilitza la propietat `position`, i els valors que pot prendre són `static`, `relative`, `absolute` o `fixed`.

Els diferents tipus de posicionament de les caixes són:

#### 1. Posicionament estàtic o normal

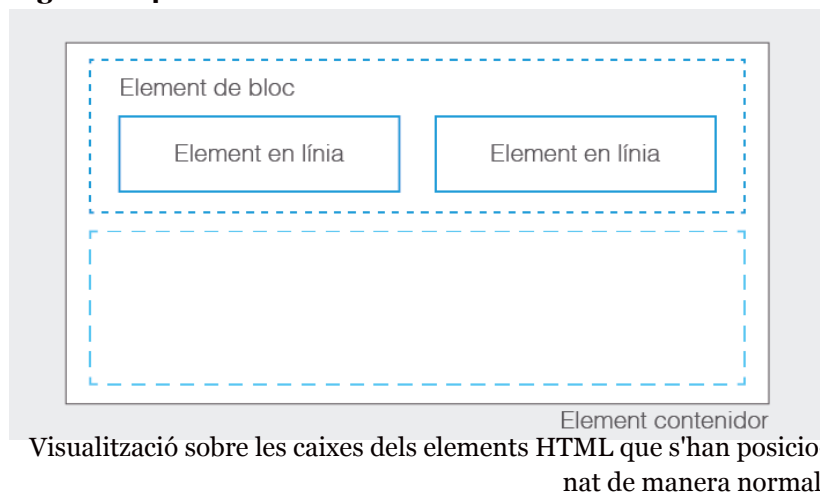
El posicionament de les caixes és estàtic si no s'indica el contrari. En aquest model només es té en compte si l'element és en bloc o en línia, les propietats `width` i `height` i el seu contingut.

En el model de caixa, la distància entre dues cai-

## DISSENY D'INTERFÍCIES WEB

En la [figura 2.14](#) es pot veure com quedarien les caixes amb aquest model.

**Figura 2.14.** Posicionament normal



Així doncs, veieu que per defecte els elements en bloc es mostren un a sota de l'altre, i que els elements en línia es mostren un darrere de l'altre. I si un element es troba dins d'un altre, l'element pare s'anomena *element contenidor* i determina tant la posició com la mida de les caixes del seu interior.

### 2. Posicionament relatiu

Model que consisteix a posicionar una caixa amb el seu posicionament normal i després desplaçar-la.

El desplaçament de la caixa es controla amb les propietats `top`, `bottom`, `right` i `left`. Així doncs, el valor d'aquestes propietats s'utilitza per moure les caixes de manera descendent, ascendent, cap a l'esquerra i cap a la dreta, respectivament. Si s'utilitzen valors negatius el seu efecte serà l'invers. Fixeu-vos, doncs, que aquest comportament és poc intuïtiu i pot causar errors si no s'hi està acostumat.

---

### Estils amb diferents tipus de posicionament

---

```
div {
  position: relative;
  width: 250px;
  height: 180px;
  border: 1px solid black;
  margin: 2px;
  box-sizing: border-box;
  padding: 2px;
  background-color: rgba(255, 0, 0, 0.2);
}

.caixaInline {
  height: 20px;
  display: inline;
}

.caixaBlock {
  width: 142px;
  height: 20px;
  margin-top: 8px;
}
```

## DISSENY D'INTERFÍCIES WEB

```

    left: 5px;
    width: 50px;
    height: 20px;
    display: inline;
}

.caixaAbs {
    position: absolute;
    top: 70px;
    left: 100px;
    width: 50px;
    height: 20px;
    display: inline;
}

.caixaFixa {
    position: fixed;
    top: 180px;
    left: 100px;
    width: 50px;
    height: 20px;
    display: inline;
}

.caixaFlotantDreta{
    float: right;
    width: 50px;
    height: 20px;
}

.caixaFlotantEsquerra{
    float: left;
    width: 50px;
    height: 20px;
}

body {
    font-family: arial;
    font-size: 0.8rem;
}

```

---



---



---

**Utilització dels estils de posicionament anteriors per comprovar la posició relativa respecte a la posició normal.**

```

<div>
  <p>Caixa contenidora: les caixes estan en posició estàtica.</p>
  <div class="caixaInline">caixa1</div>
  <div class="caixaInline">caixa2</div>
  <div class="caixaInline">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>
<div>
  <p>Caixa contenidora: les caixes estan en posició estàtica, excepte<
  <div class="caixaInline">caixa1</div>
  <div class="caixaRel">caixa2</div>
  <div class="caixaInline">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>

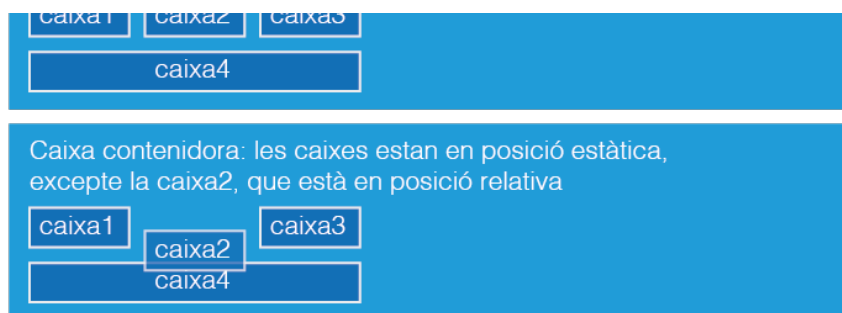
```

---

El desplaçament relatiu d'una caixa no afecta la resta de caixes adjacents, les quals es mostren a la mateixa posició (vegeu la [figura 2.15](#)).

**Figura 2.15.** Posició estàtica

## DISSENY D'INTERFÍCIES WEB



Podeu provar aquest exemple en el següent enllaç: [codepen.io/ioc-daw-mo9/pen/xVPpXX](https://codepen.io/ioc-daw-mo9/pen/xVPpXX).

### 3. Posicionament absolut

En aquest model, la posició de la caixa s'estableix de manera absoluta respecte del seu element contenidor. L'element contenidor de referència serà el que estigui posicionat de qualsevol manera diferent de `position:static`.

Si no hi ha cap element que pugui fer de referència, llavors ho serà la finestra del navegador, que no s'ha de confondre amb l'element `<body>`.

La nova posició de la caixa s'indica mitjançant les propietats `top`, `right`, `bottom` i `left`, de tal manera que:

- El valor de la propietat `top` indica el desplaçament des de la vora superior de la caixa fins a la vora superior de l'element contenidor de referència.
- El valor de la propietat `right` indica el desplaçament des de la vora dreta de la caixa fins a la vora dreta de l'element contenidor de referència.
- El valor de la propietat `bottom` indica el desplaçament des de la vora inferior de la caixa fins a la vora inferior de l'element contenidor de referència.
- El valor de la propietat `left` indica el desplaçament des de la vora esquerra de la caixa fins a la vora esquerra de l'element contenidor de referència.

Les caixes posicionades de manera absoluta **surten del flux normal de la pàgina**, fet que provoca que la resta d'elements es moguin.

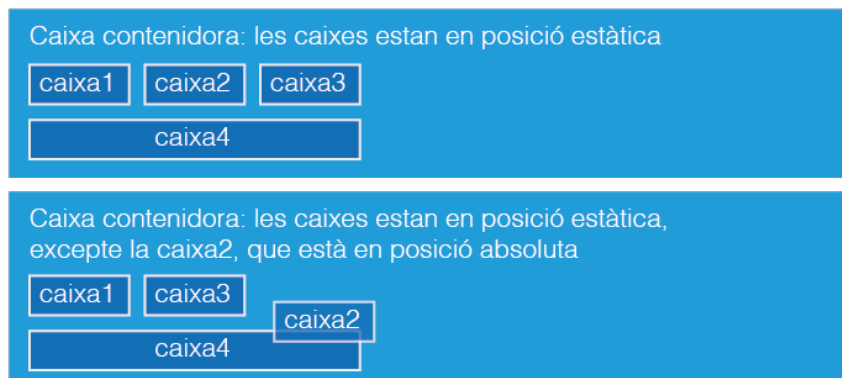
### Utilització dels estils de posicionament anteriors per comprovar la posició absoluta respecte de la posició normal

```
<div>
  <p>Caixa contenidora: les caixes estan en posició estàtica.</p>
  <div class="caixaInline">caixa1</div>
  <div class="caixaInline">caixa2</div>
  <div class="caixaInline">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>
<div>
  <p>Caixa contenidora: les caixes estan en posició estàtica, excepte
  <div class="caixaInline">caixa1</div>
```

## DISSENY D'INTERFÍCIES WEB

La caixa 2 està posicionada de manera absoluta, fet que provoca que la resta d'elements de la pàgina modifiquin la seva posició i ocupin el seu lloc (vegeu la [figura 2.16](#)).

**Figura 2.16.** Posició absoluta



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-daw-mog9pen/aNVEVy](https://codepen.io/ioc-daw-mog9pen/aNVEVy).

### 4. Posicionament fix

En aquest model, la caixa es converteix en un element inamovible, de manera que la seva posició a la pantalla sempre és la mateixa independentment dels altres elements i de si l'usuari puja o baixa la finestra del navegador. És un cas particular de posicionament absolut.

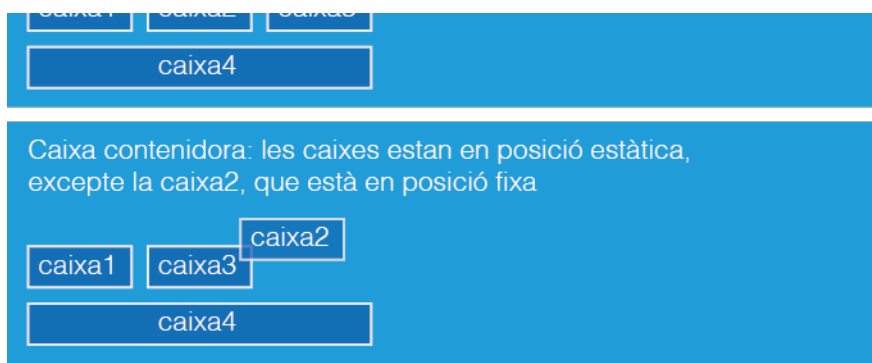
### Utilització dels estils de posicionament anteriors per comprovar la posició fixa respecte a la posició normal

```
<div>
  <p>Caixa contenidora: les caixes estan en posició estàtica.</p>
  <div class="caixaInline">caixa1</div>
  <div class="caixaInline">caixa2</div>
  <div class="caixaInline">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>
<div>
  <p>Caixa contenidora: les caixes estan en posició estàtica, excepte
  <div class="caixaInline">caixa1</div>
  <div class="caixaFixa">caixa2</div>
  <div class="caixaInline">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>
```

El resultat que obtindreu es pot veure en la [figura 2.17](#).

**Figura 2.17.** Posició fixa

## DISSENY D'INTERFÍCIES WEB



Podeu provar l'anterior exemple en l'enllaç següent: [codepen.io/ioc-daw-mo9/pen/PNOERg](https://codepen.io/ioc-daw-mo9/pen/PNOERg).

### 5. Posicionament flotant

En aquest model es desplaça la caixa tot el possible a la dreta o a l'esquerra de la línia en la qual es troba, i llavors aquesta caixa deixa de formar part del flux normal de la pàgina, de manera que la resta de caixes ocupen el seu lloc.

La propietat CSS que permet posicionar una caixa de forma flotant s'anomena `float`, i pot prendre per valors `left`, `right`, `none` i `inherit`.

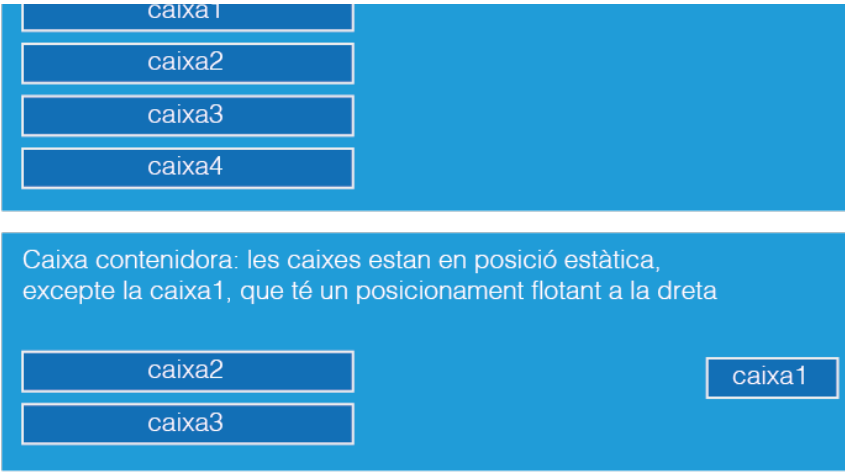
### Utilització dels estils de posicionament anteriors per comprovar la posició flotant respecte a la posició normal

```
<div>
  <p>Caixa contenidora. Les es caixes estan en posició estàtica.</p>
  <div class="caixaBlock">caixa1</div>
  <div class="caixaBlock">caixa2</div>
  <div class="caixaBlock">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>
<div>
  <p>Caixa contenidora. Les caixes estan en posició estàtica, excepte
  <div class="caixaFlotantDreta">caixa1</div>
  <div class="caixaBlock">caixa2</div>
  <div class="caixaBlock">caixa3</div>
</div>
```

El resultat que obtindreu es pot veure en la [figura 2.18](#).

**Figura 2.18.** Posició flotant

DISSENY D'INTERFÍCIES WEB



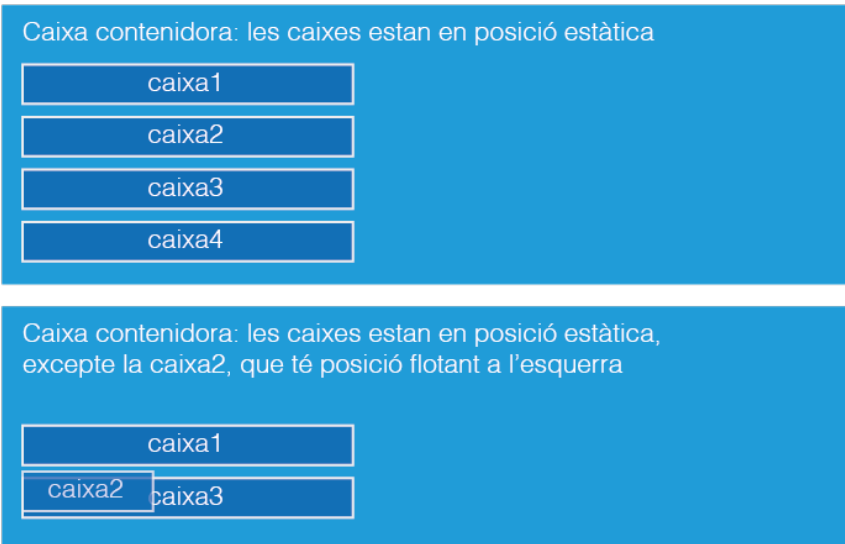
Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-daw-mo9/pen/oxopQM](https://codepen.io/ioc-daw-mo9/pen/oxopQM).

Utilització dels estils de posicionament anteriors per comprovar la posició flotant respecte a la posició normal

```
<div>
  <p>Caixa contenidora. Les caixes estan en posició estàtica.</p>
  <div class="caixaBlock">caixa1</div>
  <div class="caixaBlock">caixa2</div>
  <div class="caixaBlock">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>
<div>
  <p>Caixa contenidora. Les caixes estan en posició estàtica, excepte
  <div class="caixaBlock">caixa1</div>
  <div class="caixaFlotantEsquerra">caixa2</div>
  <div class="caixaBlock">caixa3</div>
</div>
```

El resultat que obtindreu es pot veure en la [figura 2.19](#).

**Figura 2.19.** Posició flotant respecte de la normal





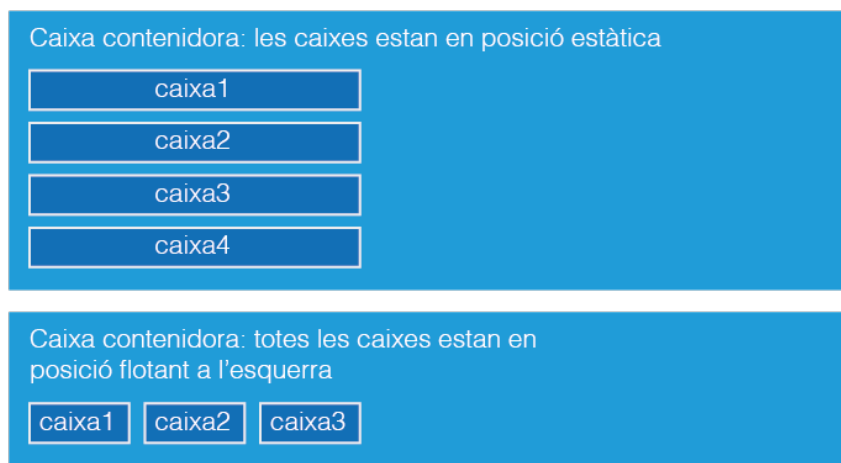
## DISSENY D'INTERFÍCIES WEB

### Utilització dels estils de posicionament anteriors per comprovar la posició flotant respecte a la posició normal

```
<div>
  <p>Caixa contenidora. Les caixes estan en posició estàtica.</p>
  <div class="caixaBlock">caixa1</div>
  <div class="caixaBlock">caixa2</div>
  <div class="caixaBlock">caixa3</div>
  <div class="caixaBlock">caixa4</div>
</div>
<div>
  <p>Caixa contenidora. Totes les caixes estan en posició flotant a
  <div class="caixaFlotantEsquerra">caixa1</div>
  <div class="caixaFlotantEsquerra">caixa2</div>
  <div class="caixaFlotantEsquerra">caixa3</div>
</div>
```

El resultat que obtindreu es pot veure en la [figura 2.20](#).

**Figura 2.20.** Posició flotant respecte a l'esquerra



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/KzyZLN](https://codepen.io/ioc-daw-m09/pen/KzyZLN).

Es poden utilitzar tots els tipus de posicionament, ja que en funció del que vulguem aconseguir ens interessarà posicionar la caixa d'un element utilitzant un tipus o un altre. Tot i això, el més utilitzat és el flotant.

A més a més, la propietat `CSS clear` permet especificar si un element pot estar al costat dels elements flotants que el precedeixen o s'ha de moure cap avall per sota d'ells. Aquesta propietat es pot aplicar tant als elements flotants com als no flotants.

### Exemple d'utilització de la propietat clear

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
```

## DISSENY D'INTERFÍCIES WEB

```

max-width:150px;
min-height: 50px;
background-color: lightblue;
margin-left: 5px;
padding: 20px 2px 2px 20px;

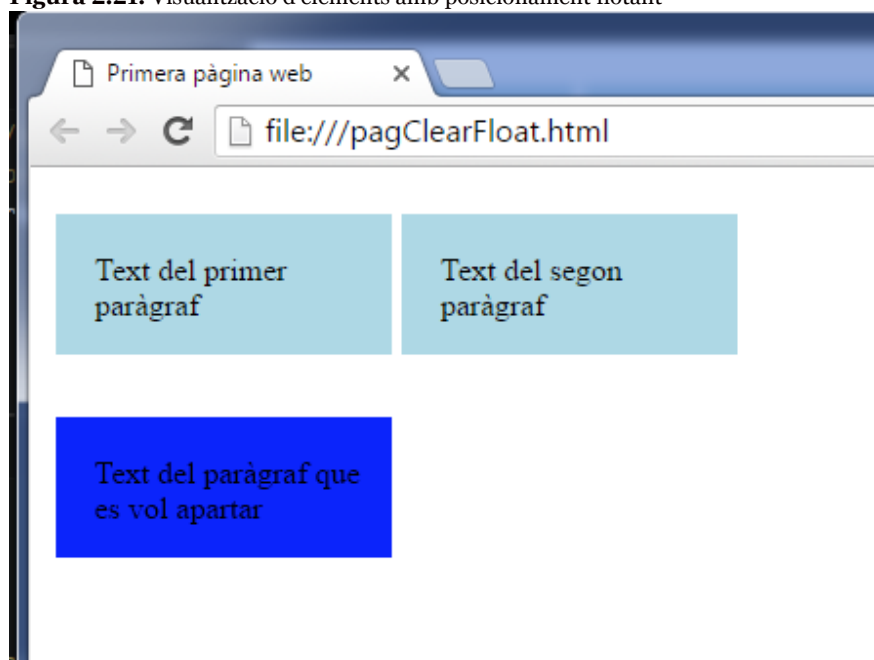
}

p.clear {
  clear: both;
  background-color: blue;
}
</style>
</head>
<body>
<p>Text del primer paràgraf</p>
<p>Text del segon paràgraf</p>
<p class="clear">Text del paràgraf que es vol apartar</p>

</body>
</html>

```

**Figura 2.21.** Visualització d'elements amb posicionament flotant



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-daw-mo9/pen/RRxjoy](https://codepen.io/ioc-daw-mo9/pen/RRxjoy).

### Visualització al navegador

Quan el navegador visualitza una pàgina web crea una caixa per a cada element **HTML**, i aquesta es pot modificar amb **CSS**. Per tant, els factors que té en compte a l'hora de generar cada caixa són:

- La mida de la caixa, si és que s'ha establert a través de les propietats **width** i **height**.
- El tipus d'element **HTML**, és a dir, *block* o *inline*.
- El posicionament de la caixa.
- Les relacions entre elements (on es troba cada element, elements descendents, etc).
- Altre tipus d'informació, com per exemple la mida de les imatges i de la finestra del navegador.

## DISSENY D'INTERFÍCIES WEB

En la [taula 2.15](#) apareixen els valors que poden prendre aquestes propietats.

**Taula: 2.15.** Valors de les propietats

Propietat	Valors
display	inline, block, none, list-item, run-in, inline-block, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption, inherit
visibility	visible, hidden, collapse, inherit
overflow	visible, hidden, scroll, auto, inherit
z-index	auto, numero, inherit

Les propietats `display` i `visibility` permeten amagar qualsevol element de la pàgina. La propietat `display` permet amagar un element perquè desaparegui de la pàgina i així els altres elements puguin ocupar el seu lloc. En canvi, `visibility` permet fer invisible un element; el navegador crearia la caixa, però no la mostraria.

De fet, la propietat `display` ofereix moltes més possibilitats que amagar o mostrar un element, i en realitat modifica la manera com es visualitza un element. Així doncs, els valors més utilitzats d'aquesta propietat són:

- `inline`: mostra l'element com si fos en línia, independentment de si ho és o no.
- `block`: mostra l'element com si fos en bloc, independentment de si ho és o no.
- `none`: oculta l'element i fa que desaparegui de la pàgina.

### Visualització de les propietats "display" i "visibility"

```
<div>
  <p>Caixa2. display:none</p>
  <p>Caixa7. visibility:hidden</p>
  <div class="caixaFlotantEsquerra"> Caixa1 </div>
  <div class="caixaFlotantEsquerra" style="display:none"> Caixa2 </div>
  <div class="caixaFlotantEsquerra"> Caixa3 </div>
  <div class="caixaFlotantEsquerra"> Caixa4 </div>
  <div class="caixaFlotantEsquerra"> Caixa4 </div>
  <div class="caixaFlotantEsquerra"> Caixa6 </div>
  <div class="caixaFlotantEsquerra" style="visibility:hidden"> Caixa7 </div>
  <div class="caixaFlotantEsquerra"> Caixa8 </div>
</div>
```

El resultat es pot veure en la [figura 2.22](#).

**Figura 2.22.** Visualització de les propietats 'display' i 'visibility'

## DISSENY D'INTERFÍCIES WEB



En general, quan s'oculta un element no es vol que segueixi ocupant lloc. Així, per aconseguir aquest efecte és millor usar 'display' que 'visibility'.

Podeu provar l'exemple anterior en el següent enllaç: [codepen.io/ioc-daw-m09/pen/ONQmMg](https://codepen.io/ioc-daw-m09/pen/ONQmMg).

La propietat `overflow` s'utilitza per controlar la manera com es visualitzen els continguts que sobresurten de la caixa que té definida.

### Visualització de la propietat "overflow"

```
<div>
  <p>Caixa1. overflow:hidden</p>
  <p>Caixa4. overflow:visible</p>
  <p>Caixa6. overflow:scroll</p>
  <div class="caixaFlotantEsquerra" style="overflow:hidden"> Caixa1
  <div class="caixaFlotantEsquerra"> Caixa2 </div>
  <div class="caixaFlotantEsquerra"> Caixa3 </div>
  <div class="caixaFlotantEsquerra"> Caixa4 </div>
  <div class="caixaFlotantEsquerra" style="overflow:visible"> Caixa5
  <div class="caixaFlotantEsquerra" style="height: 50px; overflow:scroll"> Caixa6
  <div class="caixaFlotantEsquerra"> Caixa7 </div>
  <div class="caixaFlotantEsquerra"> Caixa8 </div>
</div>
```

El resultat que obtindreu es pot veure en la [figura 2.23](#).

**Figura 2.23.** Visualització de la propietat 'overflow'

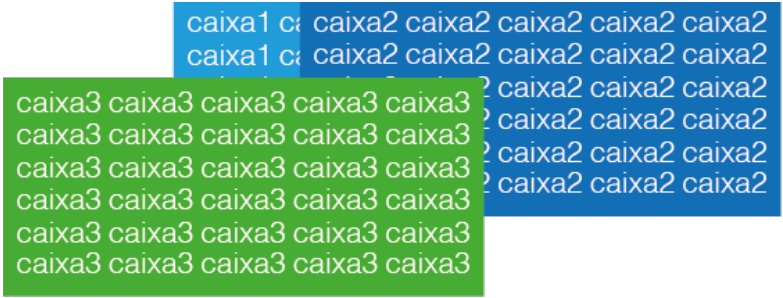


L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-m09/pen/Rajvdr](https://codepen.io/ioc-daw-m09/pen/Rajvdr).

La propietat `z-index` controla la posició tridimensional d'un element que s'estableix sobre el tercer eix Z, i així permet establir quina caixa es veu per damunt d'una altra, tal com es veu en la [figura 2.24](#).

**Figura 2.24.** Posició de les caixes

DISSENY D'INTERFÍCIES WEB



2.1.8. Regles arrova

A més de les regles `CSS` que es defineixen a través del seu selector, hi ha una sèrie de regles anomenades *regles arrova* que tenen usos molt diversos.

Les regles arrova comencen sempre amb el símbol “@” seguit d’un nom clau; es poden declarar en un full extern o incrustat. Algunes es poden consultar en la [taula 2.16](#).

Taula: 2.16. Regles @

Regla	Descripció
@import	Agrega els estils <code>CSS</code> d’un document extern.
@media	Aplica les regles que conté al tipus de dispositiu que s’especifica.
@font-face	Especifica una font no inclosa en el navegador que l’usuari es descarregarà.
@charset	Especifica quin és el joc de caràcters que farem servir dins del fitxer <code>CSS</code> .
@page	Estableix las dimensions, l’orientació i els marges del quadre d’un document.
@supports	Especifica una o diverses condicions, que en cas de complir-se s’aplicaran els estils definits.

A continuació teniu alguns exemples d’utilització d’aquestes regles:

Regla @media

```
/* Definició de diferents estils per diferents tipus de medis.*/
@media print {
  body{ font-size: 10pt }
}
@media screen {
  body { font-size: 12pt }
}
@media screen, print {
  body { line-height: 1.2 }
}
```

Regla @font-face

## DISSENY D'INTERFÍCIES WEB

```
src: url("Delicious-Roman.otf");
}

p {
  font-family: DeliciousRoman, Helvetica, Arial, sans-serif;
}
```

### Regla @charset

```
/* activa el joc de caràcters per al full d'estil a Unicode UTF-8 *,
@charset "UTF-8";
/* activa el joc de caràcters per al full d'estil Latin-9 (llengües
@charset 'iso-8859-15';
```

### Regla @page

```
/* definim els marges d'impressió de la pàgina. */
@page {margin-left: 1cm; margin-right: 0.5cm;}
```

### Regla @supports

```
/*si la propietat display pren per valor flex, s'aplicaran els següents
@supports (display:flex) {
  section { display: flex }
  ...
}
```

#### 2.1.9. Test i validació CSS

Per verificar que el codi CSS és correcte, la gran majoria dels editors d'HTML i CSS contenen unes eines que permeten fer una sèrie de proves per validar-lo.

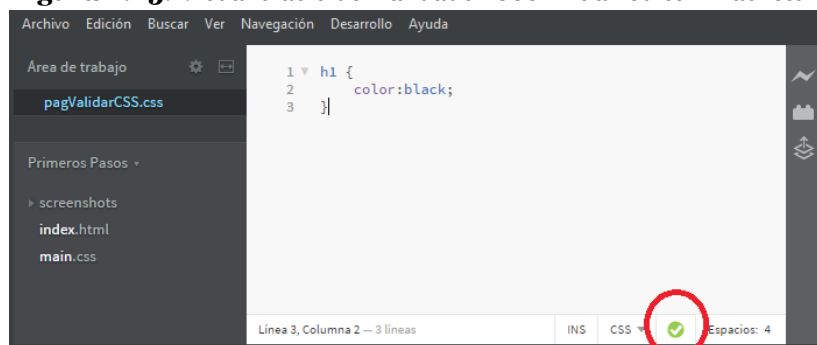
Si es vol fer aquest test amb l'editor Brackets, el que es farà serà utilitzar l'extensió CSSLint.

Aquesta extensió afegeix una icona a la barra d'estat, de tal manera que cada vegada que es guarda el full d'estils es valida.

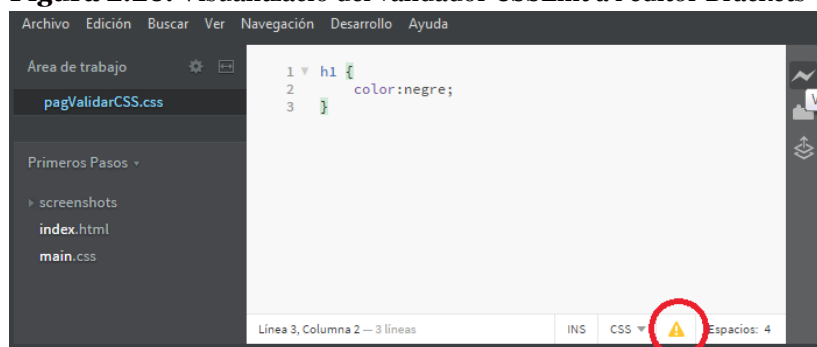
Tal com es mostra en la [figura 2.25](#), en el cas que no hi hagi cap error es mostra una icona de color verd. En cas que sí que n'hi hagi, tal com podeu veure en la [figura](#)

## DISSENY D'INTERFÍCIES WEB

**Figura 2.25.** Visualització del validador CSSLint a l'editor Brackets



**Figura 2.26.** Visualització del validador CSSLint a l'editor Brackets



Aquest validador no tan sols detecta els errors, sinó que també proporciona consells per millorar els fulls d'estils. Tal com es mostra en la [figura 2.27](#), fent clic en el triangle taronja podeu veure els errors detectats.

**Figura 2.27.** Visualització dels errors del validador CSSLint



Alternativament, també es pot verificar que el codi en CSS és correcte mitjançant la web [jigsaw.w3.org/css-validator](http://jigsaw.w3.org/css-validator).

La validació es pot dur a terme de diverses maneres:

- Indicant l'URL complet de la pàgina que desitgem validar.
- Pujant el document CSS.
- Escrivint el codi CSS directament.

Llavors seleccionem *More Options* i allí ens assegurem que a *Perfil* està seleccionat CSS *Versión 3*. Finalment, polsem el botó *Check*, tal com es mostra en la [figura 2.28](#).

**Figura 2.28.** Pantalla del validador de W3

DISSENY D'INTERFÍCIES WEB



CSS Validation Service

Check Cascading Style Sheets (CSS) and (X)HTML documents with style sheets

By URI

By file upload

By direct input

Validate by direct input

Enter the CSS you would like validated:

```
h1 {  
  color: negre;  
}
```

More Options

Profile:

CSS level 3

Medium:

All

Type:

CSS

Warnings:

Normal report

Vendor Extension

Check


En la [figura 2.29](#) veiem la pantalla d'informació dels errors CSS trobats.

**Figura 2.29.** Pantalla d'informació de l'error CSS trobat

← → ↺

https://jigsaw.w3.org/css-validator/validator

🔍 ⭐ ☰



The W3C CSS Validation Service

W3C CSS Validator results for TextArea (CSS level 3)

Jump to:

Errors (1)

W3C CSS Validator results for TextArea (CSS level 3)


Sorry! We found the following errors (1)

URI : TextArea

2

h1

Value Error : color negre is not a color value : negre



The W3C validators are hosted on server technology donated by HP, and supported by community donations.  
[Donate](#) and help us build better tools for a better web.

5641

Flattr

No style sheet found

Home



About

Documentation


Download

Feedback

Credits



COPYRIGHT © 1994-2014 W3C® (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.



En el cas que no hi hagués cap error CSS, es mostraria una pantalla com la de la [figura 2.30](#).

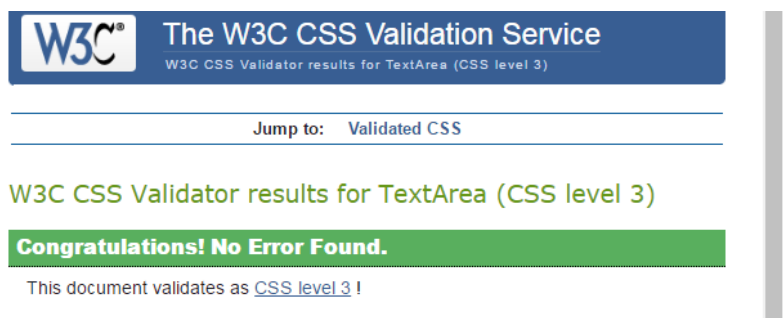
**Figura 2.30.** Pantalla que indica que no hi ha hagut cap error

40 de 74

2/10/20 15:25



## DISSENY D'INTERFÍCIES WEB



## 2.2. Disseny web adaptatiu

El creixent ús de dispositius mòbils i tauletes per accedir a Internet en substitució de l'ordinador fa palès que molts llocs web no estan optimitzats per visualitzar-se correctament en aquests dispositius. Així, molts dels dispositius solen estar limitats per la mida de la pantalla i requereixen un enfocament diferent sobre com presentar el contingut.

El contingut d'un web es podria presentar per pantalla en una columna en un telèfon mòbil o en dues columnes en una tauleta, i en tres columnes en un ordinador.

En conseqüència, tal com es pot apreciar en la [figura 2.31](#), hi ha diferents mides de pantalla en telèfons mòbils, tauletes, ordinadors de sobretaula, consoles de jocs, televisors, etc. Les mides de pantalla seran sempre canviant, per la qual cosa és important que el lloc web pugui adaptar-se a qualsevol mida de pantalla en l'actualitat o en el futur.

**Figura 2.31.** Mida de les pantalles



Es defineix el **disseny adaptatiu** o, en anglès, Responsive Web Design, com una tècnica de disseny i desenvolupament web que permet que un lloc web es vegi correctament en tots els tipus de

## DISSENY D'INTERFÍCIES WEB

El disseny adaptatiu s'aconsegueix amb l'ús dels fulls d'estil CSS aplicats als elements HTML de les pàgines web. Així, es pot canviar la mida, amagar, encongir, engrandir o moure el contingut de la pàgina per fer-lo visible i adequat a qualsevol pantalla.

### 2.2.1. Tècniques per crear un disseny adaptatiu

Per tal que el lloc web sigui adaptatiu és important que disposi de diverses versions perquè es pugui visualitzar correctament des de qualsevol dispositiu amb diferents mides de pantalla i resolucions. D'aquesta manera, almenys caldria disposar de:

- Una versió per a telèfon mòbil.
- Una versió per mòbil en mode apaïsat.
- Una versió per a la tauleta.
- Una versió per a ordinador d'escriptori.

La resolució de pantalla de l'iPhone5S és de 1.136 x 640 px, mentre que la del Nexus5 és 1.920 x 1.080 px.

No s'ha d'oblidar que l'experiència d'ús és molt diferent amb un dispositiu o amb un altre, ja que la mida de la pantalla d'un mòbil acostuma a ser més petita, a pesar que la seva resolució pot ser més gran que la d'un monitor de 17".

Així que no tan sols és necessari adaptar el contingut a la pantalla, sinó que també cal tenir en compte les necessitats de l'usuari i les capacitats del dispositiu. Per això, també caldrà considerar el que vol fer l'usuari en primer lloc i mostrar-li la informació en funció d'aquest criteri. Tampoc s'ha de suposar que l'usuari amb un telèfon mòbil no voldrà accedir a tota la informació del lloc, i per tant caldrà idear la manera d'accedir a la informació, com es mostra i en quin ordre.

Les tècniques que s'utilitzaran per crear el disseny adaptatiu són:

- **Media queries** que permetin saber quin tipus de dispositiu i quines són les dimensions de la pantalla de l'usuari.
- **Un grid flexible**, és a dir, un disseny de pàgina basat en una quadrícula que pugui adaptar-se a la resolució de la pantalla.
- **Continguts i imatges flexibles** als quals, utilitzant estils CSS, s'apliquen canvis d'escala dinàmics per tal que s'adaptin a la mida de la pantalla.

---

### Disseny web fluid 'vs.' adaptatiu

## DISSENY D'INTERFÍCIES WEB

horitzontal (*scroll*).

El problema sorgeix tant en pantalles molt grans com en pantalles molt petites, ja que el disseny es desajusta, bé perquè apareixen grans espais en blanc o bé perquè les imatges es miniaturitzen i els textos es tornen il·legibles.

Per obviar aquests problemes, els dissenyadors poden crear diverses versions en CSS segons la pantalla on s'hagi de visualitzar, suportant així una càrrega de treball addicional sobre el manteniment de les diferents versions.

En un **disseny adaptatiu o *responsive***, el disseny es basa en una quadrícula flexible i els blocs s'ordenen i es jerarquitzen de tal manera que els elements es mostren en una o diverses columnes, en funció de la mida de la pantalla ([figura 2.32](#)).

**Figura 2.32.** Visualització dels elements d'una pàgina web en diferents tipus de dispositiu



### 2.2.2. 'Media queries'

Una **media query** és una consulta que dona informació del tipus de medi en què es visualitza la pàgina web, per així poder limitar els estils que s'hi aplicaran. El resultat d'aquesta consulta serà un booleà, així que quan sigui vertader s'aplicaran les regles d'estil corresponents, seguint les normes de cascada.

Vegem amb un exemple la sintaxi d'una *media query*:

Especifiquem la **consulta** al `<head>` de la pàgina HTML a través de l'element `<link>`.

```
<link rel="stylesheet" media="screen and (width:780px)" href="estils780
```

I la mateixa consulta es podria especificar en un document CSS a través d'una regla `@media`:

## DISSENY D'INTERFÍCIES WEB

}

Els diferents tipus de medis que podem especificar són:

- *screen* (valor per defecte): per a presentació en pantalles d'ordinadors no paginats.
- *print*: per a la sortida per una impressora.
- *projection*: per a presentació en projectors.
- *handheld*: per a dispositius de mà (típicament petita pantalla, d'amplada limitada).
- *aural*: per a sintetitzadors de veu.
- *braille*: per a presentació en dispositius Braille.
- *tty*: per a pantalla en cel·les de caràcters.
- *tv*: per a presentació en televisors.
- *all*: per a tots els dispositius de sortida.

Vegeu en la [taula 2.17](#) els criteris que podem usar en les consultes.

**Taula: 2.17.** Criteris per a les 'media queries'

Criteri	Valor	Mín./Màx.	Descripció
width	Llargada	Sí	Per examinar l'amplada de la zona de visualització del navegador.
height	Llargada	Sí	Per examinar l'altura de la zona de visualització del navegador
device-width	Llargada	Sí	Per examinar l'amplada física de la pantalla de difusió.
device-height	Llargada	Sí	Per examinar l'altura física de la pantalla de difusió.
orientation	<i>Landscape</i> o <i>portrait</i>	No	Per examinar si l'usuari utilitza tauleta tàctil verticalment, <i>portrait</i> o horitzontalment, <i>landscape</i> .
aspect-ratio	Ràtio	Sí	Per examinar el coeficient ample/alt.
device-aspect-ratio	Ràtio	Sí	Per examinar el coeficient físic ample/alt de la pantalla.
color		Sí	Per examinar si el suport de difusió utilitza el color (valor per defecte en cas que no s'hagi especificat), o el blanc i negre o una escala de grisos.
color-index	Número	Sí	Per examinar el número de colors de la taula de colors.
monochrome	Número	Sí	Per examinar el número de nivells de gris per als dispositius monocroms.
resolution	DPI	Sí	Per examinar la resolució de la pantalla de visualització expressada en DPI.
scan	<i>Progressive</i> o <i>interlace</i>	No	Per examinar el tipus d'exploració de les pantalles de televisió.
grid	Número	No	Per examinar si la pantalla de difusió utilitza una quadrícula amb una única mida de font.

## DISSENY D'INTERFÍCIES WEB

Vegem alguns exemples per conèixer el funcionament de diversos criteris de cerca:

### Exemples de 'media queries'

Per examinar si la finestra té una amplada de 700 px o més i la pantalla està en horitzontal:

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

Per examinar si la finestra té una amplada de 700 px o més, la pantalla està en horitzontal i el dispositiu és un televisor:

```
@media tv and (min-width: 700px) and (orientation: landscape) { ... }
```

Per examinar si l'amplada mínima és 700 px o si el dispositiu està en horitzontal:

```
@media (min-width: 700px), handheld and (orientation: landscape) { ... }
```

Per examinar pantalles que no tinguin una resolució de 780 px:

```
@media screen and (not width:780px) {  
    ...  
}
```

Podeu consultar la següent web per obtenir més informació sobre les *media queries*: [developer.mozilla.org/en-US/docs/Web/CSS/Media\\_Queries/Using\\_media\\_queries](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries).

### El 'viewport'

Pràcticament tots els navegadors de dispositius mòbils, en entrar en un lloc web, analitzen la seva mida i l'escalen perquè es mostri completament a la pantalla; en conseqüència, segons com estigui optimitzat el web, és possible que el resultat sigui molt petit.

El **viewport** és l'àrea visible del navegador. En els navegadors per a mòbils no es correspon a la mida real de la pantalla en píxels, sinó a l'espai que la pantalla està emulant que té.

Per exemple, un iPhone que tingui la pantalla en vertical té unes dimensions de 320 px, però en realitat el dispositiu està emulant 980 px.

Per solucionar aquesta situació es pot alterar el *viewport* que està configurat en el navegador a través de l'element HTML `<meta>`, que ubicarem en el `<head>` de la

## DISSENY D'INTERFÍCIES WEB

Les característiques o els paràmetres que podem modificar del *viewport* s'especifiquen a l'atribut `content` separades per coma. Vegeu en la [taula 2.11.18](#) les diferents característiques que podem configurar.

Taula: 2.18. Paràmetres del 'viewport'

Paràmetre	Descripció
width	Amplada del <i>viewport</i> .
height	Altura del <i>viewport</i> .
initial-scale	Escala inicial del document.
minimum-scale	Escala mínima configurable del document.
maximum-scale	Escala màxima configurable del document.
user-scalable	Si es permet o no a l'usuari fer zoom.

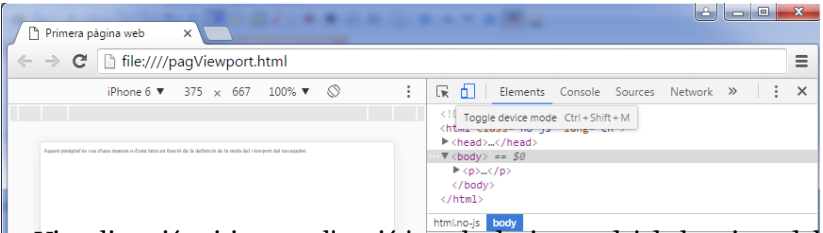
D'aquesta manera, podem fixar l'amplada i l'alçada del *viewport* amb píxels (320 px, 480 px, etc.), o bé també podem usar dues constants: `device-width` i `device-height`, referents a l'amplada i a l'alçada del dispositiu mòbil.

### Exemple de definició del 'viewport'

```
<head>
...
<meta name="viewport" content="width=device-width, initial-scale=1" />
...
</head>
```

A la [figura 2.33](#) i a la [figura 2.34](#) hi veiem com es visualitza el contingut de la pàgina web en un telèfon mòbil si no s'hi ha definit el *viewport*, i com es veu si el *viewport* està definit.

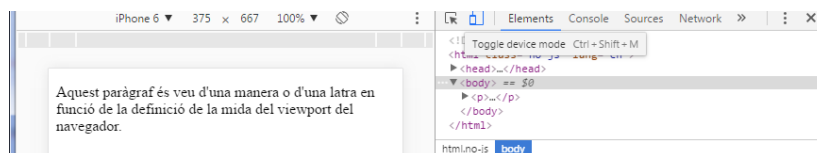
Figura 2.33. Visualització en un iPhone 6 d'una pàgina web sense el 'viewport' definit



Visualització mitjançant l'opció 'toggle device mode' de les eines del desenvolupador que inclou el navegador Chrome

Figura 2.34. Visualització en un iPhone 6 d'una pàgina web amb el 'viewport' definit

## DISSENY D'INTERFÍCIES WEB



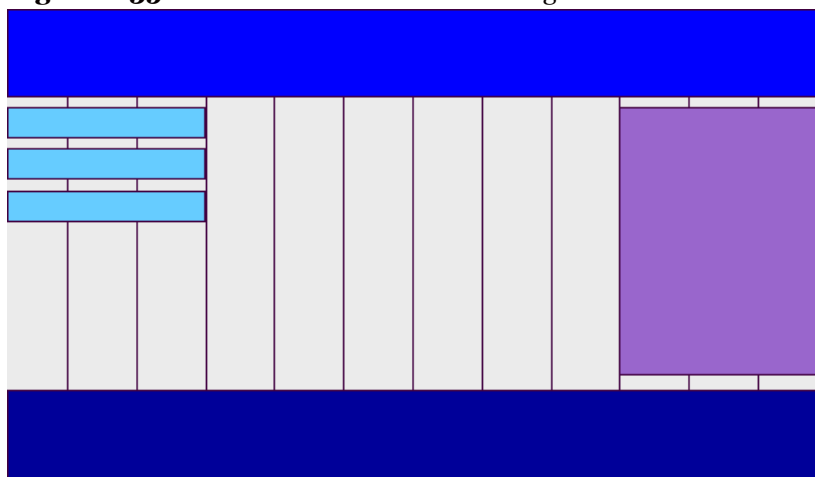
Visualització mitjançant l'opció `//toggle device mode//` de les eines del desenvolupador que inclou el navegador Chrome

Així, veieu que en la primera imatge el *viewport* no està definit i el paràgraf es veu molt petit, i que en la segona sí està definit i per tant el paràgraf es veu correctament.

### 2.2.3. Desenvolupar un 'grid' fluid

Hi ha pàgines web que estan estructurades en el que s'anomena *grid*, que significa que la pàgina està dividida en columnes. D'aquesta manera és molt senzill col·locar-hi els elements, tal com s'aprecia en la [figura 2.35](#).

**Figura 2.35.** Col·locació dels elements en el 'grid'



Visualització sobre la ubicació dels elements d'una pàgina web

Aquest disseny s'aconsegueix definint per a cada fila, *row*, la distribució del nombre de columnes, *col*, que ens interressi. Vegem en l'exemple següent el codi CSS per a la creació d'un *grid* fluid de 12 columnes i l'aplicació d'aquest *grid* en HTML.

### Estils CSS que defineixen un 'grid' fluid

```
* {
  box-sizing: border-box;
}
@media only screen and (max-width: 768px) {
  /* Per a telèfons mòbils: */
  [class*="col-"] {
    width: 100%;
  }
}
.row:after {
  content: "";
  clear: both;
  display: block;
}
[class*="col-"] {
  float: left;
  padding: 15px;
}
```

## DISSENY D'INTERFÍCIES WEB

```
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
}
html {
  font-family: "Lucida Sans", sans-serif;
}
.header {
  background-color: #9933cc;
  color: #ffffff;
  padding: 15px;
}
.footer {
  background-color: #9933cc;
  color: #ffffff;
  padding: 15px;
  text-align: center;
}
.menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
.menu li {
  padding: 8px;
  margin-bottom: 7px;
  background-color: #33b5e5;
  color: #ffffff;
  box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.1);
}
.menu li:hover {
  background-color: #0099cc;
}
.aside p {
  height: 100px;
  padding: 8px;
  margin: 0;
  background-color: #33b5e5;
  color: #ffffff;
  box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.1);
}
```

---



---



---

## Exemple d'utilització del 'grid' definit en els estils anteriors

```
<!doctype html>
<html>
<head>
  ...
<meta name="viewport" content="width=device-width, initial-scale=1.0">
  ...
</head>
<body>
<header>
  <div class="header">
    <h1>Estudis de grau superior d'Informàtica</h1>
  </div>
</header>

<content>
  <div class="row">
    <div class="col-9">
      <div class="row">
        <nav>
          <div class="col-3 menu">
            <ul>
              <li>ASIX</li>
              <li>DAM</li>
              <li>DAW</li>
            </ul>
          </div>
        </nav>
      </div>
    </div>
  </div>
</content>
```



DISSENY D'INTERFÍCIES WEB

```
<h2>M1. Sistemes informàtics</h2>
<p>Bloc 1. 118 hores (8 hores setmanals)</p>
<p>Bloc 2. 80 hores (5 hores setmanals)</p>
</article>
</div>
</div>
<div class="row">
  <div class="col-3"></div>
  <div class="col-9">
    <article>
      <h2>M2. Bases de dades</h2>
      <p>Bloc 1. 132 hores (9 hores setmanals)</p>
      <p>Bloc 2. 99 hores (7 hores setmanals). Inclou treball en
    </article>
  </div>
</div>
<div class="row">
  <div class="col-3"></div>
  <div class="col-9">
    <article>
      <h2>M3. Programació</h2>
      <p>Bloc 1. 85 hores (5 hores setmanals). Inclou treball en
      <p>Bloc 2. 80 hores (5 hores setmanals). Inclou treball en
    </article>
  </div>
</div>
</div>
<div class="col-3">
  <div class="row">
    <div class="col-12">
      <aside>
        <div class="aside">
          <p> links d'interès</p>
        </div>
      </aside>
    </div>
  </div>
</div>
</div>
</content>

<footer>
  <div class="footer">
    <p>IOC - Institut Obert de Catalunya</p>
  </div>
</footer>
</body>
</html>
```

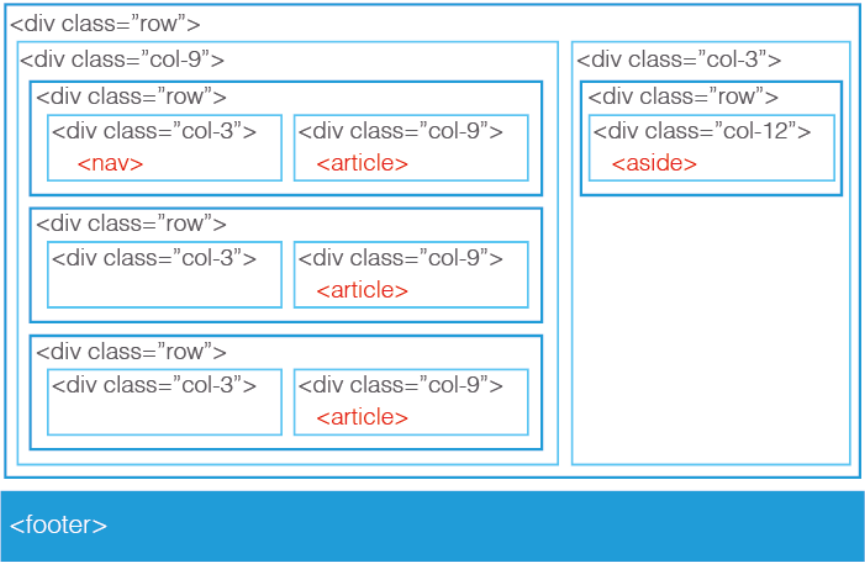
El resultat apareix en la [figura 2.36](#) i la divisió de l'espai en la [figura 2.37](#).

Figura 2.36. Visualització de la pàgina web



Figura 2.37. Divisió de l'espai

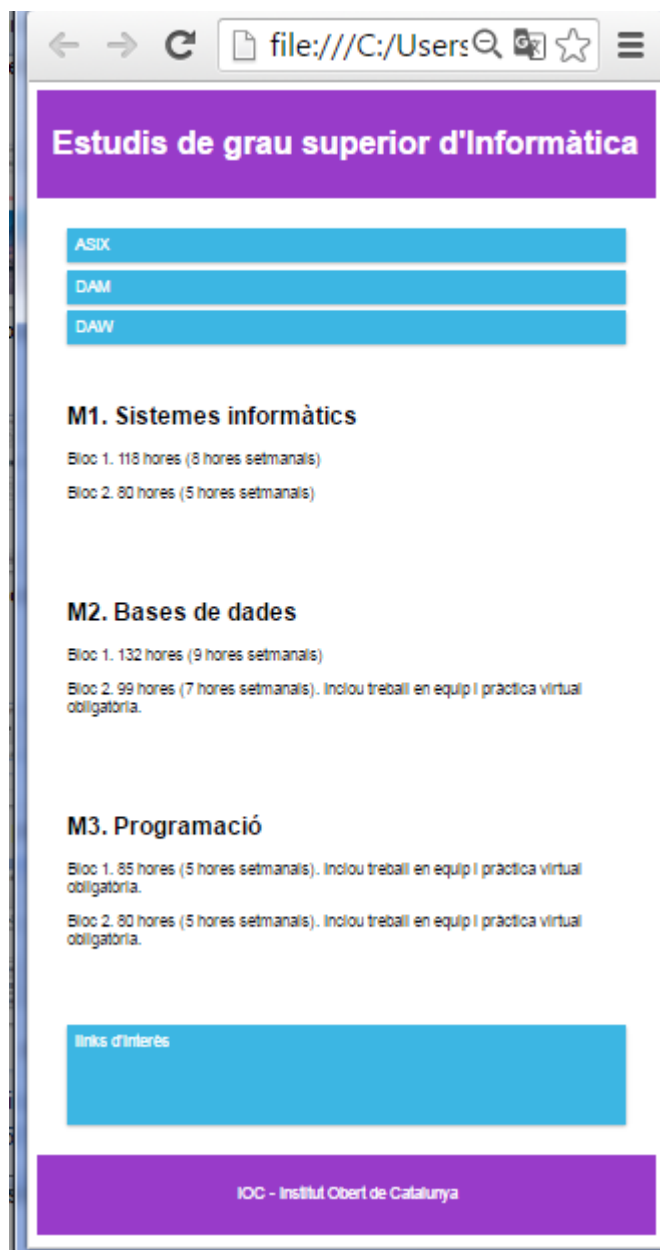
DISSENY D'INTERFÍCIES WEB



I en la [figura 2.38](#) es visualitza com es veuria en una pantalla petita.

**Figura 2.38.** Visualització en pantalla petita

## DISSENY D'INTERFÍCIES WEB



Podeu provar aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/WwjzGb?editors=1100](https://codepen.io/ioc-daw-m09/pen/WwjzGb?editors=1100).

Fixeu-vos en l'exemple anterior, com en una fila (*row*) només hi pot haver columnes (*col*) i que el contingut sempre se situa dins de les columnes. Alhora teniu que dins d'una columna hi pot haver una altra fila, la qual també es dividirà en 12 columnes.

Amb aquest disseny de *grid* es veu que els blocs es posicionen en un lloc o un altre en funció de la mida de la pantalla. Aquest funcionament és degut al fet que la posició de les columnes és flotant, i que per a mides de pantalla inferiors a 768 px l'amplada de la columna és del 100%.

Cal remarcar la dificultat que hi pot haver en aquest tipus de *grid* si es vol canviar la disposició d'alguns dels blocs quan es canvia la mida de la pantalla, com per exemple fer que el menú de navegació (els botons *ASIX*, *DAM* i *DAW*) apareguin damunt

DISSENY D'INTERFÍCIES WEB

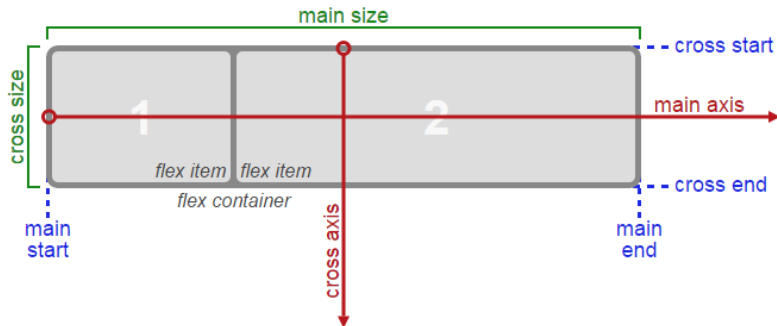
2.2.4. Desenvolupar un 'grid' flexible

El desenvolupament d'un *grid* flexible es basa en el nou model de caixa, **flexbox**. Així doncs, podem crear una caixa o contenidor flexible a través de la propietat `display: flex`. D'aquesta manera, la declaració d'aquesta propietat converteix de manera automàtica els seus elements fills directes en flexibles.

Un contenidor flexible té un eix principal, *main axis*, que és la direcció en la qual es posicionen els elements flexibles, i també té un eix transversal, perpendicular a l'eix principal. Aquests dos eixos tenen una sèrie de propietats que controlen com es posiciona cada element flexible en relació amb els altres.

Vegeu en la [figura 2.39](#) la disposició dels dos eixos.

Figura 2.39. Model de caixa flex



Font: <https://www.w3.org/TR/css-flexbox-1/>

Vegem en la [taula 2.19](#) les diferents propietats `CSS` que ens permetran crear el contenidor flexible.

Taula: 2.19. Propietats `CSS` per al contenidor flexible

Propietat	Descripció
<code>flex-direction</code>	Especifica com se situen els elements dins del contenidor. Els valors que pot prendre són <i>row</i> , <i>row-reverse</i> , <i>column</i> , <i>column-reverse</i> .
<code>flex-wrap</code>	Especifica si el contenidor té una o diverses línies. Els valors que pot prendre són: <i>no-wrap</i> , <i>wrap</i> , <i>wrap-reverse</i> .
<code>flex-flow</code>	Propietat abreujada de les propietat anteriors. <i>flex-flow: row wrap</i> .

Exemple de caixa flexible

```
#caixaFlex {
  display: flex;
  display: -webkit-flex;
  flex-direction: row;
  -webkit-flex-direction: row;
  flex-wrap: wrap;
  -webkit-flex-wrap: wrap;
}
```

DISSENY D'INTERFÍCIES WEB

Les imatges i les imatges que formen part de la pàgina web també han de ser flexibles. No és objectiu d'aquesta unitat com mostrar imatges flexibles; per tant, ens limitarem a tractar el contingut flexible.

En la [taula 2.20](#) es poden consultar les propietats CSS que es poden assignar als elements que estan dins del contenidor.

Taula: 2.20. Propietats CSS per al contingut flexible

Propietat	Descripció
order	Per establir l'ordre en què apareixen els elements d'una caixa flexible. Per defecte és 0.
flex-grow	Especifica el factor de creixement, és a dir, quant creix un element en relació amb els altres quan hi ha espai disponible en el contenidor. Per defecte és 0.
flex-shrink	Especifica el factor de reducció, és a dir, quant decreixerà un element en relació amb els altres quan no hi hagi espai disponible en el contenidor. Per defecte és 1.
flex-basis	Pren el mateix valor que la propietat width. Especifica la mida inicial de l'element abans de distribuir l'espai lliure amb les propietats anteriors. Per defecte és main-size (auto).
flex	Propietat abreujada de les anteriors.

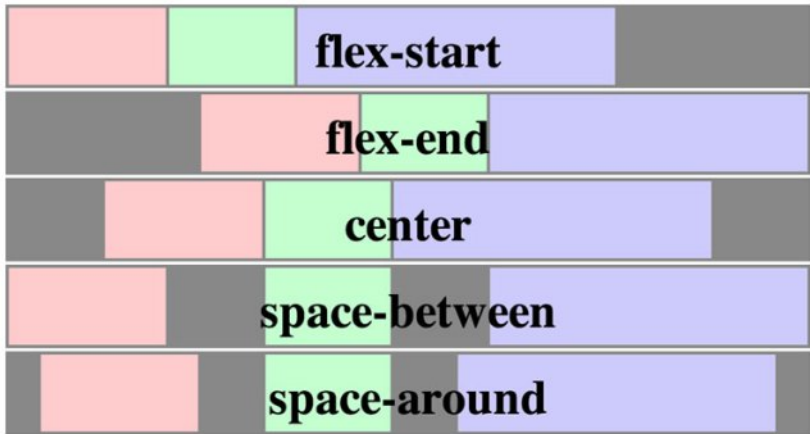
Les propietats per alinear els elements flexibles es poden consultar en la [taula 2.21](#).

Taula: 2.21. Propietats CSS per alinear els elements flexibles

Propietat	Descripció
justify-content	Permet alinear els elements en l'eix principal. Pren per valor flex-start, flex-end, center, space-between, space-around.
align-items	Permet alinear tots elements en l'eix transversal. Pren per valor: auto, flex-start, flex-end, center, baseline, stretch.
align-self	Permet alinear els elements individuals en l'eix transversal; així doncs, sobreescriu la propietat anterior.

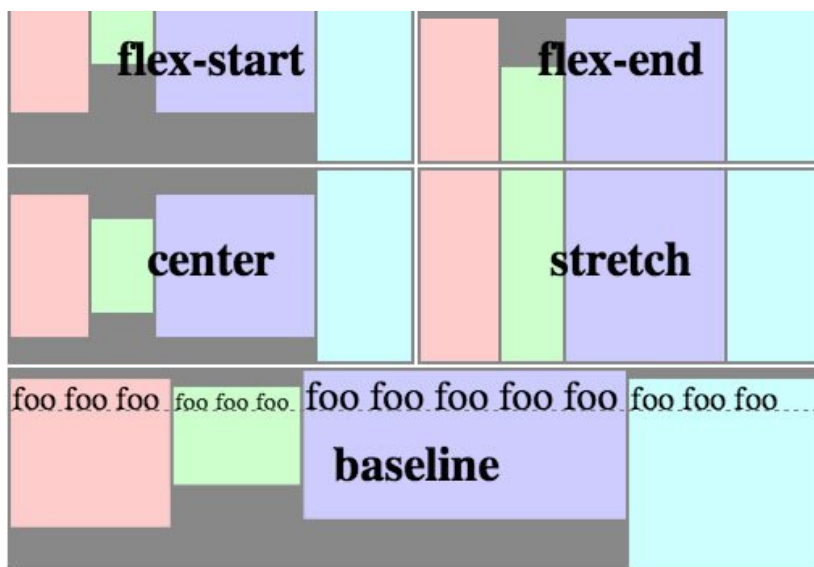
En la figura [figura 2.40](#) i la [figura 2.41](#) es pot apreciar com s'alineen els elements en funció de les propietats anteriors.

Figura 2.40. Alineament sobre l'eix principal



Font: <https://www.w3.org/TR/css-flexbox-1/>

## DISSENY D'INTERFÍCIES WEB



Font: <https://www.w3.org/TR/css-flexbox-1/>

Vegem un exemple sobre com definir un contenidor i elements flexibles:

### Estils per a un disseny flexible

```
body {
  font: 18px Arial;
  background: #eeeeee;
}

#main {
  min-height: 450px;
  margin: 0px;
  padding: 0px;
  display: -webkit-flex;
  display: flex;
  -webkit-flex-flow: row;
  flex-flow: row;
}

#main > article {
  margin: 4px;
  padding: 5px;
  border: 1px solid #ccc;
  border-radius: 2pt;
  background: #white;
  -webkit-flex: 3 1 60%;
  flex: 3 1 60%;
  -webkit-order: 2;
  order: 2;
}

#main > nav {
  margin: 4px;
  padding: 5px;
  border: 1px solid #8888bb;
  border-radius: 2pt;
  background: #33b5e5;
  -webkit-flex: 1 6 20%;
  flex: 1 6 20%;
  -webkit-order: 1;
  order: 1;
}

#main > aside {
  margin: 4px;
  padding: 5px;
  border: 1px solid #8888bb;
  border-radius: 2pt;
  background: #33b5e5;
  -webkit-flex: 1 6 20%;
  flex: 1 6 20%;
  -webkit-order: 3;
  order: 3;
}
```

## DISSENY D'INTERFÍCIES WEB

```
display: block;
margin: 2px;
padding: 15px;
min-height: 50px;
border: 1px solid #771199;
border-radius: 2pt;
background: #9933cc;
color: #ffffff;
text-shadow: 1px 1px #aaa;
}

@media all and (max-width: 640px) {
  #main,
  #page {
    -webkit-flex-flow: column;
    flex-flow: column;
  }
  #main > article,
  #main > nav,
  #main > aside {
    -webkit-order: 0;
    order: 0;
  }
  #main > nav,
  #main > aside,
  header,
  footer {
    min-height: 50px;
  }
}
```

### Exemple d'utilització dels estils anteriors

```
<header>
  <h1>Estudis de grau superior d'Informàtica</h1></header>
<div id='main'>
  <article>article</article>
  <nav>menú de navegació</nav>
  <aside>Links d'interès</aside>
</div>
<footer>IOC - Institut Obert de Catalunya</footer>
```

El resultat es pot visualitzar en la [figura 2.42](#).

**Figura 2.42.** Elements flexibles



## DISSENY D'INTERFÍCIES WEB

Heu vist a l'exemple anterior la utilització de les propietats `display`, `flex` i `order`, i també com el contingut de la pàgina en `HTML` es simplifica moltíssim.

Fixeu-vos també com, per assegurar la compatibilitat amb altres navegadors, es repeteix la mateixa propietat `CSS` amb el seu prefix `-webkit-`.

### 2.3. Bootstrap 'framework'

En moltes ocasions, per crear una interfície web es pot partir d'una infraestructura o esquelet que ja conté molts dels elements necessaris per a la implementació d'una aplicació web.

Un **framework**, en general, és una estructura conceptual de suport definida, amb mòduls concrets de *software* que poden ser utilitzats com a base per a l'organització i el desenvolupament de *software*.

L'objectiu dels *frameworks* és proporcionar una estructura comuna perquè els desenvolupadors no ho hagin de fer des de zero i puguin reutilitzar el codi proporcionat.

Actualment existeixen diversos *frameworks* per poder desenvolupar el nostre lloc web, com poden ser: Bootstrap ([getbootstrap.com](https://getbootstrap.com)), Angular ([angularjs.org](https://angularjs.org)), Foundation ([oundation.zurb.com](https://foundation.zurb.com)), Leaf ([getleaf.com](https://getleaf.com)) o Materialize ([materializecss.com](https://materializecss.com)), entre molts d'altres.

En aquest apartat s'utilitzarà el *framework* **Bootstrap 3**. Les seves característiques més rellevants són:

- És un *framework* fàcil, ràpid i gratuït per al desenvolupament de la interfície web.
- Inclou plantilles basades en `HTML` i `CSS`, així com *plugins* JavaScript.
- Proporciona la capacitat de crear dissenys adaptatius.

#### 2.3.1. Com obtenir i instal·lar Bootstrap

Hi ha dues formes d'iniciar l'ús de *Bootstrap* en el nostre lloc web:

1. Descarregar Bootstrap a [getbootstrap.com](https://getbootstrap.com) i seguir les instruccions que s'hi proporcionen.
2. Incloure Bootstrap d'una CDN (Content Delivery Network).

#### Avantatge d'usar una CDN



## DISSENY D'INTERFÍCIES WEB

Bootstrap d'una CDN, quan es visiti el nostre lloc web es carregarà des de la memòria cau. A més a més, quan l'usuari en faci la petició, aquesta serà servida del servidor més pròxim a ell.

Una vegada descarregat, el descomprimirem i obtindrem els fitxers necessaris per utilitzar Bootstrap:

- **Carpeta *css*:** aquí hi trobem els fitxers *bootstrap.css* i *bootstrap-theme.css*. El primer és el que hem de vincular en els nostres projectes. I el segon és opcional, conté estils prefabricats per a botons, barres de navegació, etc. Els fitxers *.min* són els mateixos; la diferència és la mida i la lectura, ja que a la versió mínima s'han tret les tabulacions i salts de línia.
- **Carpeta *js*:** aquí hi trobem els fitxers *bootstrap.js* i la seva versió reduïda *bootstrap.min.js*. Aquest fitxer conté totes les llibreries JavaScript, rutines que fan que Bootstrap sigui més dinàmic i interactiu. Per exemple, menús desplega-bles i altres funcionalitats.
- **Carpeta *fonts*:** hi trobem les fonts *glyphicon* Bootstrap, que ens proporcionen l'opció d'utilitzar fonts enlloc d'imatges per mostrar icones bàsiques. Amb això aconseguim que el lloc sigui molt més ràpid de carregar i que consumeixi menys ample de banda. Tots els fitxers d'aquesta carpeta responen a la mateixa font, amb diferents formats que s'adeqüen als diferents navegadors.

Per vincular el *framework* Bootstrap descarregat a la nostra pàgina web:

---

### Inclusió del 'framework' Bootstrap descarregat

---

```
<head>
...
<link rel="stylesheet" href="css/bootstrap.min.css" />
</head>

<script src="js/bootstrap.min.js"></script>
```

---

De manera alternativa, podem utilitzar els estils Bootstrap des d'una CDN. Inclourem en el codi:

---

### Inclusió del 'framework' Bootstrap des d'una CDN

---

```
<head>
...

<!-- estils CSS -->
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstr:

<!-- llibreria JQuery -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jq

<!-- plugins JavaScript -->
```

## DISSENY D'INTERFÍCIES WEB

### 2.3.2. El contenidor

Bootstrap 3 requereix d'un element contenidor per embolcallar els continguts del lloc i col·locar el *grid bootstrap*. Podem triar dos tipus de contenidors:

- **.container**: defineix un contenidor adaptable d'amplada fixa.
- **.container-fluid**: defineix un contenidor adaptable d'ample total, és a dir, la seva amplada es correspon a l'amplada del *wiewport*.

### Definició d'un contenidor fluid

```
<!DOCTYPE html>
<html lang="en">
<head>
  ...
  <meta name="viewport" content="width=device-width, initial-scale=:
</head>
<body>
  <div class="container-fluid">
    ...
  </div>
</body>
</html>
```

### 2.3.3. Bootstrap 'grid'

Els sistema de graella, *grid*, s'utilitza per crear dissenys de pàgina a través d'una sèrie de files i columnes que acullen el contingut. Bootstrap inclou un sistema *grid responsive* fluid i *mobile firts* de 12 columnes com a màxim.

Els preprocessadors SASS i LESS són eines que permeten escriure pseudocodi CSS (variables, condicions, bucles, funcions) que serà convertit en CSS real.

Aquest sistema funciona de la següent manera:

- Les files, *rows*, han d'anar dins d'un contenidor d'amplada fixa o fluida. D'aquesta manera, s'estableix el seu alineament i farciment.
- El contingut es col·loca dins de les columnes.

En la [figura 2.43](#) hi podem veure una representació del *grid* de Bootstrap en 12 columnes i com es poden combinar entre si.

**Figura 2.43.** Representació del Bootstrap 'grid'

## DISSENY D'INTERFÍCIES WEB

span 4	span 8
span 6	span 6
span 12	

Per obtenir més informació sobre els estils que conformen el *grid* de Bootstrap podeu consultar la web: [getbootstrap.com/css/#grid](https://getbootstrap.com/css/#grid).

El sistema Bootstrap *grid* ofereix quatre mides de quadrícula per als diferents tipus de dispositius. Així doncs, la classe que s'utilitza per definir cada fila de la quadrícula és `.row`, i les classes que s'utilitzen per definir les columnes són:

- `.col-xs-(num_columnes)`: per a telèfons mòbils (<768 px).
- `.col-sm-(num_columnes)`: per a tauletes (>=768 px).
- `.col-md-(num_columnes)`: per a ordinadors (>=992 px).
- `.col-lg-(num_columnes)`: per a pantalles grans (>=1200 px).

### Exemple utilització de diferents mides de columnes

```
<div class="container">
<div class="row">
  <div class="col-xs-12">
    <p>Dispositius molt petits (Telèfons, amplada més petita de 768;
  </div>
</div>
<div class="row">
  <div class="col-xs-6" style="background-color:purple;height:40px;"
  <div class="col-xs-6" style="background-color:pink;height:40px;">
</div>

<div class="row">
  <div class="col-sm-12">
    <p>Dispositius petits (tauletes, amplada més gran o igual a 768;
  </div>
</div>
<div class="row">
  <div class="col-sm-6" style="background-color:purple;height:40px;"
  <div class="col-sm-6" style="background-color:pink;height:40px;">
</div>

<div class="row">
  <div class="col-md-12">
    <p>Dispositius mitjans (ordinadors, amplada més gran o igual a 992;
  </div>
</div>
<div class="row">
  <div class="col-md-6" style="background-color:purple;height:40px;"
  <div class="col-md-6" style="background-color:pink;height:40px;">
</div>

<div class="row">
  <div class="col-lg-12">
    <p>Dispositius grans (ordinadors, amplada més gran o igual a 1200;
  </div>
</div>
<div class="row">
  <div class="col-lg-6" style="background-color:purple;height:40px;"
  <div class="col-lg-6" style="background-color:pink;height:40px;">
</div>
</div>
```

El resultat serà el que es pot veure en la [figura 2.44](#).

**Figura 2.44.** Representació de les mides de columnes

## DISSENY D'INTERFÍCIES WEB



En la [figura 2.44](#) podeu veure com totes les files es parteixen en dues columnes horitzontals, però quan modifiquem la mida del navegador cada fila s'acomoda verticalment en funció de la classe i de la resolució. Així doncs, els `divs` `col-sm-6` mantindran la seva estructura en dues columnes alineades horitzontalment, però quan la mida del *viewport* sigui menor a 768 px aquestes s'acomodaran verticalment. Igualment passa amb les altres.

Podeu provar l'exemple anterior en el següent enllaç: [codepen.io/ioc-daw-mog/pen/EKQPdZ](https://codepen.io/ioc-daw-mog/pen/EKQPdZ).

A més a més, Bootstrap proporciona columnes d'amplada mixtes. Suposem que volem distribuir l'espai en dues columnes amb el 50% cadascuna, de tal manera que les dues columnes no ocupin sempre el 50%, sinó que en dispositius més grans de 1.200 px aquestes *divs* ocupin el primer un 33% i el segon un 66%. Farem:

### Exemple d'utilització de columnes d'amplada mixta

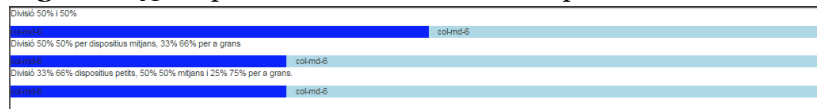
```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <p>Divisió 50% i 50% </p>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6" style="background-color:blue;">col-md-6</div>
    <div class="col-md-6" style="background-color:lightblue;">col-md-6</div>
  </div>
  <div class="row">
    <div class="col-md-12">
      <p>Divisió 50% 50% per dispositius mitjans, 33% 66% per a grans</p>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6 col-lg-4" style="background-color:blue;">col-md-6</div>
    <div class="col-md-6 col-lg-8" style="background-color:lightblue;">col-md-6</div>
  </div>
  <div class="row">
    <div class="col-md-12">
      <p>Divisió 33% 66% dispositius petits, 50% 50% mitjans i 25% 75% grans</p>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-3 col-md-6 col-lg-4" style="background-color:blue;">col-sm-3</div>
    <div class="col-sm-9 col-md-6 col-lg-8" style="background-color:lightblue;">col-sm-9</div>
  </div>
</div>
```

Divisió 50%/50% per a dispositius mitjans, 33%/66% per a grans; Divisió 33%/66% per a dispositius petits, 50%/50% per a mitjans i 25%/75% per a grans.

## DISSENY D'INTERFÍCIES WEB

Vegeu a la [figura 2.45](#) com la mida de les columnes es redimensiona en funció de l'amplada i la resolució del *viewport*.

**Figura 2.45.** Representació de columnes d'amplada mixta



De la mateixa manera, en Bootstrap també disposem d'utilitats per ocultar i mostrar un element basant-nos en la mida del dispositiu i la resolució del *viewport*.

Així doncs, podem usar les classes següents per mostrar elements *div*: `.visible-xs`, `.visible-sm`, `.visible-md` i `.visible-lg`. I per amagar elements *div* basant-nos en la mida del *viewport*: `.hidden-xs`, `.hidden-sm`, `.hidden-md` i `.hidden-lg`.

### Netejar columnes

A causa dels quatre nivells de quadrícula disponibles, ens trobem amb certs problemes quan canviem la disposició en les diferents resolucions, ja que les columnes poden no tenir el mateix contingut i algunes són més altes que d'altres. Per solucionar-ho utilitzem la classe `.clearfix`.

### Exemple sense "clearfix"

```
<div class="container">
<div class="row">
  <div class="col-xs-6 col-sm-3" style="background-color:red;height:
  <div class="col-xs-6 col-sm-3" style="background-color:blue;">.co
  <div class="col-xs-6 col-sm-3" style="background-color:green;">.co
  <div class="col-xs-6 col-sm-3" style="background-color:yellow;">.co
</div>
</div>
```

El resultat que obtindreu és el que es pot veure en la [figura 2.46](#).

**Figura 2.46.** Visualització sense `clearfix`



L'exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-m09/pen/ONQXax](https://codepen.io/ioc-daw-m09/pen/ONQXax).

Vegeu a l'exemple anterior on es posiciona la columna verda en canviar la mida del *viewport*.

### Exemple amb "clearfix"

## DISSENY D'INTERFÍCIES WEB

```
<div class="col-xs-6 col-sm-3" style="background-color:blue;">.co:
<div class="clearfix visible-xs"></div>
<div class="col-xs-6 col-sm-3" style="background-color:green;">.co
<div class="col-xs-6 col-sm-3" style="background-color:yellow;">.c
</div>
</div>
```

El resultat és el que es pot veure en la [figura 2.47](#).

**Figura 2.47.** Visualització amb `clearfix`



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-daw-mog9/pen/WwMxYJ](https://codepen.io/ioc-daw-mog9/pen/WwMxYJ).

Vegeu en l'exemple anterior com, amb la utilització de la classe `clearfix`, la columna verda s'alinea amb la columna de la seva esquerra.

### Desplaçar columnes

Utilitzem la classe `-offset-` per desplaçar qualsevol columna cap a la dreta. Aquesta classe augmenta la mida del marge esquerre de la columna en una quantitat equivalent a aquest nombre de columnes. Per exemple, la classe `.col-md-offset-6` desplaça la columna cap a la dreta o dóna marges a l'esquerra d'una amplitud equivalent a sis columnes.

### Exemple de desplaçament de columnes

```
<div class="container">
<div class="row">
  <div class="col-md-4" style="background-color:red;">.col-md-4</div>
  <div class="col-md-4 col-md-offset-4" style="background-color:blue;">.col-md-4</div>
</div>

<div class="row">
  <div class="col-md-3 col-md-offset-3" style="background-color:green;">.col-md-3</div>
  <div class="col-md-3 col-md-offset-3" style="background-color:yellow;">.col-md-3</div>
</div>

<div class="row">
  <div class="col-md-6 col-md-offset-3" style="background-color:purple;">.col-md-6</div>
</div>
</div>
```

El resultat és el que es pot veure en la [figura 2.48](#).

**Figura 2.48.** Exemple de desplaçament de columnes



L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-mog9/pen/wGyWNY](https://codepen.io/ioc-daw-mog9/pen/wGyWNY).

Vegeu com entre la columna vermella i blava hi ha un offset igual a la mida de tres columnes, i com les columnes verda, groga i també la vermella tenen un offset de tres

DISSENY D'INTERFÍCIES WEB

Reordenar elements

Amb les classes `-pull-`, el que fem és estirar l'element a la seva esquerra el nombre de columnes que especifiquem. I amb les classes `-push-`, el que fem és empènyer l'element a la dreta el nombre de columnes que especifiquem.

Exemple de reordenació de columnes

```
<div class="container">
<div class="row">
  <div class="col-md-9 col-md-push-3" style="background-color:red;">
  <div class="col-md-3 col-md-pull-9" style="background-color:blue;">
</div>
</div>
```

El resultat es pot veure en la figura [figura 2.49](#).

Figura 2.49. Ordenació d'elements



Podeu provar l'exemple anterior en el següent enllaç: [codepen.io/ioc-daw-m09/pen/WwMxWy](https://codepen.io/ioc-daw-m09/pen/WwMxWy).

Vegeu que la columna vermella l'hem empès cap a la dreta, i que la columna blava l'hem estirat cap a l'esquerra, i així les veiem en l'ordre invers.

2.3.4. Elements

El *framework* Bootstrap 3 proporciona estils per representar múltiples elements, com són la tipografia, les taules, les imatges, el formularis, els botons, etc. Totes les seves propietats es poden consultar a: [getbootstrap.com/css](https://getbootstrap.com/css).

Vegeu en la [taula 2.22](#) algunes classes definides per Bootstrap referents a la tipografia.

Taula: 2.22. Classes Bootstrap 3 per a la tipografia

Classe	Descripció
.lead	Per destacar un text.
.text-left	Alinea el text a l'esquerra.
.text-center	Centra el text.
.text-right	Alinea el text a la dreta.
.list-inline	Mostra els elements de la llista horitzontalment.
.list-unstyled	Per definir una llista sense vinyeta.

Vegeu en la [taula 2.23](#) algunes classes definides per Bootstrap referents a les taules.

Taula: 2.23. Classes Bootstrap 3 per a les taules

## DISSENY D'INTERFÍCIES WEB

<code>.table-striped</code>	Alterna el color de fons per millorar la llegibilitat dels continguts.
<code>.table-bordered</code>	Mostra les quatre vores de cada cel·la de la taula.
<code>.table-hover</code>	Modifica lleugerament l'aspecte de les files quan l'usuari passa el ratolí per damunt d'aquestes.
<code>.table-condensed</code>	Mostra els continguts de la taula de manera més compacta.

Vegeu en la [taula 2.24](#) algunes classes definides per Bootstrap per decorar les imatges.

**Taula: 2.24.** Classes Bootstrap 3 per a les imatges

Classe	Descripció
<code>.img-rounded</code>	Arrodoneix els vèrtexs de la imatge.
<code>.img-thumbnail</code>	Mostra la imatge amb un farciment blanc i una vora fina. Simulant l'apecte de fotografies antigues.
<code>.img-circle</code>	Converteix la imatge en un cercle.

Vegeu en la [taula 2.25](#) algunes classes definides per Bootstrap per als formularis.

**Taula: 2.25.** Classes Bootstrap 3 per als formularis

Classe	Descripció
<code>.form-control</code>	Estableix amplada <i>width</i> : 100% als elements <code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code> i <code>&lt;select&gt;</code> .
<code>.form-group</code>	Agrupa el camp del formulari amb el seu <code>&lt;label&gt;</code> .
<code>.form-horizontal</code>	Agrupa el camp del formulari amb el seu <code>&lt;label&gt;</code> horitzontalment.

A més a més, també inclou components `CSS` que cobreixen les necessitats més habituals: icones, menús desplegable, barres de navegació, barres de progrés, objectes multimèdia, etc.

Podeu consultar com utilitzar i incloure aquests components a: [getbootstrap.com/components](https://getbootstrap.com/components).

### Exemple de menú desplegable utilitzant Bootstrap 3

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Primera pàgina web</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
```



## DISSENY D'INTERFÍCIES WEB

```
<body>

  <div class="container">
    <h2>Menú desplegable</h2>
    <div class="dropdown">
      <button class="btn btn-info dropdown-toggle" type="button" data-
      <span class="caret"></span></button>
      <ul class="dropdown-menu">
        <li><a href="#">HTML</a></li>
        <li><a href="#">CSS</a></li>
        <li><a href="#">JavaScript</a></li>
        <li class="divider"></li>
        <li><a href="#">Qui som</a></li>
      </ul>
    </div>
  </div>

  <div class="container">
    <div class="row">
      <div class="jumbotron">
        <h1>Text destacat: programació web</h1>
        <p>Utilitza els principals llenguatges de programació: HTML,
        <p><a class="btn btn-primary btn-lg" role="button">Més...</a>
      </div>
    </div>
  </div>

</body>

</html>
```

El resultat serà el que es pot veure en la [figura 2.50](#).

**Figura 2.50.** Menú desplegable i jumbotron



L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-mo9/pen/LNJvKo](https://codepen.io/ioc-daw-mo9/pen/LNJvKo).

Cal tenir en compte les consideracions següents:

- Per crear un menú s'utilitza la classe `.dropdown` sobre l'etiqueta `<div>`.
- El botó `button` serà l'element que obre el menú. També podria ser un enllaç.
- Per donar estil al botó s'han utilitzat les classes `.btn` i `.btn-info`, les quals fan que el botó es vegi amb aquesta forma i color.
- Perquè el botó obri el menú cal utilitzar la classe `.dropdown-toggle` i l'atribut `data-toggle="dropdown"`.
- Perquè es vegi la fletxa que indica que s'obrirà un menú en fer clic al botó s'utilitza la classe `.caret`.
- Per crear el menú s'utilitza una llista `<ul>` amb la classe `.dropdown-menu`.
- Cada element del menú es representarà amb l'element `<li>`.
- Per crear el text destacat s'utilitza la classe que ofereix Bootstrap, anomenada `.jumbotron`.

## DISSENY D'INTERFÍCIES WEB

### Exemple de formulari utilitzant Bootstrap 3

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Primera pàgina web</title>
  <meta name="viewport" content="width=device-width, initial-scale=:
  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootst
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/
  <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bo
</head>

<body>

  <div class="container">
    <div class="row text-left">
      <div class="col-xs-6">
        <h2>Alta Usuari:</h2>
        <form role="form">
          <div class="form-group">
            <label for="nom_1">Nom</label>
            <input type="text" id="nom_1">
          </div>
          <div class="form-group">
            <label for="email_1">Correu electrònic</label>
            <input type="email" id="email_1">
          </div>
          <div class="form-group">
            <label for="contrase_1">Contrasenya</label>
            <input type="password" id="contrase_1">
          </div>
          <button type="submit" class="btn btn-primary">Enviar</button>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
```

El resultat es pot veure en la [figura 2.51](#).

**Figura 2.51.** Formulari amb Bootstrap

Primera pàgina web x

file:///pagFormulariBS.html

## Alta Usuari:

**Nom:**

**Correu electrònic:**

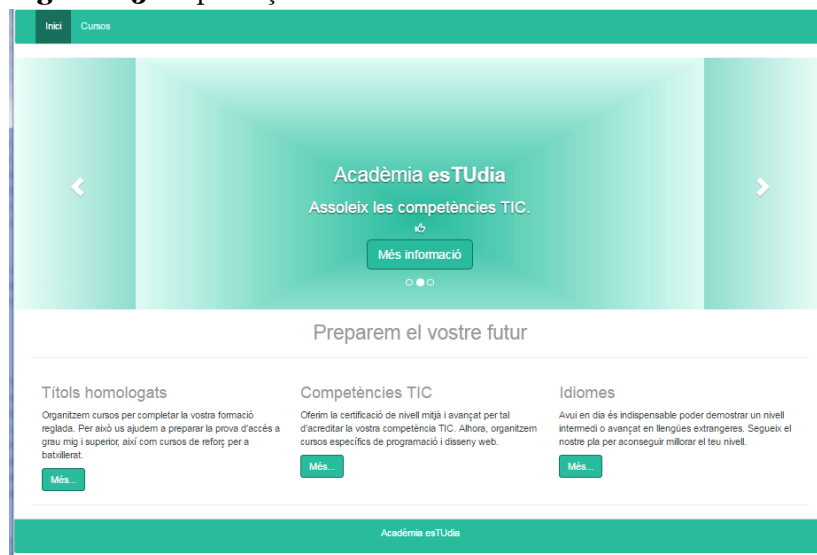
**Contrasenya:**

## 2.4. Cas pràctic amb Bootstrap: Acadèmia esTudia

## DISSENY D'INTERFÍCIES WEB

la [figura 2.52](#).

**Figura 2.52.** Aparència del lloc web d'esTUDia



Visualització en el navegador de la pàgina web inicial

Primerament, creareu la capçalera amb el menú de navegació utilitzant les classes `.navbar` i `.navbar-default`. Aquesta barra té l'identificador `bs-example-navbar-collapse-1` i mostrarà els ítems “Inici” i “Cursos”, però si la mida de la pantalla es redueix es reduirà en un sol botó.

El codi HTML serà com el de l'exemple:

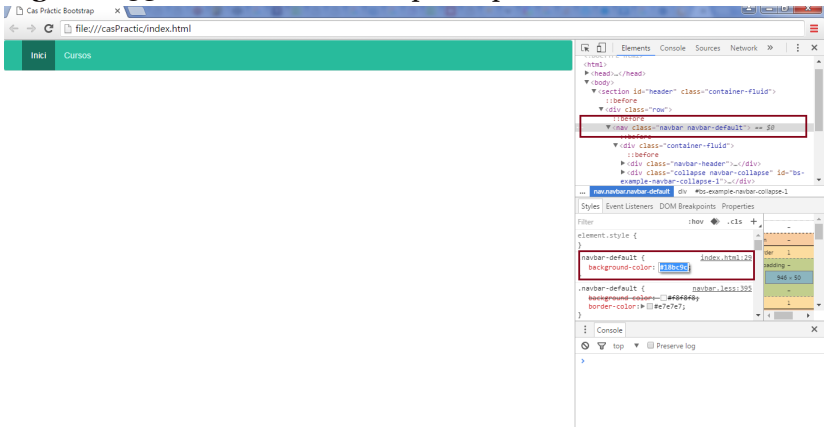
### Codi HTML per a la barra de navegació

```
<section id="header" class="container-fluid">
  <div class="row">
    <nav class="navbar navbar-default">
      <div class="container-fluid">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-toggle=
            <span class="sr-only">Menú</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
        </div>
        <div class="collapse navbar-collapse" id="bs-example-navbar-
          <ul class="nav navbar-nav">
            <li class="active"><a href="#">Inici</a></li>
            <li><a href="#seccio2">Cursos</a></li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
</section>
```

A més a més, també serà necessari modificar alguns estils CSS del *framework* Bootstrap perquè es vegi amb els colors escollits. Per tant, utilitzareu l'inspector del navegador, tal com es mostra en la [figura 2.53](#), per tal de localitzar els estils que cal modificar. Atès que no es recomana modificar el *framework*, serà necessari crear un

DISSENY D'INTERFÍCIES WEB

Figura 2.53. Visualització de l'inspector per cercar els estils



Inspector del navegador Chrome

Els estils CSS referents la barra de navegació que cal modificar per tal que la seva aparença sigui la desitjada són:

Estil per a la barra de navegació

```
.navbar-default {
  background-color: #18bc9c;
}
.navbar-default .navbar-nav > li > a {
  color: #DBF444;
}
.navbar-default .navbar-nav > li > a:hover,
.navbar-default .navbar-nav > li > a:focus {
  color: #fff;
}
.navbar-default .navbar-nav > .active > a,
.navbar-default .navbar-nav > .active > a:hover,
.navbar-default .navbar-nav > .active > a:focus {
  color: #fff;
  background-color: #0E705C;
}
```

Seguidament, per crear el carrusel amb les imatges caldrà tenir primer les imatges que mostrarà. En aquest cas sempre mostra la mateixa imatge amb el fons verd de 900 x 400 px.

Les classes que proporciona Bootstrap per crear el carrusel es poden veure en la [taula 2.26](#).

Taula: 2.26. Classes que proporciona Bootstrap per crear un carrusel

Classe	Descripció
.slide	Afegeix una transició <u>CSS</u> i efecte d'animació quan es passa d'un element al següent.
.carousel-indicator	Afegeix indicadors per al carrusel. Aquests són els petits punts a la part inferior de cada diapositiva.
.carousel-inner	Afegeix les diapositives al carrusel.
.item	Especifica el contingut de cada diapositiva.

## DISSENY D'INTERFÍCIES WEB

.right carousel-control	Afegeix un botó dret al carrusel, que permet a l'usuari tornar entre les diapositives.
.carousel-caption	Permet especificar un títol.

Així doncs, el codi per inserir el carrusel és:

### Codi HTML per inserir un carrusel

```
<section id="seccio1" class="container-fluid">
  <div class="row">
    <div id="myCarousel" class="carousel slide" data-ride="carousel">
      <!-- Indicadors -->
      <ol class="carousel-indicators">
        <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
        <li data-target="#myCarousel" data-slide-to="1"></li>
        <li data-target="#myCarousel" data-slide-to="2"></li>
      </ol>

      <!-- slides -->
      <div class="carousel-inner" role="listbox">
        <div class="item active">
          
          <div class="container">
            <div class="carousel-caption">
              <h2>Acadèmia <strong>esTUDia</strong></h2>
              <h3>Us facilitem l'obtenció de títols oficials.</h3>
              <p class="glyphicon glyphicon-thumbs-up"></p>
              <p><a class="btn btn-lg btn-primary" href="#seccio2">
                </div>
            </div>
          </div>
        </div>
        <div class="item">
          
          <div class="container">
            <div class="carousel-caption">
              <h2>Acadèmia <strong>esTUDia</strong></h2>
              <h3>Assoleix les competències TIC.</h3>
              <p class="glyphicon glyphicon-thumbs-up"></p>
              <p><a class="btn btn-lg btn-primary" href="#seccio2">
                </div>
            </div>
          </div>
        </div>
        <div class="item">
          
          <div class="container">
            <div class="carousel-caption">
              <h2>Acadèmia <strong>esTUDia</strong></h2>
              <h3>Millora la teva fluïdesa en llengües estrangeres.</h3>
              <p class="glyphicon glyphicon-thumbs-up"></p>
              <p><a class="btn btn-lg btn-primary" href="#seccio2">
                </div>
            </div>
          </div>
        </div>
      </div>

      <!-- controls a dreta i esquerra -->
      <a class="left carousel-control" href="#myCarousel" role="button">
        <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
        <span class="sr-only">Anterior</span>
      </a>
      <a class="right carousel-control" href="#myCarousel" role="button">
        <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
        <span class="sr-only">Següent</span>
      </a>
    </div>
  </div>
</section>
```

## DISSENY D'INTERFÍCIES WEB

### Estils per al carrusel

```
.carousel-inner > .item > img,
.carousel-inner > .item > a > img {
  width: 70%;
  margin: auto;
}
.carousel-control.right {
  background-image: linear-gradient(to right, #18bc9c 0%, #DBF4F4 100%);
}
.carousel-control.left {
  background-image: linear-gradient(to left, #18bc9c 0%, #DBF4F4 100%);
}
```

Després cal crear els articles, per a la qual cosa serà necessari crear un *grid* de dues files. En la primera s'hi localitza el títol, i en la segona els diferents articles, els quals, en funció de la mida de la pantalla del navegador, es mostraran en una, dues o tres columnes.

El codi HTML per inserir els articles del lloc web és:

### Codi HTML per inserir els articles

```
<section id="seccio2" class="container">
  <div class="row text-center">
    <h2>Preparem el vostre futur</h2>
    <hr class="small" />
  </div>

  <div class="row">
    <div class="col-xs-12 col-sm-6 col-md-4">
      <h3>Títols homologats</h3>
      <figure></figure>
      <p>Organitzem cursos per completar la vostra formació reglada.
      <a href="#" class="btn btn-primary">Més...</a>
    </div>

    <div class="col-xs-12 col-sm-6 col-md-4">
      <h3>Competències TIC</h3>
      <figure></figure>
      <p>Ofereixo la certificació de nivell mitjà i avançat per tal d
      <a href="#" class="btn btn-primary">Més...</a>
    </div>

    <div class="col-xs-12 col-sm-6 col-md-4">
      <h3>Idiomes</h3>
      <figure></figure>
      <p>Avui en dia és indispensable poder demostrar un nivell int
      <a href="#" class="btn btn-primary">Més...</a>
    </div>
  </div>

  <div class="row text-center">
    <hr class="small" />
  </div>
</section>
```

I els estils que cal modificar perquè els botons i el text es vegin del color adient són:

## DISSENY D'INTERFÍCIES WEB

### Estils per als botons i text del lloc web

```
.btn-primary {
  background-color: #18bc9c;
  border-color: #0E705C;
}

.btn-primary:hover, .btn-primary:active, .btn-primary:focus, .btn-!
  background-color: #0E705C;
  border-color: #041E18;
}

div[class*="row"] > h2, div[class*="col-"] > h3 {
  color: #999999;
}
```

Finalment, es crea el peu de pàgina utilitzant l'estil `.panel-footer`.

### Codi HTML per inserir el peu de pàgina

```
<section class="container-fluid">
  <div class="row">
    <footer class="panel-footer">
      <p class="text-center">Acadèmia esTUDia</p>
    </footer>
  </div>
</section>
```

I els estils que cal modificar són:

### Estil per modificar el color del peu de pàgina

```
.panel-footer {
  color: #fff;
  background-color: #18bc9c;
}
```

Així doncs, el codi resultant del document web utilitzant el *framework* Bootstrap serà:

### Codi del lloc web

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=:
  <title>Cas Pràctic Bootstrap</title>
```

## DISSENY D'INTERFÍCIES WEB

```

<!-- Estils que modifiquen i/o amplien el framework -->
<style>
.carousel-inner > .item > img,
.carousel-inner > .item > a > img {
  width: 70%;
  margin: auto;
}
.carousel-control.right {
  background-image:linear-gradient(to right, #18bc9c 0%, #DBFBF4);
}
.carousel-control.left {
  background-image:linear-gradient(to left, #18bc9c 0%, #DBFBF4);
}

.navbar-default {
  background-color: #18bc9c;
}
.navbar-default .navbar-nav > li > a {
  color: #DBFBF4;
}
.navbar-default .navbar-nav > li > a:hover,
.navbar-default .navbar-nav > li > a:focus {
  color: #fff;
}
.navbar-default .navbar-nav > .active > a,
.navbar-default .navbar-nav > .active > a:hover,
.navbar-default .navbar-nav > .active > a:focus {
  color: #fff;
  background-color: #0E705C;
}

.btn-primary {
  background-color:#18bc9c;
  border-color: #0E705C
}
.btn-primary:hover, .btn-primary:active , .btn-primary:focus, .!
  background-color:#0E705C;
  border-color: #041E18;
}

.panel-footer {
  color: #fff;
  background-color: #18bc9c;
}

div[class*="row"] > h2, div[class*="col-"] > h3 {
  color:#999999;
}

</style>
</head>
<body>

<section id="header" class="container-fluid">
  <div class="row">
    <nav class="navbar navbar-default">
      <div class="container-fluid">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-toggle="(
            <span class="sr-only">Menú</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
        </div>
        <div class="collapse navbar-collapse" id="bs-example-navbar-
          <ul class="nav navbar-nav">
            <li class="active"><a href="#">Inici</a></li>
            <li><a href="#seccio2">Cursos</a></li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
</section>

<section id="seccio1" class="container-fluid">
  <div class="row">
    <div id="myCarousel" class="carousel slide" data-ride="carousel"
    <!-- Indicators -->
    <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="acti
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
    </ol>

    <!-- slides -->
    <div class="carousel-inner" role="listbox">

```



## DISSENY D'INTERFÍCIES WEB

```

        <h2>Acadèmia <strong>esTUDia</strong></h2>
        <h3>Us facilitem l'obtenció de títols oficials.</h3>
        <p class="glyphicon glyphicon-thumbs-up"></p>
        <p><a class="btn btn-lg btn-primary" href="#seccio2" >
    </div>
</div>
</div>

<div class="item">
    
        <div class="carousel-caption">
            <h2>Acadèmia <strong>esTUDia</strong></h2>
            <h3>Assoleix les competències TIC.</h3>
            <p class="glyphicon glyphicon-thumbs-up"></p>
            <p><a class="btn btn-lg btn-primary" href="#seccio2"
        </div>
    </div>
</div>

<div class="item">
    
        <div class="carousel-caption">
            <h2>Acadèmia <strong>esTUDia</strong></h2>
            <h3>Millora la teva fluència en llengües estrangeres.
            <p class="glyphicon glyphicon-thumbs-up"></p>
            <p><a class="btn btn-lg btn-primary" href="#seccio2"
        </div>
    </div>
</div>

<!-- Controls a dreta i esquerra -->
<a class="left carousel-control" href="#myCarousel" role="bu
    <span class="glyphicon glyphicon-chevron-left" aria-hidde
    <span class="sr-only">Anterior</span>
</a>
<a class="right carousel-control" href="#myCarousel" role="l
    <span class="glyphicon glyphicon-chevron-right" aria-hidde
    <span class="sr-only">Següent</span>
</a>
</div>
</div>
</div>
</section>

<section id="seccio2" class="container">
    <div class="row text-center">
        <h2>Preparem el vostre futur</h2>
        <hr class="small" />
    </div>
    <div class="row">
        <div class="col-xs-12 col-sm-6 col-md-4">
            <h3>Títols homologats</h3>
            <figure></figure>
            <p>Organitzem cursos per completar la vostra formació reglada.
            <a href="#" class="btn btn-primary">Més...</a>
        </div>
        <div class="col-xs-12 col-sm-6 col-md-4">
            <h3>Competències TIC</h3>
            <figure></figure>
            <p>Ofereim la certificació de nivell mitjà i avançat per tal d'a
            <a href="#" class="btn btn-primary">Més...</a>
        </div>
        <div class="col-xs-12 col-sm-6 col-md-4">
            <h3>Idiomes</h3>
            <figure></figure>
            <p>Avui en dia és indispensable poder demostrar un nivell inte
            <a href="#" class="btn btn-primary">Més...</a>
        </div>
    </div>
    <div class="row text-center">
        <hr class="small" />
    </div>
</section>

<section class="container-fluid">
    <div class="row">
        <footer class="panel-footer">
            <p class="text-center">Acadèmia esTUDia</p>
        </footer>
    </div>
</section>

</body>
</html>

```

**DISSENY D'INTERFÍCIES WEB**