

M0373. Llenguatges de marques – BA 1 - CSS

INSTITUT PÚBLIC DE SABADELL

Marc Albareda

Eloi Vazquez

INDEX

- En aquest PPT veurem:
 - Introducció CSS

CSS

- CSS (Cascading Style Sheets):
 - És un llenguatge de disseny senzill que pretén simplificar el procés de presentació de les pàgines web.
 - Permet aplicar estils a les pàgines web independentment de l'HTML que conforma cada pàgina web.



CSS

- Perquè CSS?
 - Estalvia temps
 - Pots escriure un estil i reaprofitar-lo per altres pàgines.
 - Fàcil manteniment
 - Una modificació al CSS i totes les pàgines queden actualitzades a la mateixa vegada.
 - Motors de cerca
 - Fàcil lectura pels motors de cerca del contingut □ CSS és una tècnica neta de codificació.
 - Estil superior a HTML
 - Gamma d'atributs més àmplia que HTML.
 - Navegació offline
 - Pot emmagatzemar aplicacions web de forma local amb l'ajut de la memòria cau fora de línia. Després es poden veure offline.

CSS

- Hi ha 3 tipus de CSS:
 - Estil en línia
 - És un mètode per afegir el llenguatge d'estil de pàgina directament, dins d'una etiqueta HTML.
 - Aquesta manera de fer no és totalment adequada perquè descripció del format dins del document de la pàgina web a nivell de codi es converteix en una feina llarga, costosa i poc elegant.
 - Permet controlar l'estil de diverses pàgines web.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Inline CSS</title>
  </head>

  <body>
    <p style="color:#000990; font-size:50px;
      font-style:italic; text-align:center;">
      Donkey Monkey
    </p>
  </body>

</html>
```

CSS

- Hi ha 3 tipus de CSS (Continuació):
 - Full d'estil intern
 - És un full d'estil que està incrustat dins d'un document HTML.
 - Va a la dreta de l'element <head>.
 - D'aquesta manera s'obté el benefici de separar la informació de l'estil, del codi HTML pròpiament dit.
 - S'utilitza quan es vol proporcionar alguna característica a una pàgina web en un simple fitxer.

CSS

- Hi ha 3 tipus de CSS (Continuació):
 - Full d'estil intern. Exemple:

Codi CSS incrustat.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Internal CSS</title>
    <style>
      .main {
        text-align: center;
      }

      .GFG {
        color: #000090;
        font-size: 50px;
        font-weight: bold;
      }

      .geeks {
        font-style: bold;
        font-size: 20px;
      }
    </style>
  </head>

  <body>
    <div class="main">
      <div class="GFG">Donkey Monkey</div>

      <div class="geeks">
        The best arcade game
      </div>
    </div>
  </body>

</html>
```

CSS

- Hi ha 3 tipus de CSS (Continuació):
 - Full d'estil extern:
 - És un full d'estil que està emmagatzemat en un arxiu diferent a l'arxiu on s'emmagatzema el codi HTML de la pàgina web.
 - Aquesta és la manera de programar més potent □ separa completament les regles de format per la pàgina HTML de l'estructura bàsica de la pàgina.

CSS

- Hi ha 3 tipus de CSS (Continuació):
 - Full d'estil extern. Exemple:

```
<!DOCTYPE html>
<html>
```

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

```
<body>
  <div class="main">
    <div class="GFG">Donkey Monkey</div>
    <div id="geeks">
      The best arcade game
    </div>
  </div>
</body>
</html>
```

```
1  body {
2    background-color: powderblue;
3  }
4
5  .main {
6    text-align: center;
7  }
8
9  .GFG {
10   color: #000090;
11   font-size: 50px;
12   font-weight: bold;
13 }
14
15 #geeks {
16   font-style: bold;
17   font-size: 20px;
18 }
19
```

Donkey Monkey

The best arcade game

CSS

- En el cas que hi hagi més d'un estil a la pàgina web:
 - Estil en línia té més prioritat que la resta.
 - Full d'estil intern té més prioritat que el full d'estil extern.

CSS

- Sintaxi CSS:

- Inclou un conjunt de regles d'estil les quals consisteix amb un selector i un bloc de declaracions.

Selector => `h1`

El selector apunta a l'element HTML a formatjar.

Declaration => `{color:blue;font size:12px;}`

Les declaracions acaben amb punt i coma “;”.

El bloc de declaracions conté una o més declaracions separades per “;”.

Cada declaració inclou el nom de la propietat i el seu valor.

CSS

- Exemple de sintaxi:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple</title>
  </head>
  <body>
    <main>
      <p>Hello World!</p>
      <p>Exemple d'estil amb CSS</p>
    </main>
  </body>
</html>
```

HTML

```
p {
  color: blue;
  text-align: center;
}
```

Selector

Declaració

CSS

Hello World!

Exemple d'estil amb CSS

Resultat

CSS

- En resum, la sintaxi és:



CSS

- Tipus de selectors:
 - Selector universal
 - el selector universal coincideix simplement amb el nom de qualsevol tipus d'element. S'utilitza l'asterisc, “*”.
 - Selector element
 - selecciona elements en funció del nom de l'element. Un exemple és l'anterior element.
 - Selector descendent o combinació
 - S'aplica una regla d'estil a un element concret només quan es troba dins d'un element concret.

CSS

- Tipus de selectors:

- ID selector

- utilitza l'atribut id d'un element HTML per seleccionar un element específic.
 - id d'un element ha de ser únic dins d'una pàgina.
 - selecciona un element amb un identificador específic, escriu un caràcter hash (#), seguit de l'identificador de l'element.
 - el veritable poder dels selectors d'identificació és quan s'utilitzen com a fonament per als selectors descendents.

CSS

- Tipus de selectors:
 - Selector classe
 - selecciona elements amb un atribut de classe específic.
 - selecciona elements amb una classe específica, escriuiu un caràcter de punt (.), seguit del nom de la classe.
- Els selectors es poden agrupar mitjançant una ‘,’:

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```


CSS

- Exemple de selector universal:

```
<!DOCTYPE html>
<html>

  <head>
    <title>Example</title>
  </head>

  <body>
    <main>
      <p>Hello World!</p>
      <p>Exemple d'estil amb CSS</p>
    </main>
  </body>

</html>
```

```
CSS ▼
1  * {
2    color: blue;
3    text-align: center;
4  }
```

Selector universal.

Hello World!

Exemple d'estil amb CSS

CSS

- Exemple de selector element:
 - Es pot trobar a la diapositiva que parla de [sintaxi](#).

CSS

- Exemple de selector de combinació:

Selector	Exemple	
element element	div p	Selecciona tots els <p> dintre de div.
element>element	div>p	Selecciona tots els <p> i que el seu pare és div.
element+element	div+p	Selecciona tots els <p> precedits de div.
element~element	p~ul	Selecciona tots els elements ul precedits d'un element p.

CSS

- Exemple de selector de combinació (continuació):

```
<!DOCTYPE html>
<html>

  <head>

  </head>

  <body>

    <div>
      <p>Paragraph 1 in the div.</p>
      <p>Paragraph 2 in the div.</p>
      <section>
        <p>Paragraph 3 in the div.</p>
      </section>
    </div>

    <p>Paragraph 4. Not in a div.</p>
    <p>Paragraph 5. Not in a div.</p>

  </body>

</html>
```

```
div p {
  background-color: yellow;
}
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

Són els elements
que compleixen la
combinació.

CSS

- Exemple de Selector ID:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <p id="para1">Hello World!</p>
  <p>This paragraph is not affected by the style.</p>
</body>
</html>
```

```
#para1 {
  text-align: center;
  color: red;
}
```

Hello World!

This paragraph is not affected by the style.

CSS

- Exemple de Selector Classe:

```
<!DOCTYPE html>
<html>

  <head>
  </head>

  <body>

    <h1 class="center">Red and center-aligned heading</h1>
    <p class="center">Red and center-aligned paragraph.</p>

  </body>
</html>
```

```
.center {
  text-align: center;
  color: red;
}
```

Red and center-aligned heading

Red and center-aligned paragraph.

CSS

- Per afegir un fitxer CSS a un HTML, l'estructura és la següent:

```
<link rel="stylesheet" href="style.css">
```

CSS

- Un exemple:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Fitxer HTML

mystyle.css

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```


CSS

- Els comentaris en CSS van de la següent manera:
 - El text va entre /* i */
- Per exemple:

```
/* content */
```

```
/* This is a single-line comment */
```

```
p {  
  color: red;  
}
```

Comentari

CSS

- Colors:
 - Aquesta propietat ens permet donar color als diferents components de la pàgina web o document.
 - Els colors es poden definir de la següent manera:
 - Noms de colors predefinits de colors els podeu trobar al següent enllaç:
 - https://www.w3schools.com/colors/colors_names.asp
 - RGB
 - Expressar mitjançant la fórmula rgb(vermell, verd i blau)
 - RGBA
 - Expressar mitjançant la fórmula rgba(vermell, verd, blau i transparent)

CSS

- Colors:
 - Els colors es poden definir de la següent manera:
 - Notació hexadecimal
 - La notació hexadecimal comença pel símbol # i 6 caràcters que van des del 0 fins a F.
 - HSL
 - Defineix un model de color depenen de la tonalitat, saturació i luminància.
 - HSLA
 - Defineix un model HSL juntament amb la transparència igual que RGBA.

CSS

- Exemple de colors:

Exemple de color predefinit

Exemple de color rgb

Exemple de color rgba

Exemple de color hexadecimal

Exemple de color HSL

Exemple de color HSLA

```
#colorcreat {  
  color: grey;  
  text-align: center;  
}
```

```
#colorrgb {  
  color: rgb(192, 192, 192);  
  text-align: center;  
}
```

```
#colorrgba {  
  color: rgba(192, 192, 192, 0.5);  
  text-align: center;  
}
```

```
#colorhex {  
  color: #d3d3d3;  
  text-align: center;  
}
```

```
#colorhsl {  
  color: hsl(0, 0%, 50%);  
  text-align: center;  
}
```

```
#colorhsla {  
  color: hsla(0, 0%, 50%, 0.50);  
  text-align: center;  
}
```

```
<html>  
  <head>  
    <title>CSS color property</title>  
    <style>  
  
    </style>  
  </head>  
  <body>  
    <h1 id="colorcreat">  
      Witcher  
    </h1>  
  
    <h1 id="colorrgb">  
      Witcher  
    </h1>  
  
    <h1 id="colorrgba">  
      Witcher  
    </h1>  
  
    <h1 id="colorhex">  
      Witcher  
    </h1>  
  
    <h1 id="colorhsl">  
      Witcher  
    </h1>  
  
    <h1 id="colorhsla">  
      Witcher  
    </h1>  
  </body>  
</html>
```

CSS

- Enllaços

- Els enllaços poden tenir propietats addicionals
 - hover. Com s'actua quan es passa el ratolí per sobre (no clicar)
 - link. Com s'actua quan l'enllaç està sense visitar
 - visited. com s'actua quan l'enllaç ha estat visitat
 - active. Com s'actua quan l'enllaç està actiu
- El hover es pot utilitzar en moltes altres coses apart d'enllaços!

```
.cosa{  
  height: 100px;  
  width: 100px;  
  background-color: red;  
}  
/*hover es fa servir sovint amb divs apart de links*/  
.cosa:hover{  
  background-color: blue;  
}
```

```
/*mai s'aplicara perquè sempre està en un dels 4 estats*/  
a{  
  color: hotpink;  
}  
/* link no visitat */  
a:link {  
  color: blue;  
  font-family: Comic Sans MS;  
}  
/* link visitat */  
a:visited {  
  color: green;  
}  
/* link amb el ratolí a sobre */  
a:hover {  
  color: red;  
  font-family: Arial;  
}  
/* link seleccionat */  
a:active {  
  color: yellow;  
}
```

CSS

- Fons:
 - S'utilitzen aquestes propietats per definir els efectes de fons de diversos elements.
 - Les propietats de fons que veurem són les següents:
 - background-color
 - background-imatge
 - background-repeat
 - background-attachment
 - background-position

CSS

- Fons:
 - background-color
 - S'utilitza per donar color al fons d'un element.
 - A part del color també es pot utilitzar la transparència ja vista a RGBA o HSLA. Per la resta es pot utilitzar la propietat opacity.
 - background-imatge
 - Especifica una imatge per utilitzar com un fons d'un element.

CSS

- Fons:

- background-repeat:

- Per defecte, la propietat de la imatge de fons repeteix una imatge tant horitzontalment com verticalment.
 - Però algunes imatges només es poden repetir horitzontalment o verticalment.
 - Les propietats són:
 - background-repeat ☐ Indica com s'ha de repetir una imatge.
 - background-position ☐ s'utilitza per especificar la posició de la imatge de fons.

CSS

- Fons:
 - background-attachment:
 - S'utilitza per indicar si la imatge s'ha de desplaçar o corregir.
 - Per escurçar el codi, també és possible especificar totes les propietats de fons en una sola propietat. Això s'anomena propietat abreujada.
 - background ☐ és la propietat que s'agrupa totes les propietats.

CSS

- Exemple de fons:

```
<!DOCTYPE html>
<html>

  <head>

  </head>

  <body>

    <h1>Hello World!</h1>

    <p>This page has a light blue background color and opacity!
  </p>

  </body>

</html>
```

```
body {
  background-color: lightblue;
  opacity: 0.3;
}
```

Hello World!

This page has a light blue background color!

CSS

- Autre exemple de fons:

```
<!DOCTYPE html>
<html>

  <head>

</head>

  <body>

    <h1>Hello World!</h1>

    <p>This page has a light blue background color!</p>

  </body>

</html>
```

```
body {
  background-color: #ffff80;
  background-image: url("https://www.w3schools.com/css/img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

Hello World!

This page has a light blue background color!



CSS

- Vores:
 - Permet especificar les línies que es troben al límit dels elements.
 - Té diferents propietats:
 - border-style
 - border-width
 - border-color
 - border-top-style
 - border-right-style
 - border-bottom-style
 - border-left-style
 - border
 - border-radius

CSS

- Vores:
 - border-style
 - Especifica el tipus de vora a mostrar.
 - Els diferents tipus són:
 - dotted
 - dashed
 - solid
 - double
 - groove
 - ridge
 - inset
 - outset
 - none
 - hidden

CSS

- Vores:

- border-style:

- Se li podria assignar fins a 4 valors. A continuació es mostren els casos amb des de 4 valors fins a 1:

- 4 Valors. L'ordre és el següent:

- 1. top border
 - 2. right border
 - 3. bottom border
 - 4. left border

```
p {  
  border-style: dotted solid double dashed;  
}
```

És el mateix

```
p {  
  border-top-style: dotted;  
  border-right-style: solid;  
  border-bottom-style: dotted;  
  border-left-style: solid;  
}
```

CSS

- Vores:
 - border-style:
 - 3 Valors. L'ordre és el següent:
 - 1. top border
 - 2. right i left borders
 - 3. bottom border
 - 2 Valors. L'ordre és el següent:
 - 1. top i bottom borders
 - 2. right i left borders
 - 1 Valor. L'ordre és el següent:
 - Totes les vores

CSS

- Exemple de vore:

```
<!DOCTYPE html>
<html>

  <head>
  </head>

  <body>

    <h2>The border-style Property</h2>
    <p>This property specifies what kind of border to display:</p>

    <p class="dotted">A dotted border.</p>
    <p class="dashed">A dashed border.</p>
    <p class="solid">A solid border.</p>
    <p class="double">A double border.</p>
    <p class="groove">A groove border.</p>
    <p class="ridge">A ridge border.</p>
    <p class="inset">An inset border.</p>
    <p class="outset">An outset border.</p>
    <p class="none">No border.</p>
    <p class="hidden">A hidden border.</p>
    <p class="mix">A mixed border.</p>

  </body>
</html>
```

```
p.dotted {
  border-style: dotted;
}

p.dashed {
  border-style: dashed;
}

p.solid {
  border-style: solid;
}

p.double {
  border-style: double;
}

p.groove {
  border-style: groove;
}
```

```
p.ridge {
  border-style: ridge;
}

p.inset {
  border-style: inset;
}

p.outset {
  border-style: outset;
}

p.none {
  border-style: none;
}

p.hidden {
  border-style: hidden;
}
```

```
p.mix {
  border-style: dotted dashed solid double;
}
```


CSS

- Exemple de vores (continuació):

The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

CSS

- Exemple de viores:

```
<!DOCTYPE html>
<html>

  <head>
  </head>

  <body>

    <h2>Individual Border Sides</h2>
    <p class="four">4 different border styles.</p>
    <p class="three">3 different border styles.</p>
    <p class="two">2 different border styles.</p>
    <p class="one">1 border style.</p>

  </body>

</html>
```

```
body {
  text-align: center;
}

/* Four values */
p.four {
  border-style: dotted solid double dashed;
}

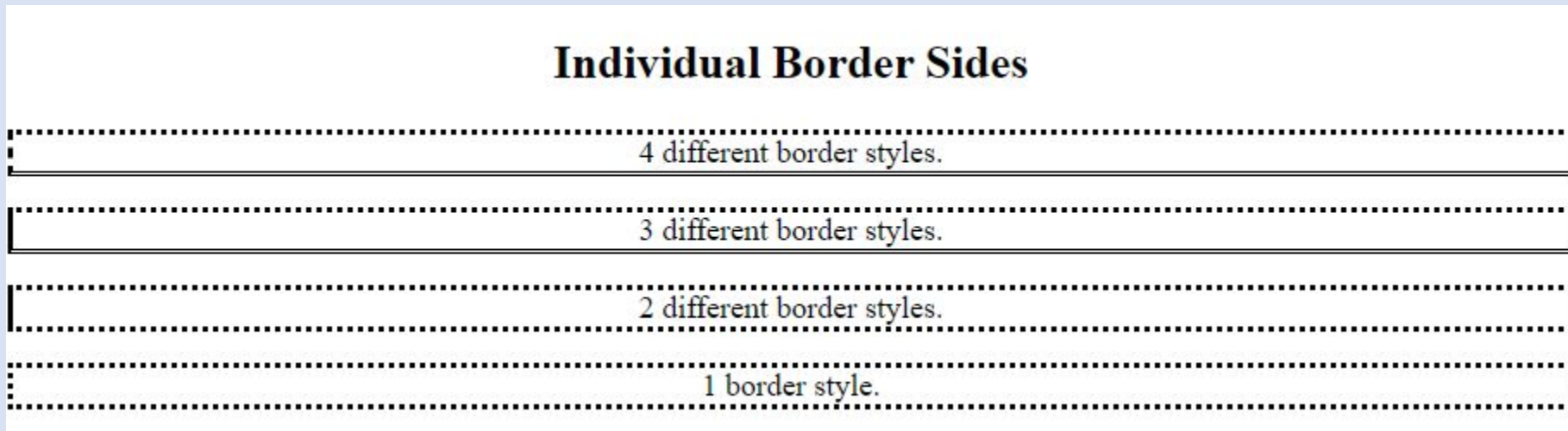
/* Three values */
p.three {
  border-style: dotted solid double;
}

/* Two values */
p.two {
  border-style: dotted solid;
}

/* One value */
p.one {
  border-style: dotted;
}
```

CSS

- Exemple de vores (continuació):



CSS

- Vores:
 - border-width:
 - Especifica l'amplada dels 4 vores.
 - També es poden especificar per cada vora de la mateixa manera que l'estil:

```
p.three {  
  border-style: solid;  
  border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and 35px left */  
}
```

CSS

- Vores:

- border-color:

- Indica el color per les 4 vores.

- border-top, border-right, border-bottom i border-left especifica l'estil de la vora de manera independent. També es pot especificar l'amplada i color

- border:

- Agrupa 3 propietats: border-width, border-style i border-color.

- border-radius:

- S'utilitza per arrodonir les vores.

- Totes les propietats de les vores els podeu trobar a:

- https://www.w3schools.com/css/css_border_rounded.asp

```
p {  
  border-left: 6px solid red;  
  background-color: lightgrey;  
}
```

```
p {  
  border: 5px solid red;  
}
```

CSS

- Exemple de Vores:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>The border Property</h2>
<p>Diferents exemples de vora:</p>

<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>

</body>
</html>
```

The border Property

Diferents exemples de vora:

Some text.

Some text.

Some text.

```
p.one {
  border-style: solid;
  border-width: 5px 20px; /* 5px top and bottom,
  20px on the sides */
  border-color: red;
}

p.two {
  border: 5px dotted lightblue;
}

p.three {
  border-style: double;
  border-width: 25px 10px 4px 35px; /* 25px top,
  10px right, 4px bottom and 35px left */
  border-color: Gold;
  border-radius: 10px;
}
```

CSS

- Marges:
 - S'utilitza per crear un espai al voltant dels elements, fora de les vores definides.
 - Les propietats que té són:
 - margin: Estableix els marges dels 4 costats. Funciona igual que les [vores](#).
 - margin-top, margin-right, margin-bottom i margin-left: Estableixen el marge al seu costat de la vora.

CSS

- Marges interiors (Padding):
 - S'utilitza per generar espai al voltant del contingut de l'element, dintre de les vores definides.
 - El funcionament és igual que els marges.
 - Les propietats són les següents amb el mateix funcionament dels marges:
 - padding
 - padding-top
 - padding-right
 - padding-bottom
 - Padding-left

CSS

- Exemple de marges:

```
<!DOCTYPE html>
<html>

  <head>
  </head>

  <body>

    <h2>The margin and padding shorthand property - 4 values</h2>

    <div>Quadre per comprovar el funcionament de margin i padding.</div>

  </body>

</html>
```

```
div {
  border: 1px solid black;
  margin: 50px 10px 10px 50px;
  padding: 25px 50px 75px 100px;
  background-color: lightblue;
}
```

The margin and padding shorthand property - 4 values

margin

padding

Quadre per comprovar el funcionament de margin i padding.

CSS

- Amplada i alçada (width i height):
 - Són propietats per indicar l'alçada i l'amplada dels elements.
 - max-width i max-height indiquen l'amplada o l'alçada màxima de l'element. En canvi, min-width i min-height és per indicar l'amplada o l'alçada mínima dels elements.

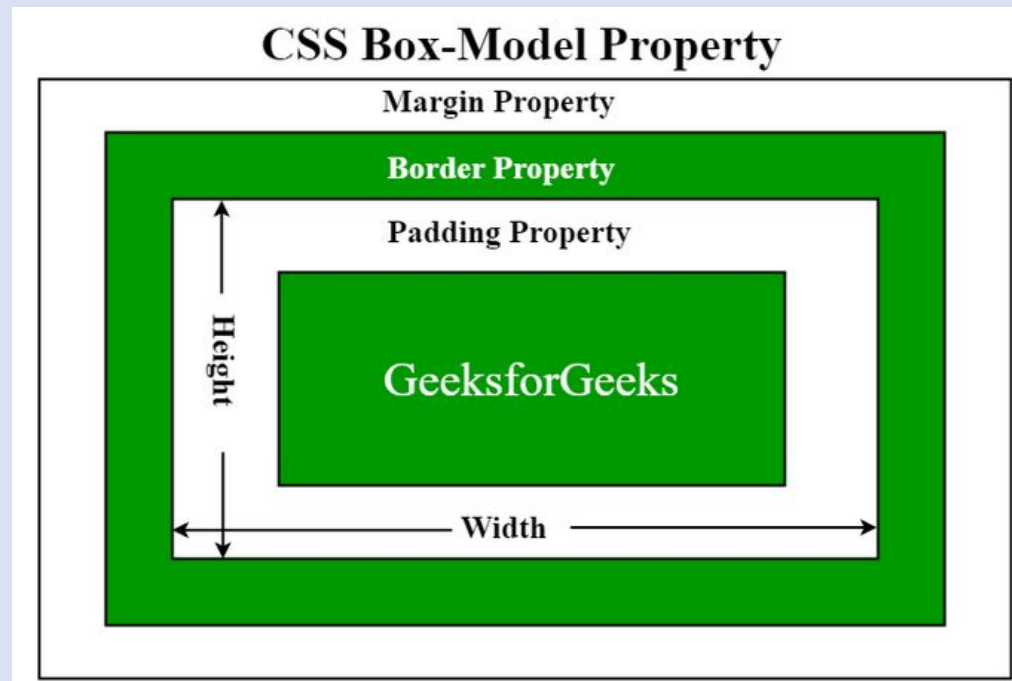
CSS

- Amplada i alçada (width i height):
 - La manera d'assignar els valor és la següent:
 - auto □ Per defecte. Ho calcula el navegador.
 - length □ Defineix la propietat amb la mesura de px o cm etc.
 - % □ Defineix la propietat amb percentatge.
 - initial □ Indica a la propietat el seu valor per defecte.
 - inherit □ Hereta els valors d'un antecessor.

CSS

- **BOX MODEL:**

- S'utilitza quan es fa referència del disseny i maquetació (layout).
- Box model li dona capes, una sobre l'altre dels diferents elements d'HTML, tal com il·lustra la següent imatge:



CSS

- BOX MODEL:

- Cada element HTML, com ara divs, paràgrafs (p), encapçalaments (h1, h2, etc.) i altres, es considera una "caixa" en el context del Box Model. Aquestes capes s'apilen unes sobre les altres per formar el contingut visual de la pàgina web.
- Un dels problemes amb el model de capses que hi ha és que l'amplada depèn dels marges i la vora. Tant l'amplada com l'alçada es calculen de la següent manera:
 - $\text{width} + \text{padding} + \text{border} = \text{amplada actual}$

CSS

- Box Model:
 - Així doncs, l'amplada o l'alçada és més gran del previst. Per solucionar aquest problema hi ha l'element box-sizing que inclou tant l'amplada o alçada el padding (marge intern) i border (vora):
 - box-sizing té 2 opcions:
 - "content-box" (valor predeterminat) → calcula l'amplada i l'alçada sense tenir en compte el "padding" i la vora.
 - "border-box" → calcula l'amplada i l'alçada i inclou el "padding" i la vora en el càlcul de l'amplada i l'alçada.

Només cal copiar això

```
* {  
  -webkit-box-sizing: border-box;  
  -moz-box-sizing: border-box;  
  box-sizing: border-box;  
}
```

CSS

- Exemple de box model:

```
<!DOCTYPE html>
<html>

  <head>
  </head>

  <body>

    <h2>Demonstrating the Box Model</h2>

    <p>Aquí afegeixo un text per diferencia l'element que es troba a sota.</p>

    <div>Aquest exemple és per comprovar com hi ha capes una sobre l'altre per cada element HTML. Consisteix amb vores, marges, marges interns i el contingut actual que pot ser tant un text com una imatge.</div>

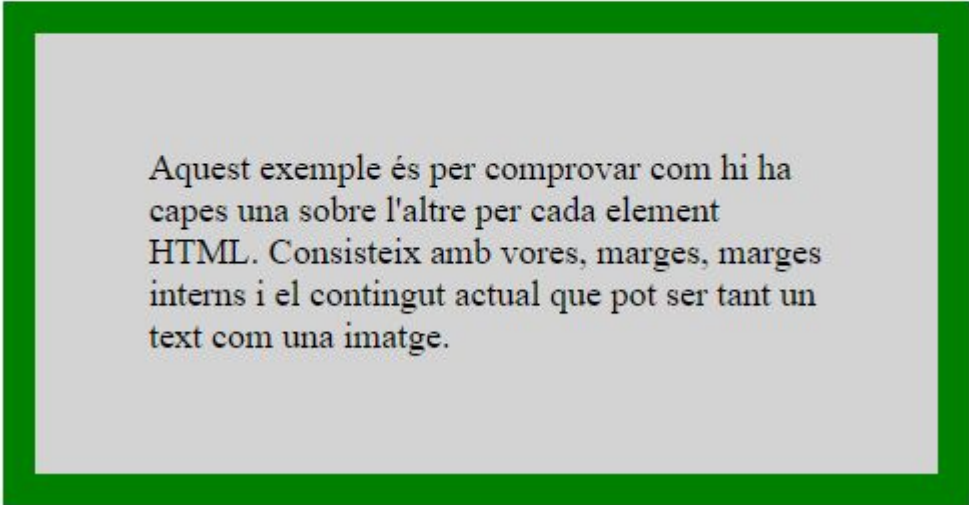
  </body>

</html>
```

```
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

Demonstrating the Box Model

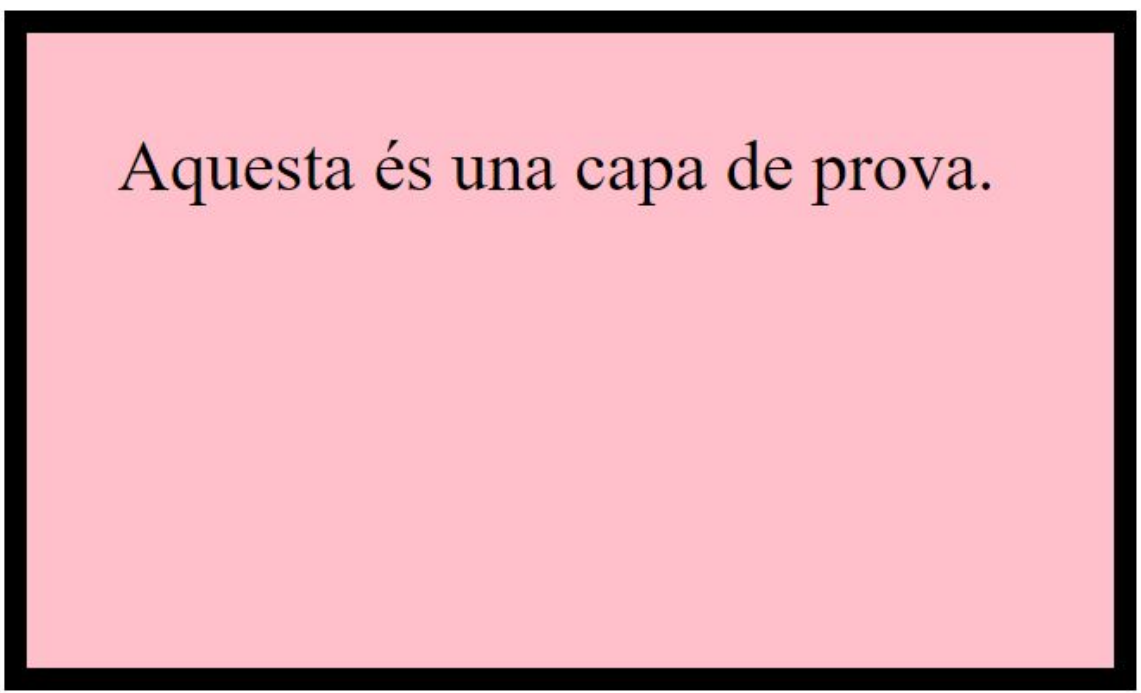
Aquí afegeixo un text per diferencia l'element que es troba a sota.



Aquest exemple és per comprovar com hi ha capes una sobre l'altre per cada element HTML. Consisteix amb vores, marges, marges interns i el contingut actual que pot ser tant un text com una imatge.

CSS

- Exemple de box model 2:



Aquesta és una capa de prova.

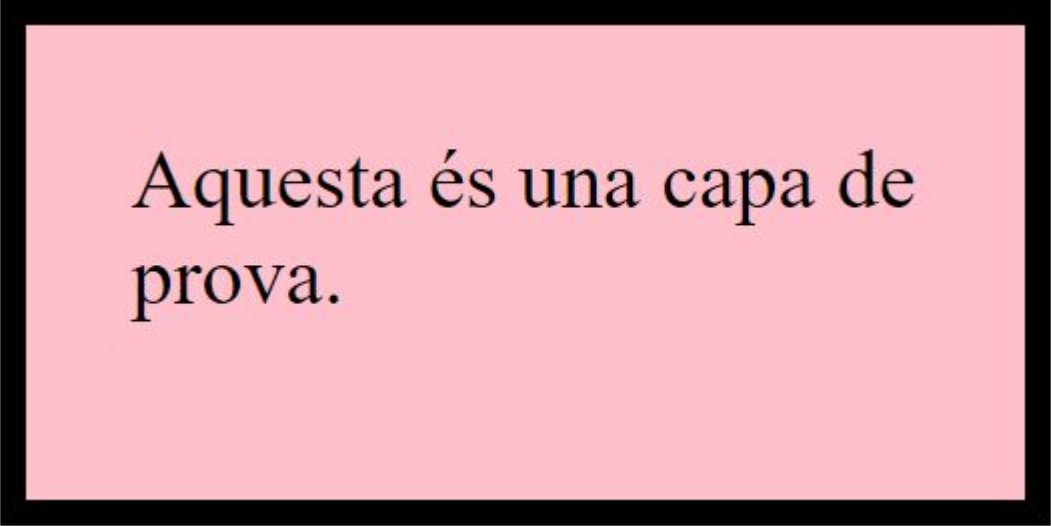
```
#caixa {  
    width: 200px;  
    height: 100px;  
    padding: 20px;  
    border: 5px solid black;  
    background-color: pink;  
}
```


CSS

- Exemple de box model 2 (cont.):
 - En aquest exemple, tenim una única capa (div) amb la classe "caixa". Hem definit les següents propietats:
 - width: Amplada de la capa = 200px.
 - height: Alçada de la capa = 100px.
 - padding: Espai interior de la capa = 20px.
 - border: Vora de la capa = 5px de gruix, de color negre.
 - background-color: Color rosa de fons de la caixa.
 - Aplicant el concepte de Box Model, l'amplada total de la caixa:
 - width + padding-left + padding-right + border-left-width + border-right-width
 - = 200px + 20px + 20px + 5px + 5px = 250px
 - I l'alçada de la caixa, es calcula de manera semblant, substituint right i left per top i bottom i ens donaria 150 px.

CSS

- Exemple de box model 2 (cont.):
 - Si volem una alçada i amplada de 100px i 200px, s'ha de canviar la propietat box-sizing a border-box.



Aquesta és una capa de
prova.

```
#caixa {  
    width: 200px;  
    height: 100px;  
    padding: 20px;  
    border: 5px solid black;  
    background-color: pink;  
    box-sizing: border-box;  
}
```

CSS

- Display (Visualització):
 - Indica com un element s'ha de visualitzar.
 - Els tipus de visualització que hi ha són:
 - Bloc (block)
 - En línia (inline)
 - Cap (none)
 - Altres

CSS

- Display i block:

- Els elements adjacents, els que en la declaració de l'HTML es troben abans i a continuació, apareixen adalt i a sota de l'element.
- La separació entre aquests elements és un salt de línia.
- Els elements en bloc intenten abarcar tota l'amplada de la pàgina.



CSS

- Display i block:
 - Els exemples més típics són:
 - `<div>`
 - `<h1>` - `<h6>`
 - `<p>`
 - `<form>`
 - `<header>`
 - `<footer>`
 - `<section>`
 - Els elements en bloc com "div", "p", "form", etc., per defecte comencen en una nova línia.

CSS

- Display i block (Exemple):

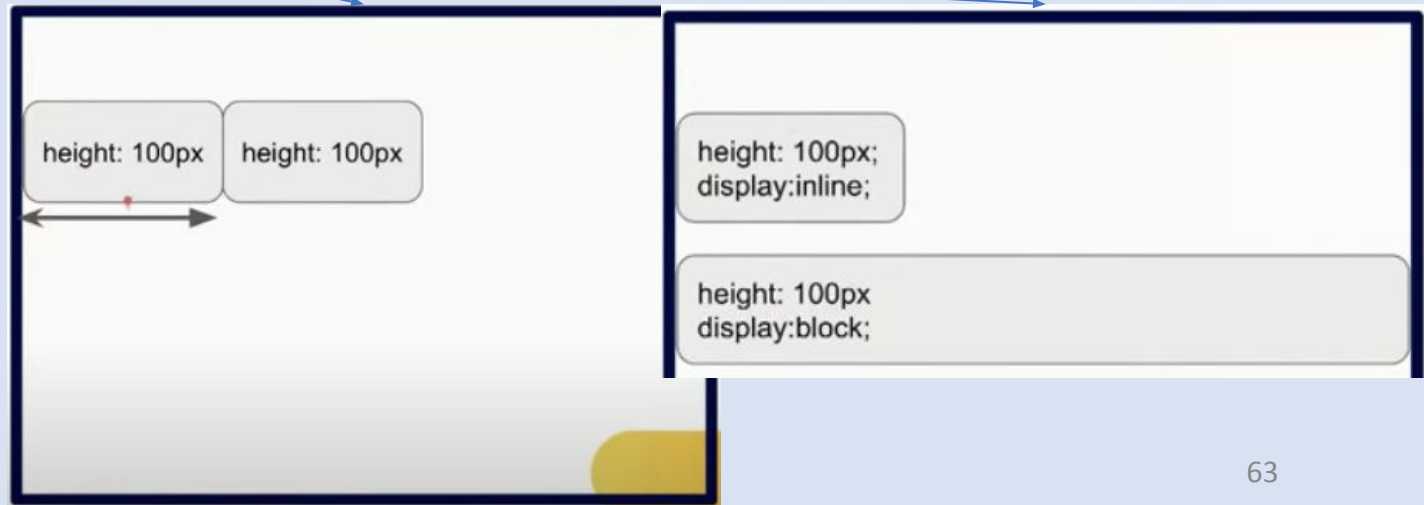
Aquest és un paràgraf en un div.

Un altre paràgraf en el mateix div.

```
div {  
    display: block;  
    background-color: lightblue;  
}  
  
p {  
    display: block;  
}
```

CSS

- Display i inline:
 - No comença en una línia nova i només ocupa tanta amplada com sigui necessària.
 - Els elements adjacents poden aparèixer als costats de cada element.
 - Apareixen un al costat de l'altre quan els elements adjacents també són en línia. Però en el cas que sigui element en línia i després un element en bloc, apareixeran en bloc.



CSS

- Display i inline:
 - Els elements més típics són:
 - ``
 - `<a>`
 - ``
 - `<input>`
 - `<button>`
 - Els elements en línia com "span", "a", "img", etc., no comencen en una línia nova i ocupen només l'amplada necessària per al seu contingut.
 - Poden estar al costat d'altres elements en línia. Per exemple un span al costat d'una a.

CSS

- Display i inline (Exemple):

Això és un text en línia. Això també és en línia.

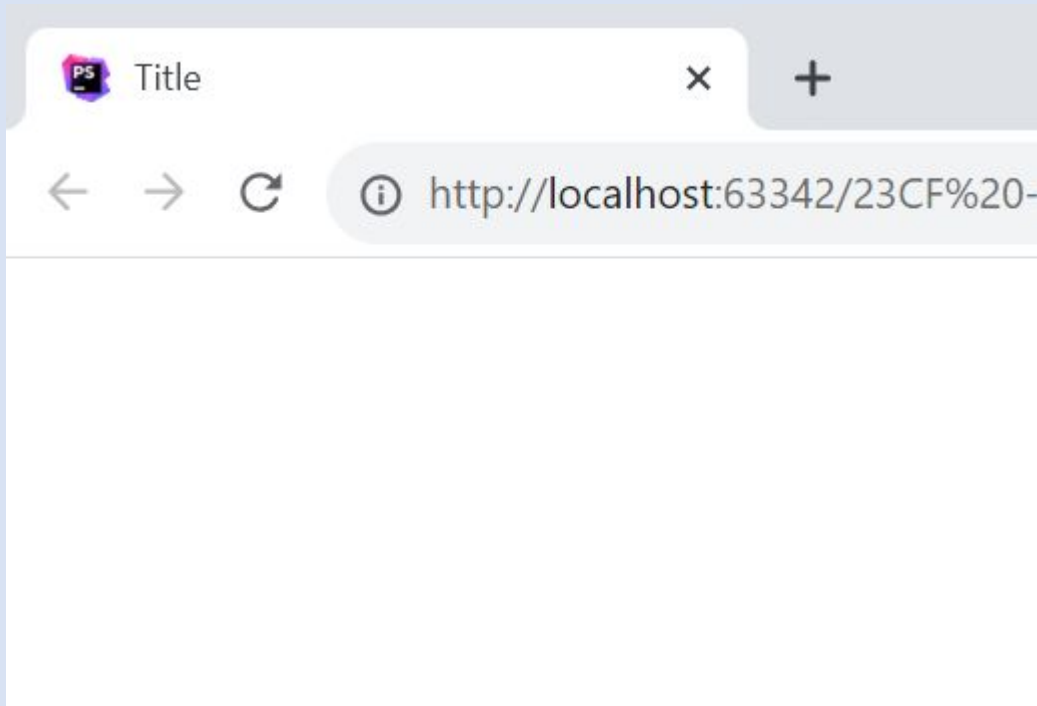
```
span {  
    display: inline;  
    background-color: lightgreen;  
}
```

CSS

- Display i inline:
 - S'utilitza habitualment amb JavaScript per amagar i mostrar elements sense suprimir-los i recrear-los.
- Display legacy:
 - Es conserven per no afectar al comportament anterior. Però no són els elements incorporats més nous.
- Display none:
 - Pot fer que un element sigui completament invisible i no ocupi espai a la pàgina web. És útil per amagar elements sense eliminar-los del codi.

CSS

- Display none (Exemple):



```
p {  
    display: none;  
}
```

CSS

- Display y altres casos. Per veure altres casos, es pot consultar el següent enllaç:

<https://developer.mozilla.org/en-US/docs/Web/CSS/display>

CSS

- Display resum (6 categories amb 28 valors):

<display-outside>

- block
- inline-block
- run-in

<display-inside>

- flow
- flow-root
- table
- flex
- grid
- ruby

<display-internal>

- table-row-group
- table-header-group
- table-footer-group
- table-row
- table-cell
- table-column-group
- table-column
- table-caption
- ruby-base
- ruby-text
- ruby-base-container
- ruby-text-container

<display-box>

- contents
- none

<display-legacy>

- inline-block
- inline-table
- inline-flex
- inline-grid

<display-listitem>

- list-item

CSS

- Sobre escriure el valor per defecte de display s'utilitza l'etiqueta display. Per exemple:

```
span {  
    display: block;  
}
```

- L'etiqueta visibility serveix per ocultar un element.

```
h1.hidden {  
    visibility: hidden;  
}
```

CSS

- Exemple de display:

```
<!DOCTYPE html>
<html>

  <head>
    <title>CSS | Display property</title>

  </head>

  <body>
    <div class="titol">GeeksforGeeks</div>
    <div class="subtitol">display: block; property</div>
    <div class="main">
      <div id="bloc1">Block 1 <span id="main1">Inline</span></div>
      <div id="bloc2">Block 2<span id="main2">Inline-block</span></div>
      <div id="bloc3">Block 3</div>
      <div id="bloc4">Block 4</div>
    </div>
  </body>

</html>
```

```
#bloc1 {
  height: 100px;
  width: 200px;
  background: teal;
  display: block;
}
#bloc2 {
  height: 100px;
  width: 200px;
  background: gold;
  display: block;
}
#bloc3 {
  height: 100px;
  width: 200px;
  background: cyan;
  display: none;
}
```

```
#bloc4 {
  height: 100px;
  width: 200px;
  background: green;
  display: block;
  visibility: hidden;
}

.titol {
  margin-left: 20px;
  font-size: 42px;
  font-weight: bold;
  color: #009900;
}

.subtitol {
  font-size: 25px;
  margin-left: 30px;
}
```

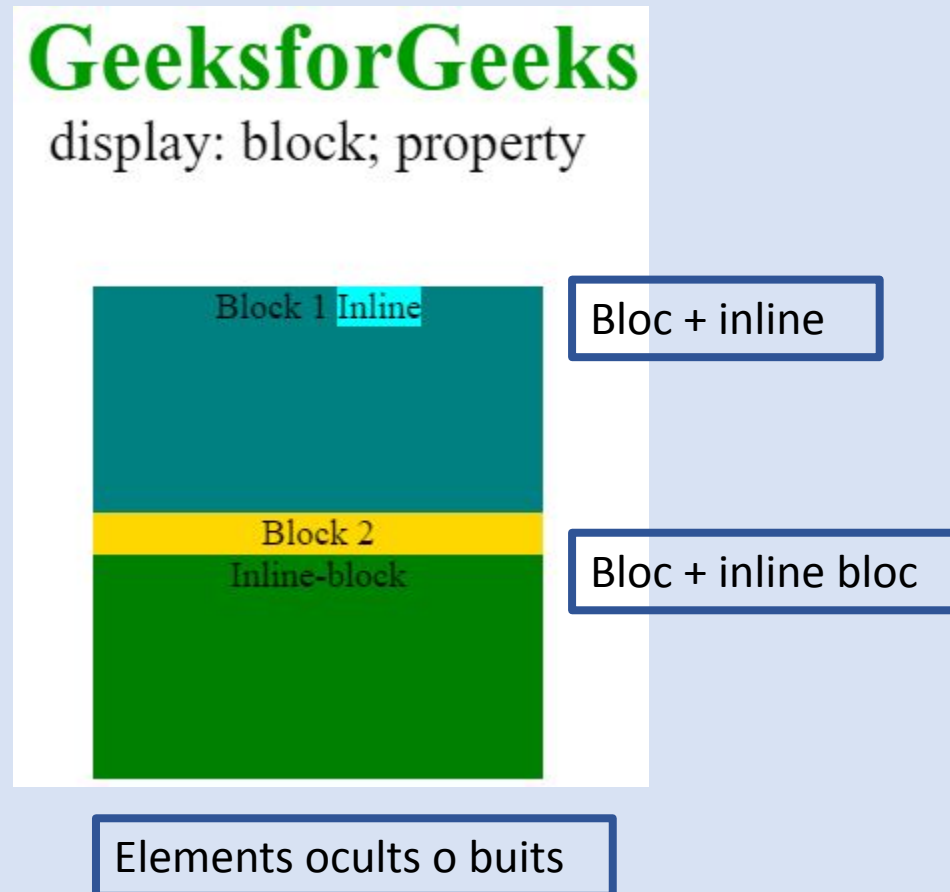
```
.main {
  margin: 50px;
  text-align: center;
}

#main1 {
  height: 200px;
  width: 200px;
  background: cyan;
  display: inline;
}

#main2 {
  height: 100px;
  width: 200px;
  background: green;
  display: inline-block;
}
```

CSS

- Exemple de display (Continuació):



CSS

- Posicionament(Position):
 - Especifica el tipus de mètode de posicionament utilitzat per a un element.
 - Els tipus de posicionament són:
 - static
 - relative
 - fixed
 - absolute
 - sticky

CSS

- Position – Static:
 - Els elements no estan afectats per les propietats: top, bottom, left i right.
 - Sempre estan posicionats segons el flux de la pàgina.
 - És el valor per defecte i que els elements es comporten com a elements "static" de manera predeterminada.
 - L'assignació és position:static.

CSS

- Position – estàtica (Exemple):

Posició Estàtica

Aquest element està en posició estàtica i segueix el flux normal del document.

Posició Estàtica

Aquest element està en posició estàtica, no desplaçat 50px cap amunt i 50px cap a l'esquerra respecte la seva posició normal.

```
.static-box {  
    position: static;  
    background-color: lightblue;  
}  
  
.static-box2 {  
    background-color: lightcoral;  
    position: static;  
    top: 50px;  
    left: 50px;  
}
```

CSS

- Position – relative:

- Es posiciona amb relació amb la seva posició normal. Aquest desplaçament no afectarà a la resta d'elements de la pàgina.
- És molt semblant a static amb la diferència que es pot modificar les propietats top, bottom, left i right.
- La resta de contingut no s'ajustarà perquè s'adapti a cap espai deixat per l'element. Així doncs, quedarà buit aquest espai.

CSS

- Position – Static vs Position – relative(Exemple):

Posició Estàtica

Aquest element està en posició estàtica i segueix el flux normal del document.

Posició Relativa

Aquest element està en posició relativa, desplaçat 50px cap amunt i 50px cap a l'esquerra respecte la seva posició normal.

CSS

- Position – Fixed:
 - Posicionament "fixed" es mantenen fixos a la finestra del navegador i no es mouen quan es fa scroll.
 - Les propietats top, bottom, left i right ajuden a situar l'element.
 - No deixa cap espai a la pàgina on s'hauria d'utilitzar. És a dir, aquests elements no ocupen l'espai on haurien d'estar.
- Position – Absolute:
 - Enlloc de posicionar-se a la finestra com fa fixed, es posiciona segons un element antecessor més proper que no sigui "static".

CSS

- Position – Absolute vs Position - Fixed(Exemple):

Posició Absoluta

Aquest element té una posició absoluta i es col·loca respecte al seu element pare (container).

Posició Fixa

Aquest element té una posició fixa i es manté sempre visible a la part superior de la finestra, fins i tot quan es fa scroll.

```
.absolute-box {  
    position: absolute;  
    top: 20px;  
    left: 20px;  
    background-color: lightcoral;  
}
```

```
.fixed-box {  
    position: fixed;  
    top: 20px;  
    left: 20px;  
    background-color: lightblue;  
}
```

CSS

- Position – Sticky:
 - Es col·loca en funció de la posició de desplaçament de l'usuari.
 - Es commuta entre de relative i fixed, depenen de la posició de desplaçament. És relatiu fins que es troba una posició de desplaçament determinada a la finestra gràfica. Després es queda al seu lloc.
 - Pot ser útil per a la creació d'elements d'encapçalament o menús que segueixin a l'usuari mentre es fa scroll.

CSS

- Position – Static vs Position – sticky(Exemple):

Posició Relativa

Posició Sticky

Aquest element té una posició sticky i es manté "enganxat" a la part superior de la finestra quan fas scroll, fins que arriba a la posició especificada (top: 50px).

Posició Sticky

Aquest element té una posició sticky i es manté "enganxat" a la part superior de la finestra quan fas scroll, fins que arriba a la posició especificada (top: 50px).

Amb sticky, l'element es manté enganxat a la part superior de la finestra fins que arriba a la posició especificada. 50 px en aquest cas.

CSS

- Exemple de posicionament:

```
<!DOCTYPE html>
<html>

  <head>

  </head>

  <body>
    <div>
      
    </div>
    <div>
      
    </div>
    <div>
      
    </div>
    <div>
      
    </div>
    <div>
      
    </div>
  </body>

</html>
```

CSS

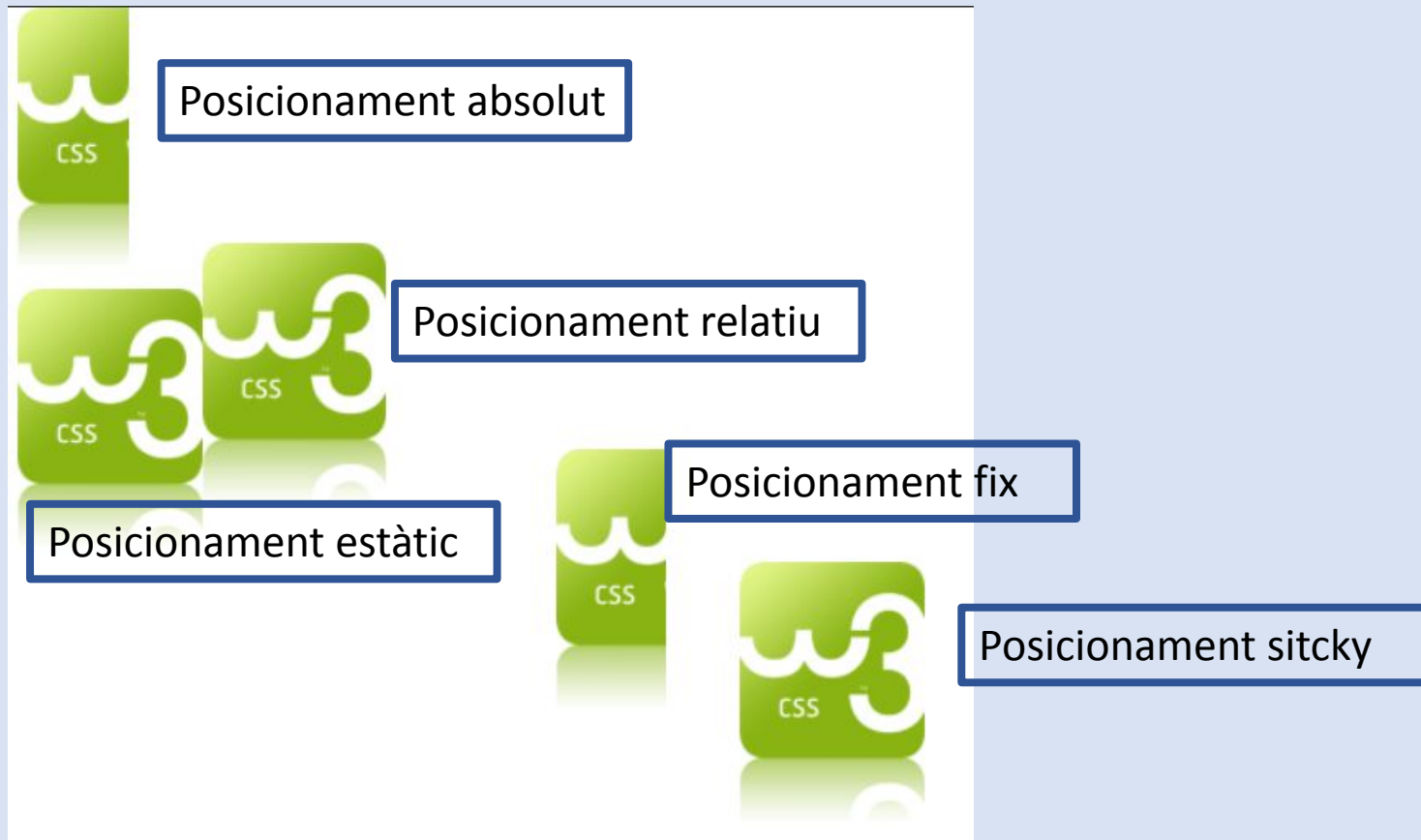
- Exemple de posicionament (continuació):

```
#abs {  
  position: absolute;  
  top: 0px;  
  clip: rect(0px, 60px, 200px, 0px);  
}  
  
#rel {  
  position: relative;  
  top: 120px;  
  left: 100px;  
  clip: rect(0px, 60px, 200px, 0px);  
}  
  
#sta {  
  position: static;  
  top: 300px;  
  left: 200px;  
  clip: rect(0px, 60px, 200px, 0px);  
}
```

```
#fix {  
  position: fixed;  
  top: 240px;  
  left: 300px;  
  clip: rect(0px, 60px, 200px, 0px);  
}  
  
#sti {  
  position: -webkit-sticky;  
  position: sticky;  
  top: 480px;  
  left: 400px;  
  clip: rect(0px, 60px, 200px, 0px);  
}
```

CSS

- Exemple de posicionament (continuació):



CSS

- Element flotant (Float):

- Serveix per a posicionar i donar-li format a un contingut.
- Per exemple que un text rodegi una imatge.
- Les etiquetes són:
 - float □ indicar com l'element hauria de flotar. Permet als elements d'HTML desplaçar-se cap a la dreta o cap a l'esquerra dels seus contenidors, segons com es vulgui que es disposin.
 - clear □ ajuda a controlar com altres elements de la pàgina es comporten respecte a un element flotant. Per exemple, si vols assegurar-te que cap altre element pugui flotar al costat d'un element amb "float", pots utilitzar "clear" per forçar una nova línia a continuació.
 - clearfix □ s'utilitza per corregir quan la imatge sobre surt de l'element que conté. És un patró comú utilitzat per afrontar problemes de disseny quan els elements flotants poden sortir dels seus contenidors i afectar altres parts de la pàgina. Clearfix implica l'ús de CSS per netejar i corregir aquest comportament inesperat.

CSS

- Exemple d'element flotant



En aquest exemple, tenim un div amb la classe caixa que conté un paràgraf i una imatge. L'element caixa està configurat per flotar cap a l'esquerra, la qual cosa farà que altres elements contigus s'alinien a la seva dreta.

CSS

- Exemple de flotant:
 - Hi ha un exemple complet de flotació al següent enllaç:
 - https://jsfiddle.net/Eloi_Vazquez/bh9svdj6/
 - Tal com es veu a l'exemple hi ha flotació a la dreta, esquerra o sense.
 - També hi ha l'ús de les propietats clear i clearfix.

CSS

- Media query:
 - S'utilitza per incloure a un bloc de propietats CSS si es compleix una determinada solució.
 - Utilitza l'etiqueta @media.

CSS

- Exemple Media Query:

```
<!DOCTYPE html>
<html>

  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>

  <body>

    <p>Resize the browser window. When the width of this document is 600 pixels or less, the background-color is "lightblue", otherwise it is "lightgreen".</p>

  </body>
</html>
```

```
body {
  background-color: lightgreen;
}

@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

Resize the browser window. When the width of this document is 600 pixels or less, the background-color is "lightblue", otherwise it is "lightgreen".

Pantalla menor i igual a 600 pixels

Resize the browser window. When the width of this document is 600 pixels or less, the background-color is "lightblue", otherwise it is "lightgreen".

Pantalla major que 600 pixels

CSS

- Un exemple que resumeix els trets més importants el podeu trobar al següent enllaç:
- <https://jsfiddle.net/malbareda/gnfsojp0>

CSS

- Si voleu ampliar coneixements, podeu revisar aquests enllaços:
- <https://www.w3schools.com/css/default.asp>
- <https://www.geeksforgeeks.org/css-introduction/?ref=lbp>