FEINA D'AULA 4.5

Repàs: la caixa, la lletra i el position

quatre hores per tenir-ho clar abans de divendres

Quatre hores, quatre blocs

QUAN	BLOC	EN ACABAR SABRÀS
Dc · 1a hora	**A · La caixa**	quant ocupa de debò una caixa
Dc · 2a hora	**B · La lletra**	passar de píxels a **rem**
Dj · 1a hora	**C · El position**	posar un fill on vulguis dins del pare
Dj · 2a hora	**D · Tot junt**	la targeta de producte i el simulacre

Cada bloc: explicació curta, una **regla d’or** i exercicis. Els de paper són com els de l’examen; els del 💻, a l’ordinador, a `~/projectes/repas-caixes`.

I si un dia no hi ets: els tres vídeos tenen el minut de cada cosa.

PREGUNTA PER A LA CLASSE

«width 200, padding 20, vora de 10 i margin 40.»

Quants píxels ocupa?

Tota caixa té quatre capes



contingut
el text o la imatge: el **width**

padding
l’aire de **dins**; es pinta amb el fons

border
la vora

margin
l’aire de **fora**; no es pinta mai

De dins a fora: contingut, padding, border, margin.

▶ vídeo, 1:30

El parany: el width només és el contingut

```
.caixa { width: 200px; padding: 20px; border: 10px solid; margin: 40px; }
```

```
200 + 20 + 20 + 10 + 10              = 260 px  ← el que es veu
                        + 40 + 40 = 340 px  ← el lloc que ocupa
```

Per defecte, `box-sizing: content-box`: el padding i la vora se sumen **a sobre**, i **dues vegades**, una per banda.

Volies una caixa de 200 i en tens una de 260.

▶ vídeo, 2:58

El truc: border-box a la primera línia

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

Ara el `width` compta **contingut + padding + vora**.

Poses 200 i ocupa 200. El navegador encongeix el contingut:

```
200 - 40 - 20 = 140
```

El **margin** continua anant a part.

Tots els teus CSS comencen així. Tots.

▶ vídeo, 6:13

Les dreceres: com les agulles del rellotge

DECLARACIÓ	VOL DIR
`padding: 10px;`	les quatre bandes
`padding: 10px 20px;`	dalt i baix 10 · costats 20
`padding: 10px 20px 30px;`	dalt 10 · costats 20 · baix 30
`padding: 10px 20px 30px 40px;`	dalt · dreta · baix · esquerra
`margin: 20px auto;`	centrada (cal que tingui width)

Quatre valors: dalt, dreta, baix, esquerra. Com un rellotge.

Ho mirem a l'inspector: l'exercici A1

A LA WEB DEL MÒDUL

Feina 4.5 → Les caixes de l'A1 per a l'inspector. Sis caixes, una per fila de la taula.

PAS 1

F12, tria una caixa, pestanya **Calculat**: passa el ratolí pel dibuix. Fila 1: **260**.

PAS 2

A **Estils**, afegeix `box-sizing: border-box` a la caixa.

L'amplada passa a **200** i el contingut, a **140**. El padding, la vora i el margin no es mouen.

Les respostes de paper, el navegador te les confirma.

▶ vídeo, 1:14

Les dues columnes que no hi caben

```
.columna { width: 50%; padding: 20px; border: 2px solid; }
```

Dues columnes dins d’un pare de **1000 px**.

SENSE border-box

544 + 544 = 1088

500 + 40 + 4 cadascuna. Sobren **88 px**: la segona baixa a sota.

AMB border-box

500 + 500 = 1000

El contingut fa 456. Hi caben **justes**.

Per això el truc va a dalt de tot.

REGLA D’OR A

Padding i vora compten dues vegades. I el border-box, a la primera línia.

Així el `width` que escrius és el que ocupa. El margin sempre va a part.

PREGUNTA PER A LA CLASSE

«El header fa 50 px de lletra, i el seu h1, 50 %.»

El h1 fa 8 px? 25? 50?

Quatre unitats per a la lletra

UNITAT	ES CALCULA A PARTIR DE…	EL PROBLEMA
`px`	res: és fixa	si l’usuari engrandeix la lletra, no creix
`%` `em`	la lletra del **pare**	s’encadenen
`vw`	l’amplada de la pantalla	al mòbil, minúscula
`rem`	la lletra de l’**html**	**cap**

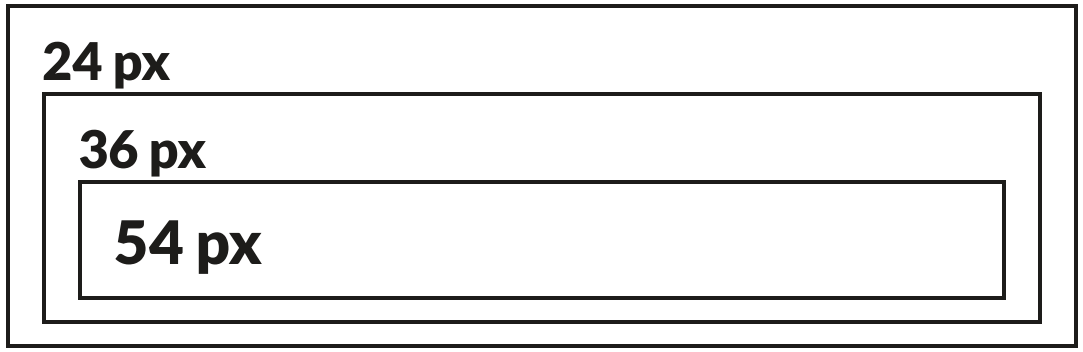
Totes funcionen. Només una no et donarà sorpreses.

▶ vídeo, 30:38

em i % s’encadenen; rem, no

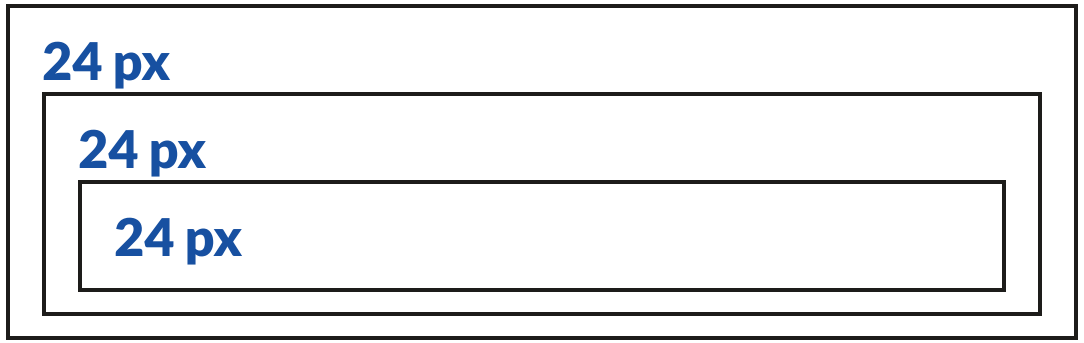
```
.caixa { font-size: 1.5em; }     /* tres .caixa, una dins de l’altra */
```

AMB 1.5em



Cada una multiplica la del **pare**.

AMB 1.5rem



Totes miren l’**html**.

El h1 de la pregunta: 50 % de 50 = 25 px.

▶ vídeo, 32:00

rem: la r és de root

Un `rem` és la lletra de l'element `html`, l'arrel. Valgui el que valgui, val el mateix a tota la pàgina. Si no la toques, **16 px**.

```
rem = píxels ÷ 16

40px → 40 ÷ 16 = 2.5rem
30px → 30 ÷ 16 = 1.875rem
```

Funciona. Però dividir entre 16 de cap...

▶ vídeo, 36:26

El truc del 62,5 %: 1rem = 10px

```
html {
  font-size: 62.5%;  /* 10px */
}
body {
  font-size: 1.6rem; /* 16px */
}
h1 { font-size: 4rem; } /* 40px */
h2 { font-size: 3rem; } /* 30px */
```

El 62,5 % de 16 és **10**. Ara es divideix entre 10, de cap.

El body torna el text normal a 16 px; si no, tot seria petitíssim.

36 px? 3.6rem. Sense pensar.

▶ vídeo, 39:50

L’usuària que hi veu poc

Ha posat la lletra del navegador a **20 px**. La teva web té `html { font-size: 62.5%; }`.

h1 AMB 4rem

50 px

62,5 % de 20 = 12,5. I 4 × 12,5 = 50. **Creix** amb ella.

h1 AMB 40px

40 px

No s’ha mogut. La teva web **no l’escolta**.

Per això el 62,5 % és un percentatge i no 10px.

REGLA D’OR B

La lletra, sempre en rem.

Amb `html { font-size: 62.5%; }`, `1rem = 10px` i el body a `1.6rem`.
Sense el truc, `1rem = 16px`.

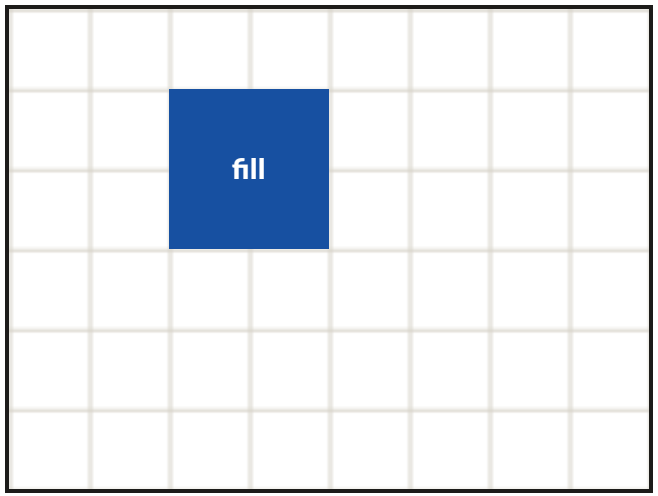
PREGUNTA PER A LA CLASSE

«Vull l’etiqueta -20 % a la cantonada de dalt de la foto.»

Com la poso exactament allà?

Pare relative, fill absolute

```
.pare {
  position: relative;
}
.fill {
  position: absolute;
  top: 50px;
  left: 100px;
}
```



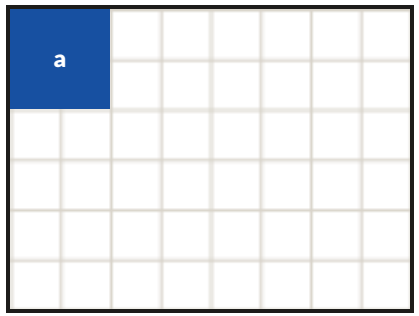
El pare fa 400 × 300; cada quadre, 50 px.

El **pare** diu «mesura des de mi». El **fill** es col·loca a aquella distància de les vores del pare.

Dues caixes, dues paraules.

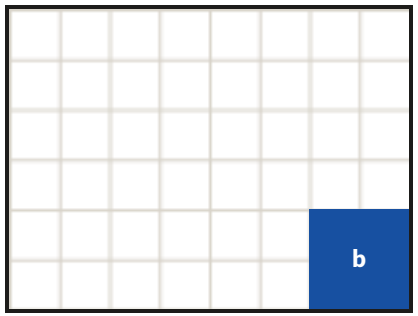
▶ vídeo, 2:09

top, left, bottom i right: distància a la vora del pare



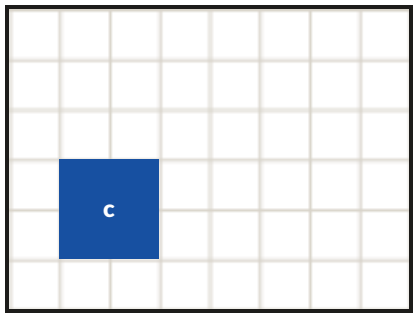
a)

```
top: 0;
left: 0;
```



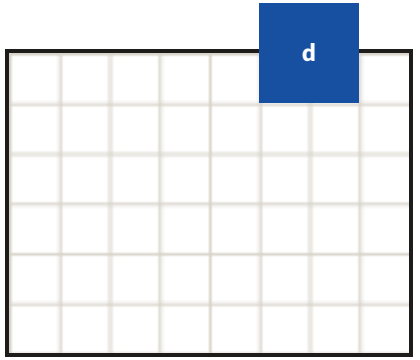
b)

```
bottom: 0;
right: 0;
```



c)

```
bottom: 50px;
left: 50px;
```



d)

```
top: -50px;
right: 50px;
```

top + **left** o **bottom** + **right**, mai top i bottom alhora. Els **negatius** treuen el fill fora. I sempre amb `px`: `top: 10` no val.

El fill absolute surt del flux: el de sota puja.

▶ vídeo, 4:40

Troba els tres errors

```
.targeta  { width: 300px; height: 200px; }
.etiqueta { position: relative; top: 10; right: 10px; }
```

Volia l’etiqueta a 10 px de la cantonada de dalt a la dreta. Surt a dalt de la pàgina.

1	A `.targeta` li falta `position: relative`.
2	`.etiqueta` ha de ser `absolute`, no `relative`.
3	`top: 10` no té unitat: `top: 10px`.

«Se me’n va a la cantonada de la pantalla»: al pare li falta el relative.

El fill que tapa tot el pare: un vel sobre la foto

```
.foto {
  position: relative;
  height: 300px;
  background-image: url(../img/bosc.jpg);
  background-size: cover;
}
```

```
.vel {
  position: absolute;
  top: 0;  left: 0;
  width: 100%;  height: 100%;
  background-color: rgba(0, 0, 0, 0.5);
  color: white;
}
```

`rgba` té un quart número, l'**alfa**: 1 opac, 0 invisible, 0.5 mig transparent. El `100%` és del **pare**, que és qui té el relative.

El text blanc es llegeix, sigui quina sigui la foto.

▶ vídeo, 6:33

REGLA D’OR C

Pare relative, fill absolute, i el fill es mou des de les vores del pare.

Amb `top/left` o `bottom/right`. Si se te’n va a la cantonada de la pantalla, al pare li falta el relative.

Tot junt: la targeta de producte



A	els dos trucs a dalt; `.producte` de **300 px justos** amb padding i vora
B	body a `1.6rem`, h2 a 24 px i preu a 28 px, **en rem**
C	`.producte` pare de `.oferta`; `.foto` pare de `.vel`
Comprova	l'inspector diu **300**; sense el truc en serien 336

45 minuts. L'HTML el tens a la fitxa.

Simulacre d’examen

1 · CALCULA

Amplada visible i contingut amb border-box d’una caixa.

2 · CONVERTEIX

Píxels a rem, amb el truc del 62,5 % i sense.

3 · VERTADER O FALS

Quatre afirmacions, i per què.

4 · ESCRIU EL CSS

Una icona a la cantonada d’un botó.

15 minuts, sense ordinador ni apunts. Després el corregim entre tots.

Com divendres, però avui no compta.

El que has de saber sí o sí

La caixa	padding i vora × 2; border-box a dalt de tot.
Les dreceres	dalt, dreta, baix, esquerra.
La lletra	en rem; 62,5 % i 1rem = 10px.
em i %	miren el pare: s’encadenen.
El position	pare relative, fill absolute.
El vel	top 0, left 0, 100 %, i rgba.