

UD3.1 INTRODUCCIÓ AL CSS

ÍNDEX

1.	<i>INTRODUCCIÓ A CSS (CASCADING STYLE SHEETS)</i>	2
1.1.	Què és CSS?	2
1.2.	Per què és important CSS en el disseny web?	2
1.3.	Història i evolució de CSS	3
2.	<i>INCORPORANT CSS EN HTML</i>	4
2.1.	Selectors de Tipus, Classe i ID	4
2.2.	Ús de CSS en línia, Incrustat i Extern	6
2.3.	Pseudo-Classes	7
3.	<i>PROPIETATS I VALORS</i>	9
4.	<i>EL MODEL CAIXA</i>	11
4.1.	Caixes en Bloc i en Línia	11
4.2.	Parts d'una caixa	12
4.3.	El model de caixes CSS estàndard	13
4.4.	El model de caixes CSS alternatiu	14

1. INTRODUCCIÓ A CSS (CASCADING STYLE SHEETS)

1.1. Què és CSS?

CSS, que significa "Cascading Style Sheets" en anglès, és un llenguatge de fulles d'estil utilitzat en la creació i disseny de pàgines web. La seua funció principal és controlar la presentació i l'aspecte visual d'un lloc web, permetent als dissenyadors i desenvolupadors web definir com es veuran els elements HTML en el navegador. A través de CSS, es poden establir propietats com el color, la tipografia, l'espaiat i la posició dels elements en una pàgina web.

Entre les seues característiques bàsiques estan:

- **Separació de contingut i presentació:** Una de les característiques clau de CSS és la separació de contingut i presentació. Això significa que podem mantindre el contingut d'una pàgina web (HTML) separat del seu estil (CSS), la qual cosa facilita la gestió i el manteniment del lloc.
- **Reutilització:** El codi CSS es pot emprar per a moltes pàgines distintes i aprofitar-ho.
- **Cascading (Cascada):** El terme "cascading" en CSS es refereix a la forma en què les regles s'apliquen i es combinen. Les regles poden heretar-se i anul·lar-se en funció de la seua especificitat i ubicació en el codi, la qual cosa brinda un alt grau de flexibilitat en el disseny.

1.2. Per què és important CSS en el disseny web?

CSS exerceix un paper fonamental en el disseny web per diverses raons:

- **Millora la usabilitat:** L'ús adequat de CSS permet millorar la usabilitat d'un lloc web en proporcionar una presentació neta i coherent. Els usuaris poden navegar i comprendre el contingut amb major facilitat.
- **Flexibilitat de disseny:** CSS ofereix una gran flexibilitat en la creació de dissenys web. Pots posicionar elements, canviar grandàries, colors i fonts de manera precisa, la qual cosa et permet aconseguir dissenys únics i atractius.
- **Adaptabilitat:** Amb CSS, pots crear llocs web que s'adapten a diferents dispositius i grandàries de pantalla, la qual cosa és essencial en l'era dels dispositius mòbils.
- **Eficiència en el manteniment:** En mantindre la presentació per separat del contingut, qualsevol canvi en l'estil es reflecteix de manera consistent en totes les pàgines del lloc, la qual cosa simplifica les actualitzacions i el manteniment.

1.3. Història i evolució de CSS

El desenvolupament de CSS es remunta a la dècada de 1990, quan es va reconéixer la necessitat de separar la presentació del contingut en la web. Al llarg dels anys, CSS ha experimentat diverses versions i millores, la qual cosa ha portat a un estàndard més madur i poderós. Algunes fites en l'evolució de CSS inclouen:

- **CSS1 (1996)**: La primera versió de CSS, que va introduir les propietats bàsiques per a l'estil de text i el disseny de pàgines.
- **CSS2 (1998)**: Va ampliar les capacitats de CSS, permetent un major control sobre el disseny i la presentació de pàgines web.
- **CSS3 (en curs)**: CSS3 és una especificació en constant evolució que ha introduït nombroses característiques noves, com a animacions, ombres, transformacions i suport per a fonts web personalitzades.
- **CSS4 (en desenvolupament)**: A mesura que la web continua evolucionant, s'estan desenvolupant noves característiques i mòduls per a CSS, la qual cosa amplia encara més el seu potencial.

2. INCORPORANT CSS EN HTML

Una part essencial de treballar amb CSS és aprendre com integrar-ho en un document HTML. Existeixen diverses maneres de fer-ho, i cadascuna té els seus propis avantatges i desavantatges. A continuació, explorarem les diferents maneres d'inserir CSS en un document HTML:

2.1. Selectors de Tipus, Classe i ID

Abans de submergir-nos en les tècniques d'integració, és important comprendre els tipus de selectors que pots utilitzar en CSS. Això influirà en la forma en què apliques els teus estils.

- **Selector de Tipus:** Selecciona tots els elements d'un tipus específic. Per exemple, per a aplicar estils a tots els paràgrafs en el teu document HTML, usaries el selector de tipus "p".

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

- **Selector de Classe:** Selecciona elements que tenen una classe específica. Pots aplicar la mateixa classe a diversos elements en el teu HTML i després estilitzar-los de manera uniforme. El selector de classe s'indica amb un punt, com ".classe".

```
<p class="destacado">Este párrafo está en la clase destacado.</p>
```

```
.destacado {  
    color: red;  
    font-weight: bold;  
}
```

- **Selector d'ID:** Selecciona un element amb un ANEU específic. Els IDs han de ser únics en una pàgina, per la qual cosa este selector és útil per a aplicar estils a un sol element. S'indica amb un símbol de coixinet, com "#aneu".

```
<div id="encabezado">Este es el encabezado</div>
```

```
#encabezado {  
    background-color: gray;  
    color: white;  
}
```

- **Selector universal:** El selector universal selecciona tots els elements en la pàgina. Pots aplicar estils generals a tots els elements d'esta manera. Ací tens un exemple:

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

- **Selector de descendència:** El selector de descendència et permet seleccionar elements niats dins d'altres elements. Per exemple:

```
div p {  
    color: green;  
}
```

2.2.ús de CSS en línia, Incrustat i Extern

CSS es pot incorporar en un document HTML de diverses formes. Cada enfocament té les seues pròpies aplicacions i avantatges.

- **CSS en línia:** Pots agregar estils directament a un element HTML utilitzant l'atribut "style". El CSS en línia és útil per a estils específics en elements individuals, però pot ser difícil de mantindre en pàgines grans. Per exemple:

```
<p style="color: blue;">Est és un paràgraf blau</p>
```

- **CSS incrustat (en línia interna):** Pots inserir CSS en la secció <head> del teu document HTML utilitzant l'etiqueta <style>. Això permet que el CSS s'aplique a tota la pàgina. El CSS incrustat és útil per a aplicar estils a tota la pàgina, però encara no proporciona la separació ideal de contingut i presentació. Per exemple:

```
<head>
<style>
p {
  color: blue;
}
</style>
</head>
```

- **CSS extern:** La forma més comuna i eficient d'aplicar CSS és mitjançant un arxiu extern. Crea un arxiu CSS separat (per exemple, "estils.css") i vincula-ho al teu document HTML amb l'etiqueta <link> en la secció <head>. El CSS extern promou la separació de contingut i presentació, facilitant la gestió d'estils en llocs web grans i complexos. A més, els canvis en l'arxiu CSS es reflectiran automàticament en totes les pàgines que l'utilitzen. Per exemple:

```
<head>
<link rel="stylesheet" type="text/css" href="estils.css">
</head>
```

2.3. Pseudo-Classes

Les pseudo-classes en CSS són selector espcionals que et permeten aplicar estils a elements en funció del seu estat o posició en relació amb altres elements. Ací hi ha algunes pseudo-classes comunes juntament amb exemples per a il·lustrar el seu ús:

- **:hover.** Esta pseudo-classe s'aplica quan el cursor del mouse se situa sobre un element. És molt útil per a crear efectes de ressaltat o interactivitat.

```
a:hover {  
    color: xarxa;  
}
```

En este exemple, quan el cursor se situa sobre un enllaç (`<a>`), el text de l'enllaç es tornarà roig.

- **:active.** S'aplica quan un element està sent activat, com quan es fa clic en un enllaç. És comú usar-ho per a canviar l'estil d'un element durant una interacció.

```
button:active {  
    background-color: #00f;  
    color: #fff;  
}
```

Ací, quan es fa clic en un botó, el seu fons es torna blau (`#00f`) i el text es torna blanc (`#fff`).

- **:focus.** Esta pseudo-classe s'aplica quan un element rep l'enfocament, generalment mitjançant tabulació en un formulari o fent clic en un element interactiu. És útil per a proporcionar retroalimentació visual als usuaris.

```
input:focus {  
    border: 2px solid green;  
}
```

Quan un camp d'entrada (`<input>`) obté el focus, la seua vora es torna verda.

- **:first-child.** Selecciona el primer element fill d'un element pare.

```
ul li:first-child {  
    font-weight: bold;  
}
```

En este cas, s'aplicarà un estil de text en negreta al primer element de la llista dins d'un element ``.

- `:last-child`. Selecciona l'últim element fill d'un element pare.

```
ul li:last-child {  
    text-decoration: underline;  
}
```

L'últim element de la llista dins d'un `` tindrà una decoració de subratllat.

- `:nth-child(n)`. Permet seleccionar elements específics en funció de la seua posició en relació amb els seus germans. Pots usar fòrmules matemàtiques per a definir `n` i seleccionar elements concrets.

```
li:nth-child(odd) {  
    background-color: #f2f2f2;  
}
```

Els elements imparells en una llista `` tindran un fons gris clar.

- `:not(selector)`. Exclou elements que coincideixen amb el selector especificat.

```
p:not(.destacat) {  
    color: #666;  
}
```

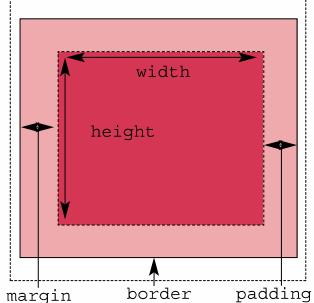
Tots els paràgrafs (`<p>`) que no tenen la classe `destacat` tindran un color de text gris (#666).

3. PROPIETATS I VALORS

CSS. Guía de referencia rápida

Sintaxis	Unidades
<pre>/* Comentario */ selector { propiedad: valor; }</pre>	px em ex % in cm mm pt pc #ff6633 #f63 rgb(0,255,0) rgb(0%,100%,50%) aqua black blue fuchsia gray green lime maroon navy olive purple red silver teal white yellow.
Selectores	Pseudo-clases y pseudoelementos
* Todos los elementos <div> <div> <div "casa"="" "en".="" "foto"="" "x"="" #casa="" >="" <="" <a>="" <div>="" ="" 'x'="" *="" +="" .foto="" <="" <div=""]="" a[class="x" a[class]="" a[class~="x" a[lang="en" atributo="" clase="" class="" comienza="" como="" con="" cuando="" de="" dentro="" div#itemid="" div>="" div.class="" el="" elementos="" ellos="" es="" id="" la="" lang="" lista="" padre="" por="" precedido="" span="" su="" td="" una="" uno="" valores="" y=""><td> :active Elemento seleccionado :focus Elemento enfocado :hover Elemento con el cursor del ratón encima :link Vínculo no visitado :visited Vínculo visitado :first-child Elemento que es el primer hijo de otro elemento :lang Lenguaje utilizado en un elemento :first-letter La primera letra de un texto :first-line La primera línea de un texto :before Contenido antes de un elemento :after Contenido después de un elemento </td></div></div>	:active Elemento seleccionado :focus Elemento enfocado :hover Elemento con el cursor del ratón encima :link Vínculo no visitado :visited Vínculo visitado :first-child Elemento que es el primer hijo de otro elemento :lang Lenguaje utilizado en un elemento :first-letter La primera letra de un texto :first-line La primera línea de un texto :before Contenido antes de un elemento :after Contenido después de un elemento

Propiedad / Valor

Cajas	Posicionamiento	Dimensiones
border (border-left, border-bottom, border-right, border-top) border-width border-style border-color border-width (border-left-width; border-top-width; border-right-width; border-bottom-width) thin medium thick <i>longitud</i> border-color (border-left-color; border-top-color; border-right-color; border-bottom-color) color border-style (border-left-style; border-top-style; border-right-style; border-bottom-style) none hidden dotted dashed solid double groove ridge inset outset margin (margin-top; margin-right; margin-bottom; margin-left) auto <i>longitud</i> padding (padding-top; padding-right; padding-bottom; padding-left) <i>longitud</i>		height auto <i>longitud</i> line-height * normal number <i>longitud</i> max-height none <i>longitud</i> max-width none <i>longitud</i> min-height <i>longitud</i> min-width <i>longitud</i> width auto <i>longitud</i>
<small>* Propiedades que se heredan. En rojo, propiedades de acceso rápido y, entre paréntesis, las propiedades a las que llaman. En cursiva, valores que deben ser incluidos por el usuario.</small>		

Contorno	Tablas
outline <i>outline-color outline-style outline-width</i> outline-color <i>color invert</i> outline-style <i>none dotted dashed solid double groove ridge inset outset</i> outline-width <i>thin medium thick longitud</i>	border-collapse * <i>collapse separate</i> border-spacing * <i>longitud longitud</i> caption-side * <i>top bottom left right</i> empty-cells * <i>show hide</i> table-layout <i>auto fixed</i>
Fondo	Listas
background <i>background-color background-image background-repeat background-attachment background-position</i> background-attachment <i>scroll fixed</i> background-color <i>Color transparent</i> background-image <i>url none</i> background-position <i>top left top center top right center left center center center right bottom left bottom center bottom right x-% y-% x-pos y-pos</i> background-repeat <i>repeat repeat-x repeat-y no-repeat</i>	list-style * <i>list-style-type list-style-position list-style-image</i> list-style-image * <i>none url</i> list-style-position * <i>inside outside</i> list-style-type * <i>none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha</i>
Fuente	Texto
font * <i>font-style font-variant font-weight font-size/line-height font-family</i> caption icon menu message-box small-caption status-bar font-family * <i>family-name generic-family</i> font-size * <i>xx-small x-small small medium large x-large xx-large smaller larger longitud</i> font-size-adjust * <i>none numero</i> font-stretch * <i>normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded</i> font-style * <i>normal italic oblique</i> font-variant * <i>normal small-caps</i> font-weight * <i>normal bold bolder lighter 100 200 300 400 500 600 700 800 900</i>	color * <i>color</i> direction * <i>ltr rtl</i> letter-spacing * <i>normal longitud</i> text-align * <i>left right center justify</i> text-decoration <i>none underline overline line-through blink</i> text-indent * <i>longitud</i> text-shadow <i>none color longitud</i> text-transform * <i>none capitalize uppercase lowercase</i> unicode-bidi <i>normal embed bidi-override</i> white-space * <i>normal pre nowrap</i> word-spacing * <i>normal longitud</i>
Miscelánea I	Miscelánea II
clear <i>left right both none</i> cursor * <i>url auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help</i> content <i>string url counter(name) counter(name, list-style-type)</i> <i>counters(name, string) counters(name, string, list-style-type) attr(X)</i> <i>open-quote close-quote no-open-quote no-close-quote</i> counter-increment <i>none numero identificativo</i> counter-reset <i>none numero identificativo</i>	display <i>none inline block list-item run-in compact marker table</i> <i>inline-table table-row-group table-header-group table-footer-group</i> <i>table-row table-column-group table-column table-cell table-caption</i> float <i>left right none</i> marker-offset <i>auto longitud</i> position <i>static relative absolute fixed</i> quotes * <i>none string string</i> visibility <i>visible hidden collapse</i>

4. EL MODEL CAIXA

Tot en CSS té una caixa al voltant, i comprendre estos caixes és clau per a poder crear dissenys amb CSS o per a alinear elements amb altres elements. En este punt, donarem una ullada més de prop al model de caixes en CSS amb el qual podràs crear dissenys de compaginació més complexos amb una comprensió de com funciona i la terminologia relacionada.

4.1. Caixes en Bloc i en Línia

En CSS, en general, hi ha dos tipus de caixes: caixes en bloc i caixes en línia. Estes característiques es refereixen al mode com es comporta la caixa en termes de flux de pàgina i en relació amb altres caixes de la pàgina:

Si una caixa es defineix com un bloc, es comportarà de les maneres següents:

- La caixa força un salt de línia en arribar al final de la línia.
- La caixa s'estendrà en la direcció de la línia per a omplir tot l'espai disponible que hi haja en el seu contingut. En la majoria dels casos, això significa que la caixa serà tan ampla com el seu contingut, i omplirà el 100% de l'espai disponible.
- Es respecten les propietats `width` i `height`.
- El farciment, el marge i la vora mantenen als altres elements allunyats de la caixa.
- Llevat que decidim canviar el tipus de visualització a en línia, elements com els encapçalats (per exemple, `<h1>`) i tots els elements `<p>` usen per defecte `block` com a tipus de visualització externa.

Si una caixa té una visualització externa de tipus inline, llavors:

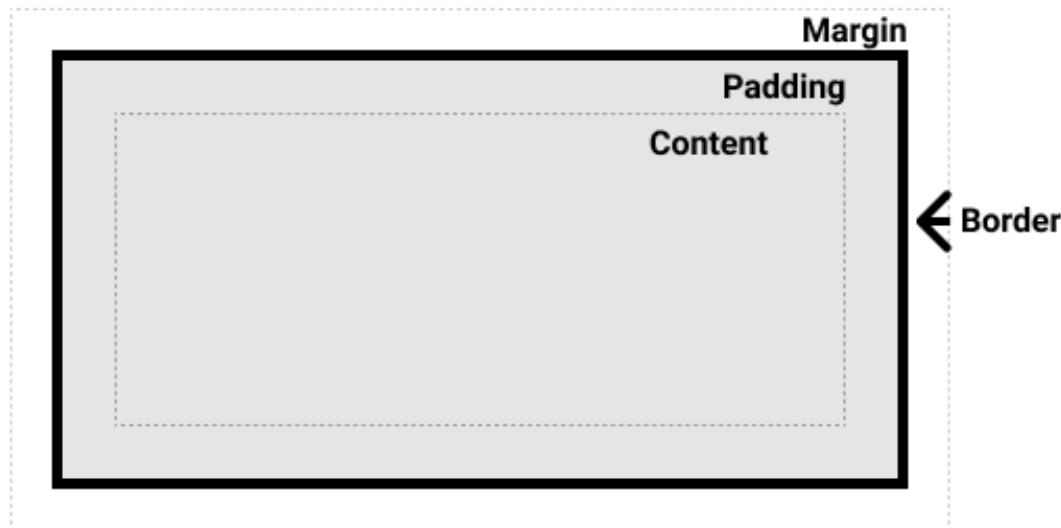
- La caixa no força cap salt de línia en arribar al final de la línia.
- Les propietats `width` i `height` no s'apliquen.
- S'apliquen farcit, marge i vores verticals, però no mantenen allunyades altres caixes en línia.
- S'apliquen farcit, marge i vores horizontals, i mantenen allunyades altres caixes en línia.
- L'element `<a>`, que s'utilitza per als enllaços, i els elements ``, `` i `` són exemples d'elements que es mostren en línia per defecte.

4.2. Parts d'una caixa

En fer una caixa de tipus bloc en CSS tenim els elements següents:

- **El contingut de la caixa (o content box):** L'àrea on es mostra el contingut, la grandària del qual pot canviar-se utilitzant propietats com width i height.
- **El farciment de la caixa (o padding box):** El farciment és espai en blanc al voltant del contingut; és possible controlar la seu grandària usant la propietat padding i altres propietats relacionades.
- **La vora de la caixa (o border box):** La vora de la caixa embolica el contingut i el de farciment. És possible controlar la seu grandària i estil utilitzant la propietat border i altres propietats relacionades.
- **El marge de la caixa (o margin box):** El marge és la capa més externa. Embolica el contingut, el farciment i la vora com a espai en blanc entre la caixa i altres elements. És possible controlar la seu grandària usant la propietat margin i altres propietats relacionades.

El diagrama següent mostra estes capes:



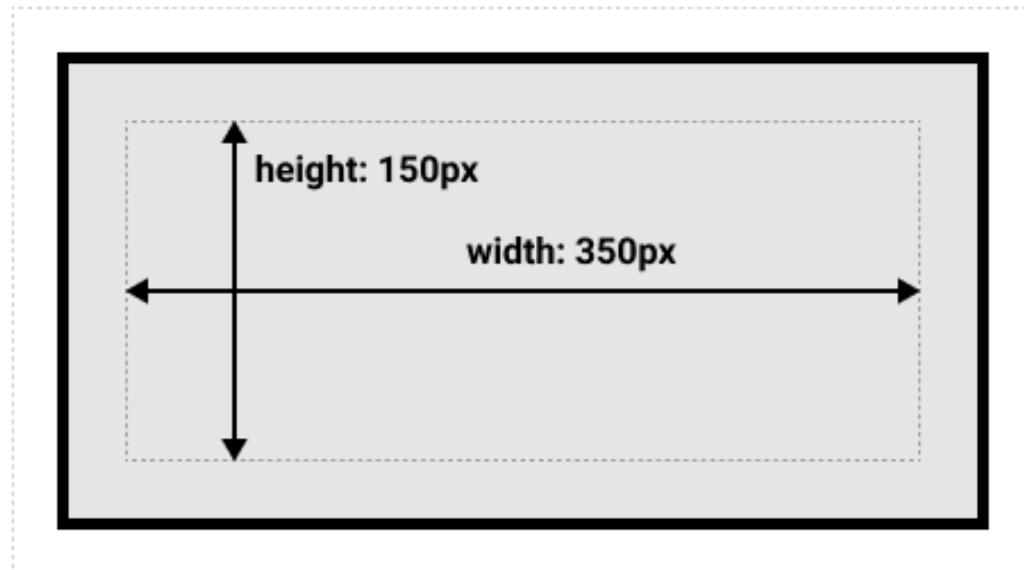
4.3. El model de caixes CSS estàndard

En el model de caixes estàndard, quan estableixes els atributs width i height per a una caixa, defineixes l'ample i l'alt del contingut de la caixa. Qualsevol àrea de farciment i vora s'afig a eixe ample i alt per a obtindre la grandària total que ocupa la caixa. Això es mostra en la imatge que trobaràs a continuació.

Si suposem que la caixa té el CSS següent, que estableix els valors per a les propietats width, height, margin, border, i padding:

```
.box {  
    width: 350px;  
    height: 150px;  
    margin: 10px;  
    padding: 25px;  
    border: 5px solid black;  
}
```

L'espai que ocupa la nostra caixa usant el model de caixes estàndard serà en realitat de 410 px ($350 + 25 + 25 + 5 + 5$); i la seua altura, de 210 px ($150 + 25 + 25 + 5 + 5$), perquè l'àrea de farciment i la vora s'afigen a l'ample que s'utilitza per al contingut de la caixa.



4.4. El model de caixes CSS alternatiu

Podries pensar que és més aviat incòmode haver de sumar la vora i l'àrea de farciment per a obtindre la grandària real de la caixa, i tens raó! Per este motiu, CSS va introduir un model de caixa alternatiu algun temps després del model de caixes estàndard. Amb este model, qualsevol ample és l'ample de la caixa visible en la pàgina, per tant, l'ample de l'àrea de contingut és eixe ample menys l'ample per al farciment i la vora. El mateix CSS que hem usat abans donaria llavors el resultat següent (ample = 350 px, altura = 150 px).



Per defecte, els navegadors usen el model de caixes estàndard. Si desitges activar el model de caixes alternatiu per a un element, fes-lo configurant box-sizing: border-box. Amb això, li dius al navegador que prenga com la vora de la caixa l'àrea definida per qualsevol grandària que establisques.

```
.box {  
    box-sizing: border-box;  
}
```

Si vols que tots els teus elements usen el model de caixes alternatiu (opcio comuna entre els desenvolupadors) has d'establir la propietat box-sizing en l'element <HTML>. Després has de configurar tots els altres elements perquè hereten eixe valor, com es veu en el fragment de codi següent. Si desitges comprendre què hi ha darrere, consulta l'article de CSS-Tricks sobre la grandària de les caixes.

```
HTML {  
    box-sizing: border-box;  
}  
*,  
*::before,  
*::after {  
    box-sizing: inherit;  
}
```