CAPÍTOL 4

CSS: selectors i caixes

dir com es veuen les coses que ja has escrit

PREGUNTA PER A LA CLASSE

«Ja tens cinc pàgines fetes, i totes es veuen igual de tristes.»

Qui decideix com es veuen?

El HTML diu què és. El CSS diu com es veu.

JA HO SAPS FER

HTML

«Això és un títol, això una llista, això una taula.» El **significat**.

AVUI

CSS

«Els títols, blaus; la taula, amb vores.» L'**aspecte**.

Que un <h1> surti gros i negre no ho has decidit tu: és el que fa el navegador quan ningú no li diu res. A partir d'avui l'hi dius tu.

Les teves pàgines no són lletges: és que encara no has dit res.

La regla: tres peces i prou

```
h2 { color: #1750a1; }
```

`h2`	`color`	`#1750a1`
SELECTOR · a qui	PROPIETAT · què	VALOR · com

Les claus `{ }` tanquen el bloc. Els dos punts separen la propietat del valor. El **punt i coma** acaba cada línia.

El punt i coma que falta és l'error que més veuràs.

Les propietats de la primera sessió

`color`	el color del text	`background-color`	el color del fons
`font-family`	la lletra	`font-size`	la mida
`text-align`	esquerra, centre, dreta	`border`	la vora
`padding`	l'aire de dins	`margin`	l'aire de fora

Amb aquestes vuit ja pots canviar com es veu tot el que has fet.

On s'escriu: tres llocs, i només un de bo

ON	COM	QUAN
A l'etiqueta	`style="color: red"`	**Mai.** No es pot reaprofitar
Al <head>	`<style> … </style>`	Per a una prova ràpida
Fitxer a part	`css/estils.css`	**Sempre.** Un full per a tot el web

Amb el fitxer a part, canvies el color una vegada i canvia a **les cinc pàgines**. Recorda el menú de la feina 3.3: aquella pena no la vols una altra vegada.

Avui el farem a part, des del primer minut.

El full d'estil, i on va la carpeta

LA CARPETA, COM AL CAPÍTOL 1

```
aplicacions-web/
├── index.html
├── horari.html
├── proces.html
└── css/
    └── estils.css
```

I AL HEAD DE CADA PÀGINA

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

És una etiqueta **buida**, com la <img>. I la ruta és **relativa**: la del capítol 1.

Al capítol 2 vaig dir que el full d'estil arribaria. Ja hi som.

PREGUNTA PER A LA CLASSE

«Vull que els títols siguin blaus a les cinc pàgines.»

On escrius la regla, i quantes vegades?

Selector d'element: totes les que siguin així

```
li   { font-size: 18px; }
h2   { color: #1750a1; }
table { border-collapse: collapse; }
```

S'escriu el **nom de l'etiqueta**, sense angles. Agafa **totes** les de la pàgina: si poses `li`, canvies els del menú i els del procés alhora.

Potent i perillós: de vegades agafa més del que vols.

Selector de classe: només les que tu marquis

AHIR VAS ESCRIURE AIXÒ

```
<section class="proces">
```

I et vaig dir que no feia res.

AVUI JA FA

```
.proces {
  background-color: #f3f1ec;
}
```

El **punt** vol dir «classe».

Una classe la pots posar a **tantes etiquetes com vulguis**, i una etiqueta pot tenir **més d'una classe**, separades per espais.

És el selector que faràs servir gairebé sempre.

id: existeix, i quasi mai el vols

```
<footer id="peu">     →     #peu { color: grey; }
```

	class	id
Quantes vegades	les que vulguis	**una** per pàgina
Per a què	estils	enllaçar-hi i JavaScript
Al CSS	`.nom`	`#nom`

Per donar estil, class. L'id te'l guardes per a altres coses.

Selector descendent: «els d'aquí dins»

```
.proces li { font-size: 20px; }
```

Un **espai** entre dos selectors vol dir «**dins de**». Això agafa els `<li>` que hi ha dins de la secció del procés, i **deixa en pau** els del menú.

SENSE ESPAI

.proces.gran

Les que tenen **les dues** classes.

AMB ESPAI

.proces li

Els `li` de **dins** de `.proces`.

Un espai ho canvia tot. Mira'l bé.

Agrupar: la mateixa regla per a diversos

```
h1, h2, h3 {
  font-family: Arial, sans-serif;
}
```

La **coma** vol dir «i també». És la manera de no repetir el mateix bloc tres vegades.

Compte amb la diferència: `.proces li` és «els li de dins del procés»; `.proces, li` és «el procés **i** tots els li de la pàgina».

Coma: i també. Espai: de dins.

Prova-ho: el teu horari

```
table {
  border-collapse: collapse;
}
th, td {
  border: 1px solid #999;
  padding: 8px;
  text-align: center;
}
```

Amb aquestes dues regles l'horari passa de ser una llista de paraules a ser una graella. Treu el `border-collapse` i mira què passa: surten **dues** vores.

Prova-ho: el procés selectiu

```
.proces li {
  font-size: 20px;
  padding: 6px;
}
.autoconeixement {
  background-color: #f3f1ec;
  padding: 12px;
}
```

Encara no s'assembla a la pissarra: falten les caixes i col·locar-les. Arriba a les properes sessions.

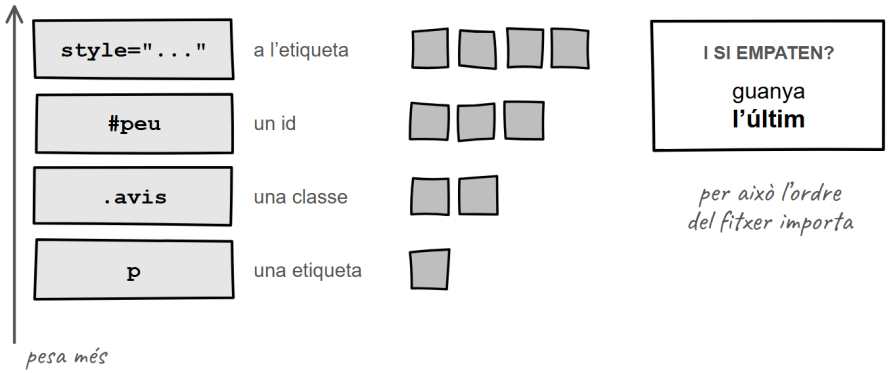
Avui el color i la lletra. Les caixes, la setmana que ve.

PREGUNTA PER A LA CLASSE

«Tinc dues regles que diuen que el mateix paràgraf ha de ser negre i vermell.»

De quin color surt?

La cascada: guanya el més específic



`style=""`	el que més pesa
`#peu`	un id
`.avis`	una classe
`p`	una etiqueta

Si empaten, guanya l'últim

MATEIX PES

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

Surt **verd**: decideix l'ordre.

PESOS DIFERENTS

```
.avis { color: red; }
p { color: green; }
```

Surt **vermell**: la classe pesa més.

I l'herència: les propietats de text baixen als elements de dins; les de caixa, **no**.

Regles generals a dalt, excepcions a baix.

Els colors

COM	EXEMPLE	QUAN
Pel nom	`red`	per provar
Hexadecimal	`#1750a1`	**el que veuràs pertot**
rgb()	`rgb(23, 80, 161)`	quan et donen els números

L'hexadecimal són **tres parelles**: vermell, verd i blau. Si les tres són iguals, surt un gris.

```
#000000 negre     #ffffff blanc     #888888 gris
```

Si es repeteixen, #fff és igual que #ffffff.

Les unitats: px, rem i %

UNITAT	QUÈ ÉS	PER A QUÈ
`px`	una mida fixa	vores i separacions petites
`rem`	vegades la lletra base	**text**
`%`	del que el conté	amplades

La base sol ser 16 píxels: `1.5rem` són 24. Si algú hi veu poc i engrandeix la lletra del navegador, el que has posat en `rem` creix amb ell.

Per a text, rem. Per a vores, px.

Variables: el color en un sol lloc

ES DECLAREN A :ROOT

```
:root {
  --principal: #1750a1;
}
```

I ES FAN SERVIR AMB var()

```
h2 { color: var(--principal); }
```

El nom comença amb **dos guions**. Posa-hi noms que diguin **per a què serveix**, no quin color és: `--principal` envelleix millor que `--blau`.

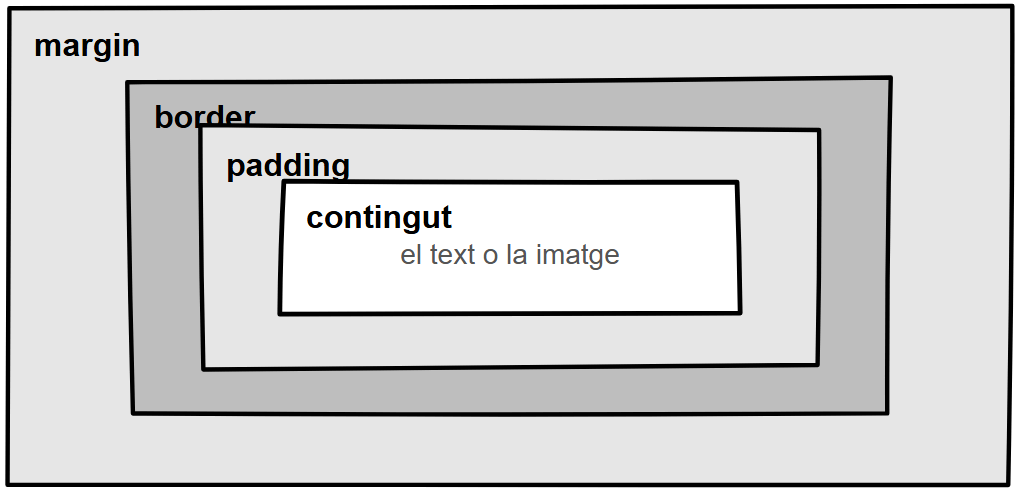
El dia que el canviïs a verd, «--blau» serà una mentida.

PREGUNTA PER A LA CLASSE

«Una caixa amb width 200, padding 20 i vora de 2.»

Quants píxels ocupa a la pantalla?

Tot element és una caixa



el padding és dins de la vora; el margin, fora

si poses fons de color, el padding es pinta i el margin no

El padding és dins de la vora. El margin, fora.

El parany de l'amplada

```
.caixa { width: 200px; padding: 20px; border: 2px; }
```

PER DEFECTE

244px

El `width` només compta el **contingut**: s'hi sumen 40 de padding i 4 de vora.

AMB `border-box`

200px

El `width` compta contingut, padding i vora **tot junt**.

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

Posa-ho a dalt de tot i oblida-te'n.

El teu primer estils.css, en aquest ordre

```
/* 1 */  * { box-sizing: border-box; }
/* 2 */  :root { --principal: #1750a1; }
/* 3 */  body { font-family: Arial, sans-serif; }
/* 4 */  nav a { text-decoration: none; }
/* 5 */  .avis { border: 2px solid var(--principal); }
```

De **més general a més concret**. Les excepcions han de poder guanyar, i amb el mateix pes guanya l'última: si les poses a dalt, les generals se les menjaran.

I comenta'l: d'aquí a dos mesos no te'n recordaràs.

El que has de saber sí o sí

La regla	selector { propietat: valor; } i el punt i coma.
On va	en un fitxer a part, enganxat amb link al head.
Selectors	etiqueta · .classe · #id · espai «de dins» · coma «i també».
Qui guanya	el més específic; si empaten, l'últim.
Unitats	rem per a text, px per a vores, % per a amplades.
Variables	a :root amb dos guions, i var() per fer-les servir.
La caixa	contingut, padding, border i margin; i border-box sempre.