

---

# CCS

## *Cascading Style Sheets*

---

# 1.- Introducció

---

Cascading Style Sheets (CSS) és un llenguatge de marques utilitzat donar **format** i **estils** al contingut d'una pàgina web. És una part essencial del desenvolupament web modern, ja que permet als desenvolupadors controlar l'aparença dels seus llocs i garantir una experiència d'usuari agradable.

Opera conjuntament amb llenguatges de marques com **HTML** i **XML**

**CSS proporciona mecanismes per a definir com s'han de presentar els elements marcats (XML, HTML).**

Amb CSS podem especificar com volem que es mostren cadascun dels elements d'una pàgina web, incloent fonts, colors, mides, marges, i moltes altres propietats.

CSS ens permet definir les **regles de presentació** d'un document **a partir de les etiquetes** que conté, fa que no calgui barrejar les dades amb el contingut. Però hi ha que anar amb compte, ja que no tots els navegadors tracten correctament ni de la mateixa forma CSS. S'han de provar els documents en diferents navegadors per comprovar que tot es visualitza bé

La idea és **separar contingut i estil**, esta separació ens permet una major **flexibilitat** en la creació de llocs web i ens facilita les tasques de **manteniment** i **actualitzacions** dels mateixos

## Versions de CSS

- **CSS 1:** La versió inicial oferia unes poques capacitats de visualització però ja no està suportada pel W3C
- **CSS 2:** Expandeix les característiques de CSS1 afegint noves funcionalitats
- **CSS 3:** Està definit en mòduls i dona un control més potent de la visualització dels documents, és la més utilitzada actualment.

## Tipus de llenguatges de marques amb CSS

- **HTML (Hypertext Markup Language):** llenguatge de marques estàndard per a la creació de pàgines web. Utilitzant CSS, podem estilitzar tots els elements HTML, com ara encapçalaments, paràgrafs, llistes, taules i molts altres.
- **XML (Extensible Markup Language):** CSS també s'utilitza amb XML, que és un llenguatge de marques extensible utilitzat per a la transferència de dades. Podem utilitzar CSS per donar format a les dades contingudes en documents XML.
- **SVG (Scalable Vector Graphics):** Podem aplicar CSS a elements SVG, que són **imatges vectorials basades en XML**. Això permet controlar l'estil dels gràfics i les il·lustracions d'una manera similar al contingut HTML.

Els documents XML i HTML són semblants i per tant es podran tractar de la mateixa forma, encara que hi ha diferències perquè tenen **objectius diferents**.

- **Visualització d'XML i HTML**

La manera més habitual de visualitzar documents XML o HTML són els navegadors web.

En HTML, els navegadors representen les etiquetes amb un estil predeterminat, es a dir, a cada etiqueta HTML li assignen una forma per defecte de representar-a, per exemple: **<h1>: Arial, 26px, negreta, ...**

En canvi com XML té un numero infinit d'etiquetes no es pot donar un estil predeterminat i per tant, mostren tal qual estan escrits.

**XML es veu com un format per emmagatzemar dades**, per tant necessitem alguna forma per mostrar la informació als humans. XML no està pensat per a ser llegit, es pot interpretar però no sol ser còmode ni fàcil.

Per això hi ha vocabularis XML pensats per a ser visualitzats com:

- SVG: Mostrar imatges vectorials2D
- XHTML: Representació de dades en format web
- MathXML: Fórmules matemàtiques
- etc...

Per mostrar les dades XML per a poder llegir-les tenim dos opcions:

- **XSLT**: En proporciona mecanismes pe a transformar el document a un format per a ser llegit per les persones (HTML, PDF, etc..)
- **CSS**: Ens permet especificar la manera en que s'han de mostrar les dades

## 2.- Associar CSS amb documents XML i HTML

### • CSS i XML

Per aplicar fulles d'estil **CSS** a **XML** es fa servir l'etiqueta especial **xml-stylesheet**, s'ha de definir sempre darrere de l'etiqueta de definició d'XML.

```
<?xml-stylesheet type="text/css" href="estil.css"?>
```

També es poden afegir més atributs com:

- **Title:** El nom o el títol del full
- **Media:** Indica sota quin medi es farà servir el CSS: screen, print, tty, aural, braille, handheld, tv, all (Se'n poden posar diversos alhora)
- **Charset:** Indica el codi de caràcters que es fa servir
- **Alternate:** Indica si el CSS que es fa servir és el preferit. Pot ser 'yes' o 'no'. (Per defecte és 'no')

XML	CSS (estil.css)
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;?xml-stylesheet type="text/css" href="estil.css"?&gt; &lt;recepta&gt;   &lt;titol&gt;Paella de Verdura&lt;/titol&gt;   &lt;ingredients&gt;     &lt;ingredient&gt;300g de fesols&lt;/ingredient&gt;     &lt;ingredient&gt;2 tomàquets madurs&lt;/ingredient&gt;     &lt;ingredient&gt;200g de carabasseta&lt;/ingredient&gt;     &lt;ingredient&gt;1 pebrot verd&lt;/ingredient&gt;     &lt;ingredient&gt;200g de garrofó&lt;/ingredient&gt;     &lt;ingredient&gt;300g de arròs&lt;/ingredient&gt;     &lt;ingredient&gt;4 tasses de caldo de verdura&lt;/ingredient&gt;     &lt;ingredient&gt;Azafrà, safrà o colorant alimentari&lt;/ingredient&gt;     &lt;ingredient&gt;Oli d'oliva&lt;/ingredient&gt;     &lt;ingredient&gt;Sal al gust&lt;/ingredient&gt;   &lt;/ingredients&gt;   &lt;instruccions&gt;     &lt;pas&gt;Sofregeix fesols, pebrot i la carabasseta amb oli d'oliva.&lt;/pas&gt;     &lt;pas&gt;Afegeix els tomàquets ratllats&lt;/pas&gt;     &lt;pas&gt;Cuina fins que la verdura estiga tendra.&lt;/pas&gt;     &lt;pas&gt;Afegeix el garrofó i l'arròs i remena bé.&lt;/pas&gt;     &lt;pas&gt;Afegeix el caldo calent, l'azafrà o colorant i la sal. &lt;/pas&gt;     &lt;pas&gt;Deixa coure durant 15-20 minuts.&lt;/pas&gt;     &lt;pas&gt;Deixa reposar uns minuts abans de servir.&lt;/pas&gt;   &lt;/instruccions&gt; &lt;/recepta&gt;</pre>	<pre>recepta {   display: flex;   flex-direction: column;   align-items: center; }  titol {   font-size: 24px;   font-weight: bold;   color: #0066cc;   text-align: center;   margin-bottom: 15px; }  ingredients, instruccions {   width: 70%;   margin-bottom: 20px; }  ingredient, instruccio {   margin-bottom: 10px; }</pre>

#### Paella de Verdura

300g de fesols 2 tomàquets madurs 200g de carabasseta 1 pebrot verd 200g de garrofó 300g de arròs 4 tasses de caldo de verdura  
Azafrà, safrà o colorant alimentari Oli d'oliva Sal al gust

En una paella, sofregeix els fesols, el pebrot i la carabasseta amb oli d'oliva. Afegeix els tomàquets ratllats i cuina fins que la verdura estiga tendra. Afegeix el garrofó i l'arròs i remena bé. Afegeix el caldo calent, l'azafrà o colorant i la sal. Deixa coure durant 15-20 minuts. Deixa reposar uns minuts abans de servir.

**Nota:** Els navegadors no suporten CSS embegut en un XML

## • CSS i HTML

Podem incloure CSS en HTML de dos maneres, amb **un fitxer extern** i **incrustat** en el propi document CSS.

### CSS incrustat:

No és la forma més recomanada de fer-ho ja que complica la compartició, la flexibilitat i el manteniment i modificació de llocs web.

L'etiqueta **<STYLE>** : permet especificar regles CSS dins del document HTML

```
<style type="text/css">
... regles ...
</style>
```

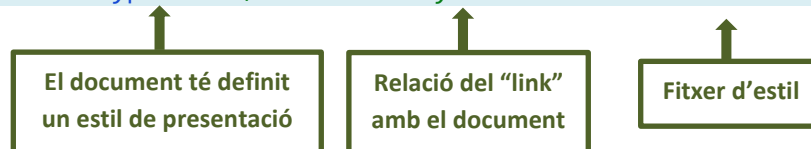
Atribut **style**: permet definir regles específiques per a un element en particular.

```
<p style="color:green">
```

### Fitxer Extern:

L'etiqueta **<LINK>**: serveix per definir un document CSS extern, accepta més o menys els mateixos atributs que l'etiqueta xml-stylesheet

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



## 3.- Regles CSS

### 3.1.- Regles bàsiques

Un conjunt de regles CSS consta d'un **selector** i un bloc de **declaració**:

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



- El **selector** apunta a l'etiqueta a la que volem aplicar l'estil.
- El **bloc de declaració** conté **una o més declaracions** separades per punt i coma (;).
  - Cada declaració inclou un nom de **propietat** CSS i un valor separat per dos punts (:).
  - Varies declaracions de CSS es separen amb (;) i els blocs de declaració els envoltam amb claus {}.

**Exemple:**

```
h1 {
  font-family:arial;
  font-weight:bold;
  color:#FF0000;
  background-color:#00FF00;
}
```

- **Herència**

La "C" de **CSS** vol dir "Cascading" i vol dir que les propietats es poden heretar pels elements fill:

```
h1, h2 {
  font-family:arial;
  font-weight:bold;
  color:#FF0000;
  background-color:#00FF00;
}
h2 { font-style: italic; }
```

En este exemple "h2" a més de les característiques anteriors estarà en cursiva

Si repetim una etiqueta en el mateix nivell i es repeteix alguna propietat, l'últim valor és el vàlid:

```
nom {
  font-family:arial;
  color: blue;
}
nom { color:red; }
```

Com la darrera definició de "nom" és **color:red**; el contingut de nom es pintarà de color roig.

## 3.2.- Regles CSS- Selectors

---

A banda d'especificar etiquetes disposem d'uns selectors predefinitos.

L'estructura bàsica d'una regla CSS és la següent:

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

On:

- **Selector:** Indica quins elements seran afectats per la regla.
- **Propietat:** Especifica l'aspecte que volem canviar.
- **Valor:** Indica el valor de la propietat que s'aplicarà als elements seleccionats.

La selecció d'elements la podem fer de varies maneres, les més bàsiques són:

- **Per element**

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

- **Per classe**

```
.classe {  
    font-size: 18px;  
}
```

Html:

```
<p class="classe">Text amb estil</p>
```

- **Per id**

```
#identificador {  
    background-color: lightgray;  
}
```

Html:

```
<div id="identificador">Contingut amb estil</div>
```

- **Selector de descendents**

```
article p {  
    font-style: italic;  
}
```

- **Selector de grups**

```
h1, h2, h3 {  
    text-align: center;  
}
```

- **Selector universal**

```
* {  
    margin: 0;  
    padding: 0;  
}
```

- **Pseudo selectors:**

També hi ha una sèrie de selectors especials anomenats pseudoselectors. Alguns d'ells són:

Pseudo-selector	Funció
:first-line	Selecciona la primera línia
:first-letter	Selecciona la primera lletra
:before	Selecció abans de l'etiqueta
:after	Selecció després de l'etiqueta



### 3.3.- Regles CSS- Disseny

#### • Model de Caixa

En CSS es treballa amb caixes, ajustant el seu tamany, el color, la posició, etc. Es podria pensar que els elements HTML/XML com caixes apilades una damunt l'altra

Quan es visualitza qualsevol cosa amb CSS és tractada com si estés dins d'una caixa rectangular, cada caixa té **quatre** components:

- **contingut** : El contingut que es va a mostrar
- **padding** (emplenament), és l'espai al voltant del contingut
- **border** (marc), és la línia que es troba fora del
- **margin** (marge), és l'espai fora del element.



Podem modificar el color de fons o del contingut (generament lletra) amb **background-color** i **color** respectivament

#### • Amplada/Altura

Els atributs "**width**" i "**height**" (ample i alt) permeten forçar la mida de la caixa

```
nom { width: 300px; height:100px; }
```



#### • Disseny

Per defecte els valors de margin, padding i border estan a zero podem canviar-ne els valors amb les propietats:

- **Margin:** **margin**, **margin-left**, **margin-right**, **margin-top**, **margin-bottom**
- **Padding:** **padding**, **padding-left**, **padding-right**, **padding-top**, **padding-bottom**

Exemple:

```
nom { margin: 3px; padding-left:4px; }
```

## • Border

La propietat **border** en CSS s'utilitza per estilitzar les **vores** dels elements HTML. Té diverses subpropietats que permeten controlar l'estil, l'amplada i el color de la vora.

- **border-style**: Defineix l'estil de la vora, pot ser **none**, **solid**, **dashed**, **dotted**, **double**, **groove**, **ridge**, **inset**, **outset**

```
border-style: solid; /* Vora sòlida */
```

- **border-width**: Especifica l'amplada de la vora. S'expressa normalment en **píxels** (px).

```
border-width: 2px; /* Amplada de 2 píxels */
```

- **border-color**: indica el color de la vora. Pot ser especificat amb el nom del color en **anglès**, la funció **rgb()** o el codi hexadecimal.

```
border-color: red; /* Color roig */
```

La funció **rgb()** permet especificar un color mitjançant els seus components **Red** (Roig), **Green** (Verd) i **Blue** (Blau). Aquesta funció pren **tres valors numèrics** que representen la intensitat de cada color. Cada valor ha de ser un nombre entre 0 i 255.

```
.example {  
  border-style: solid;  
  border-width: 2px;  
  border-color: rgb(255, 0, 0); /* Vermell pur */  
}
```

- **Border**: És una propietat abreviada que permet especificar tota la informació de la vora al mateix temps, en este ordre: **amplada**, **estil** i **color**.

```
border: 1px solid #3498db; /* Amplada de 1 píxel, vora sòlida i color blau */
```

Exemples:

```
/* Vora amb punts verds */  
.example1 {  
  border-style: dotted;  
  border-width: 2px;  
  border-color: green;  
}
```

```
/* Vora doble vermelles amb amplada 3 píxels */  
.example2 {  
  border: 3px double red;  
}
```

### 3.4.- Regles CSS- Propietats

CSS defineix moltes propietats i a cada versió n'afegeixen més podem veure un resum en les següents webs:

- **Propietats estables:**

<http://www.mclibre.org/consultar/htmlcss/css/css-propiedades.html>

- **Propietats beta:**

<http://www.mclibre.org/consultar/htmlcss/css/css3-propiedades.html>

- **Unitats**

Existeixen molts tipus d'unitats en CSS, les podem utilitzar cadascuna d'elles en funció de les nostres necessitats:

Longitud	Color
Pixels ( <b>px</b> ), ems ( <b>em</b> ), percentatge (%), punts ( <b>pt</b> ), polzades ( <b>in</b> ), centímetres ( <b>cm</b> )	<b>rgb(r,g,b),</b> <b>#rrggbb,</b> <b>#rgb</b>
<b>width:</b> 744px; <b>margin-left:</b> 1.25em; <b>left:</b> 34%; <b>margin-top:</b> 1.25in; <b>margin-bottom:</b> 1.5cm;	<b>background-color:</b> rgb(255,0,0); <b>color:</b> #FF2200; <b>color:</b> #f20;

La unitat "**em**" en CSS és relativa i es basa en la grandària de la lletra de l'element pare. Un valor de "**1em**" equival a la grandària de la lletra de l'element pare. Per exemple, si l'element pare té una grandària de lletra de 16 píxels, "1em" serà igual a 16 píxels. Si apliquem "2em", la grandària serà el doble de la grandària de la lletra de l'element pare.

Per exemple:

```
body {
  font-size: 16px; /* Grandària de la lletra per defecte del document */
}

p {
  font-size: 1.2em; /* La grandària de la lletra dels paràgrafs serà 1.2 vegades la del body */
}

h1 {
  font-size: 2em; /* La grandària de la lletra dels títols h1 serà el doble de la del body */
}
```

Podem agrupar les propietats en 4 grans grups:

- Propietats de tipus de lletres
- Propietats de color i fons de pantalla
- Propietats de text
- Propietats de caixes

- **Tipus de Lletres:**

## Basades en la propietat **font**

<b>font-family</b>	Defineix la família de tipus de lletres ( <b>font</b> ) a utilitzar. Pot incloure múltiples noms de famílies separats per comes, en cas que la primera no estiga disponible. <p><code>p { font-family: "New Century Schoolbook", Times, serif }</code></p>
<b>font-size</b>	Estableix la grandària del tipus de lletra. Pot ser especificada en diferents unitats vistes anteriorment ( <b>px</b> ), ( <b>em</b> ), ( <b>%</b> ), ( <b>pt</b> ), ( <b>in</b> ), ( <b>cm</b> ). <p><code>h2 { font-size: 12pt; }</code></p>
<b>font-weight</b>	Indica el <b>gruix</b> dels caràcters del tipus de lletra. Pot ser valors numèrics com <b>normal</b> o <b>bold</b> , o noms com <b>lighter</b> o <b>bolder</b> . <p><code>span { font-weight: bold; }</code></p>
<b>font-variant</b>	Especifica si el tipus de lletra ha de ser mostrat en versió <b>normal</b> o amb majúscules petites ( <b>small-caps</b> ). <p><code>H3 { font-variant: small-caps; }</code></p>
<b>font-style</b>	Defineix l'estil del tipus de lletra. Pot ser <b>normal</b> , <b>italic</b> o <b>oblique</b> . <p><code>p { font-style: italic; }</code></p>
<b>font</b>	Permet especificar múltiples propietats relacionades amb el tipus de lletra en una sola línia: <p><code>h1 { font: bold small-caps 24px/1.5 "Arial", sans-serif; }</code>  <b>Nota:</b> <code>24px/1.5</code> estableix la grandària a 25 px i l'altura de la línia dins del bloc de text a 1.5 (<b>line-height</b>)</p>

- A CSS, les "famílies de fonts" es refereixen a grups de tipus de lletres amb característiques similars.
- Quan especifiquem la família de fonts, és una bona pràctica **proporcionar alternatives genèriques** en cas que la primera opció no estiga disponible, utilitzant una llista de famílies de fonts separades per comes, amb el **tipus genèric al final**.
- Si el nom d'una font conté més d'una paraula el posem entre cometes: **"Courier New"**

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	<i>Brush Script MT</i> <i>Lucida Handwriting</i>
Fantasy	<b>Copperplate</b> Papyrus

Per exemple:

```
p {
  font-family: "Times New Roman", Times, serif;
}
```

## • Colors i fons

### Propietats CSS de colors

Propietat	Descripció	Valors
<b>color</b>	Defineix el color del text. <b>color: red;</b>	<b>nom</b> de color (p. ex.: red), valor <b>hexadecimal</b> (#FF0000), valor <b>RGB</b> (rgb(255, 0, 0)), <b>initial, inherit</b>
<b>background-color</b>	Estableix el color de fons. <b>background-color: #f0f0f0;</b>	<b>nom</b> de color, valor <b>hexadecimal</b> , valor <b>RGB</b> , <b>transparent</b> , <b>initial</b> , <b>inherit</b>
<b>opacity</b>	Especifica la transparència. <b>opacity: 0.5;</b>	valor entre: <b>0 (transparent)</b> i <b>1 (opac)</b> , <b>initial, inherit</b>
<b>rgba</b>	Especifica el color en el model RGBA. <b>background-color : rgba(255, 0, 0, 0.5);</b> (el 0.5 és la opacitat)	

- **inherit**: Este valor fa que l'element prengui el valor de la propietat del seu element pare.

```
div {
  color: blue;
}

h2 {
  color: inherit; /* L'h2 serà blau com el div pare */
}
```

- **initial**: Valor que restableix la propietat a la seva configuració inicial.

```
h2 {
  border-color: green;
}

h2 {
  border-color: initial; /* El color de la vora de l'h2 es restableix a 'none' */
}
```

Podem definir els colors per el seu nom (predefinit) o usant valors RGB, HEX, HSL, RGBA, HSLA

Exemples noms predefinits:



## Propietats CSS de fons

Propietat	Descripció	Valors
<code>background-image</code>	Especifica una imatge com a fons. <code>background-image: url('imatge.jpg');</code>	URL, none
<code>background-repeat</code>	Controla la repetició de la imatge de fons. <code>background-repeat: repeat-x</code>	repeat, repeat-x, repeat-y, no-repeat, space, round
<code>background-position</code>	Defineix la posició inicial de la imatge de fons. <code>background-position: center top;</code>	top, bottom, left, right, center, x% y%, x-pos y-pos
<code>background-size</code>	Controla la mida de la imatge de fons. <code>background-size: cover;</code>	auto, cover, contain, length, percentage
<code>background-attachment</code>	Especifica si la imatge de fons es desplaça amb la pàgina o permaneceix fixa. <code>background-attachment: fixed</code>	scroll, fixed, local
<code>background</code>	Permet definir totes les propietats de fons en una sola línia. <code>background: #f0f0f0 url('imatge.jpg') repeat-x center top;</code>	color, image, repeat, attachment, position/size, origin, clip

## Propietats de Caixa

Propietat	Descripció	Exemple
<b>width</b>	Estableix l'amplada d'un element.	<b>width: 300px;</b>
<b>height</b>	Fixa l'altura d'un element.	<b>height: 200px;</b>
<b>margin</b>	Defineix els marges externs de l'element.	<b>margin: 10px;</b>
<b>padding</b>	Estableix l'espai interior entre contingut i vora.	<b>padding: 15px;</b>
<b>border</b>	Defineix la vora d'un element. ( <b>border-width</b> , <b>border-color</b> , <b>border-style</b> )	<b>border: 2px solid #000</b>
<b>box-sizing</b>	Controla com es calcula la mida total de l'element.	<b>box-sizing: border-box;</b>
<b>display</b>	Defineix el comportament de visualització de l'element.	<b>display: block;</b>
<b>position</b>	Controla com es posiciona l'element dins del seu contenidor.	<b>position: relative;</b>
<b>float</b>	Fa que un element flote cap a la dreta o cap a l'esquerra.	<b>float: left</b>
<b>clear</b>	Controla com els elements es comporten quan es troben amb elements flotants.	<b>clear: both;</b>
<b>visibility</b>	Normalment es fa servir perquè una caixa no es vegi amb <b>'hidden'</b> . Conserva l'espai	<b>visibility: visible;</b> <i>/*L'element serà visible*/</i>  <b>visibility: hidden;</b> <i>/* L'element estarà ocult, però continuarà ocupant el seu espai */</i>

## 4.- Posicionament CSS

---

El posicionament en CSS és crucial per **controlar com els elements es disposen a la pàgina web**. Les propietats clau per aquest propòsit són **position**, **top**, **right**, **bottom** i **left**.

- **Posició (position):**

- **static** (per defecte): Els elements segueixen l'ordre normal del flux del document.
- **relative**: L'element es desplaça de la seva posició normal, però el seu espai reservat original es manté.
- **absolute**: L'element es posiciona respecte al seu primer element posicionat (no static) parell.
- **fixed**: L'element es manté fix en la pantalla, independentment de l'scroll.
- **sticky**: Combinació de relative i fixed. Es comporta com relative fins que l'element arriba a una posició específica en l'scroll, moment en què es comporta com fixed.

- **Propietats de Posicionament**

### Top, Right, Bottom, Left:

- **top**: Indica la distància des de la part superior.
- **right**: Indica la distància des de la part dreta.
- **bottom**: Indica la distància des de la part inferior.
- **left**: Indica la distància des de la part esquerra.

Exemple:

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

### Z-index (z-index)

- Controla l'apilament d'elements que es solapen.
- Els elements amb un z-index més alt estan per sobre els de menor valor.

Exemple:

```
}  
.element1 {  
  z-index: 2;  
}  
  
.element2 {  
  z-index: 1;  
}
```



## Float (**float**)

- Fa que un element flote cap a la dreta o cap a l'esquerra.
- Altres elements fluiran al voltant d'aquest.

Exemple:

```
.flotant {  
  float: left;  
}
```

## Clear (**clear**)

- Controla com els elements han de comportar-se al voltant dels elements flotants.

Exemple:

```
.neteja {  
  clear: both;  
}
```