FEINA D'AULA 5.3

Black Friday

una botiga de zero: amb Flexbox, i després amb Grid

Avui

PART	QUÈ	EN ACABAR SABRÀS
A	**L’HTML**	muntar una targeta de producte i la pàgina sencera
B	**Flexbox**	`flex-direction: column`, el marge `auto` i la llista amb `calc()`
C	**Grid**	la mateixa llista amb `grid-template-columns` i `fr`
10 min	**Lliurament**	commit, push i dues captures

Una carpeta nova, `black_friday/`. **De zero**: no aprofites res de la galeria.

Una pàgina, dues maneres.

PREGUNTA PER A LA CLASSE

«Una fila de botons, o una quadrícula de 12 productes.»

Flexbox o Grid?

Flexbox o Grid?

FLEXBOX

una línia

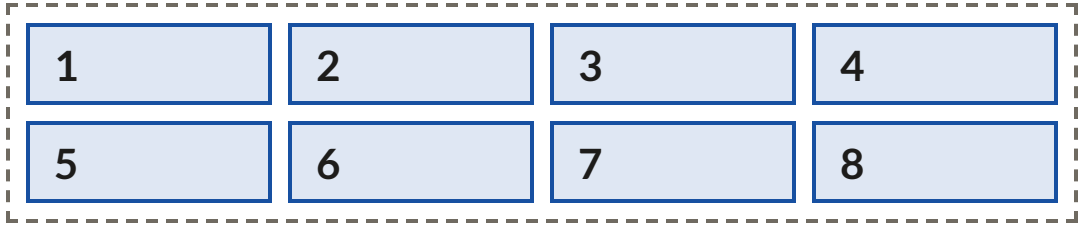
Una fila **o** una columna. La mida la decideixen els **fills**.



GRID

una quadrícula

Files **i** columnes alhora. La mida la decideix el **pare**.



No són enemics: avui la llista és Grid, i la targeta per dins, Flexbox.

La targeta, peça a peça

```
<article class="producte">
  <img src="img/ratoli.jpg" alt="…">
  <h2>Ratolí sense fil</h2>
  <p class="preus">
    <del>39 €</del>
    <strong>25 €</strong>
    <span class="descompte">-36 %</span>
  </p>
  <a class="boto" href="#">Afegeix…</a>
</article>
```

`article`	una peça que té sentit tota sola
`del`	text que ja no val: surt ratllat
`strong`	important: el preu d'ara
`span`	cap significat: un ganxo per a l'estil
`a`	un enllaç; encara no hi ha cistella

12 targetes iguals dins d'un div.productes.

La targeta: Flexbox en columna

```
.producte {
  display: flex;
  flex-direction: column;
}
.boto {
  margin-top: auto;
}
```

Els fills van **l'un sota l'altre**. Les targetes d'una fila fan **la mateixa alçada**: el `margin-top: auto` es menja l'espai que sobra i baixa el botó.

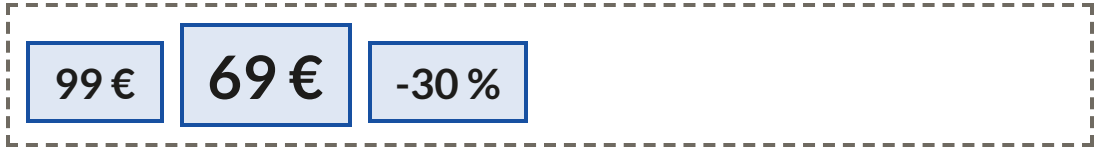


Exercici E1.

El marge auto empeny

```
.preus {
  display: flex;
  align-items: baseline;
  gap: 8px;
}
.descompte { margin-left: auto; }
```

SENSE margin-left: auto



AMB margin-left: auto



En un ítem flex, un marge **auto** ocupa tot l’espai lliure. A l’esquerra empeny cap a la dreta; a dalt, cap a baix.

baseline: alineats per la base de la lletra.

La llista amb Flexbox

```
.productes {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
}
.producte { width: 100%; }
```

A PARTIR DE	PER FILA	WIDTH DE LA .PRODUCTE
—	**1**	`100%`
`576px`	**2**	`calc((100% - 20px) / 2)`
`768px`	**3**	`calc((100% - 40px) / 3)`
`992px`	**4**	`calc((100% - 60px) / 4)`

Forats = productes per fila − 1.

PREGUNTA PER A LA CLASSE

«Tenim 12 productes, i 1, 2, 3 o 4 per fila.»

Per què 12?

La resposta

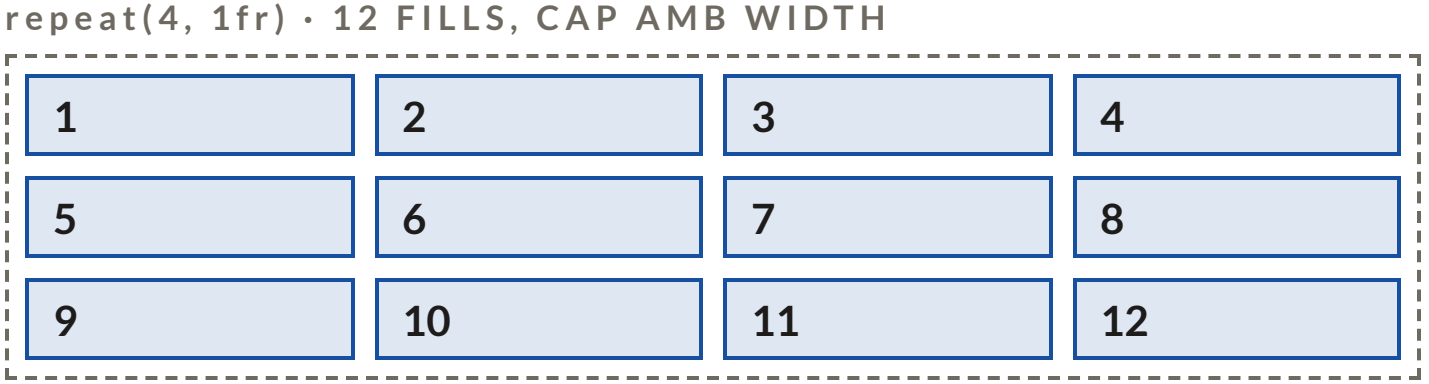
PER FILA	FILES	L’ÚLTIMA FILA
2	**6**	plena
3	**4**	plena
4	**3**	plena
5	**3**	5 + 5 + **2**: coixa

12 es divideix entre 1, 2, 3 i 4: a totes les amplades, les files queden plenes.

Quan triïs quants per fila, mira quants n’hi ha.

Grid: el pare dibuixa les columnes

```
.productes {
  display: grid;
  grid-template-columns:
    repeat(4, 1fr);
  gap: 20px;
}
```



Els fills entren a les cel·les d’esquerra a dreta i de dalt a baix. **No** porten `width`.

repeat(4, 1fr) = 1fr 1fr 1fr 1fr.

La unitat fr: una part del que queda

grid-template-columns: 1fr 1fr 1fr



grid-template-columns: 1fr 2fr 1fr



Primer el navegador resta els gaps; el que queda, ho reparteix en parts. **Per això ja no cal `calc()`.**

fr = fracció.

De flex.css a grid.css: tres canvis

FLEX.CSS

```
.productes {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
}
.producte { width: 100%; }
@media (min-width: 576px) {
  .producte {
    width: calc((100% - 20px) / 2);
  }
}
```

GRID.CSS

```
.productes {
  display: grid;
  grid-template-columns: 1fr;
  gap: 20px;
}
.producte { }   /* sense width */
@media (min-width: 576px) {
  .productes {
    grid-template-columns: repeat(2, 1fr);
  }
}
```

La targeta per dins no canvia: continua sent Flexbox.

On és la mida?

FLEXBOX

als fills

Les media queries canvien la `.producte`: cada targeta diu quant fa, i resta els forats amb `calc()`.

GRID

al pare

Les media queries canvien la `.productes`: el pare diu quantes columnes hi ha, i els fills hi entren.

Exercici E4.

Repte: una quadrícula sense media queries

```
.productes {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(220px, 1fr));
  gap: 20px;
}
```

auto-fill

tantes com càpiguen

El navegador posa totes les columnes que hi caben…

minmax(220px, 1fr)

mínim 220 px

…de 220 px com a mínim, i les estira per omplir la fila.

Responsive sense cap @media. Però no tries tu quantes per fila.

REGLA D’OR

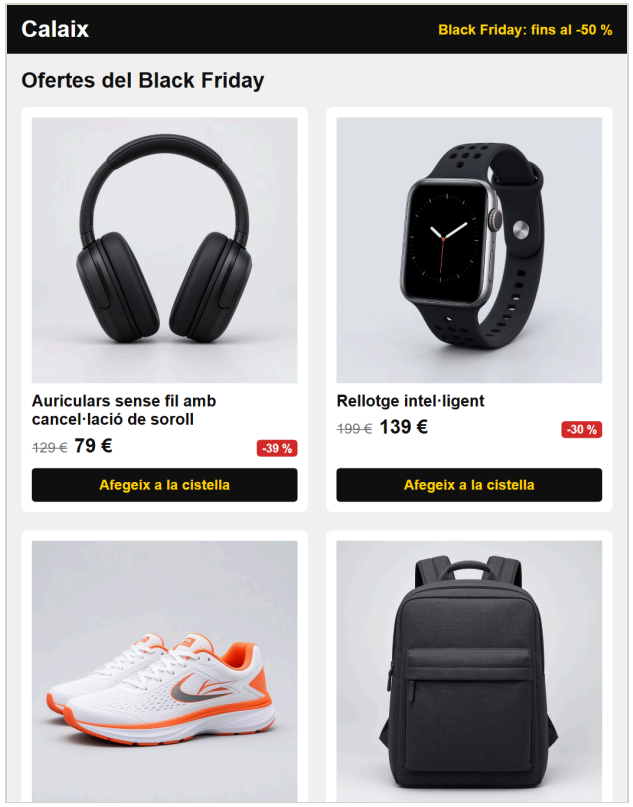
Flexbox per a una línia.
Grid per a una quadrícula.

A Flexbox, la mida la porten els fills. A Grid, la dibuixa el pare amb `grid-template-columns`. I sovint, tots dos junts.

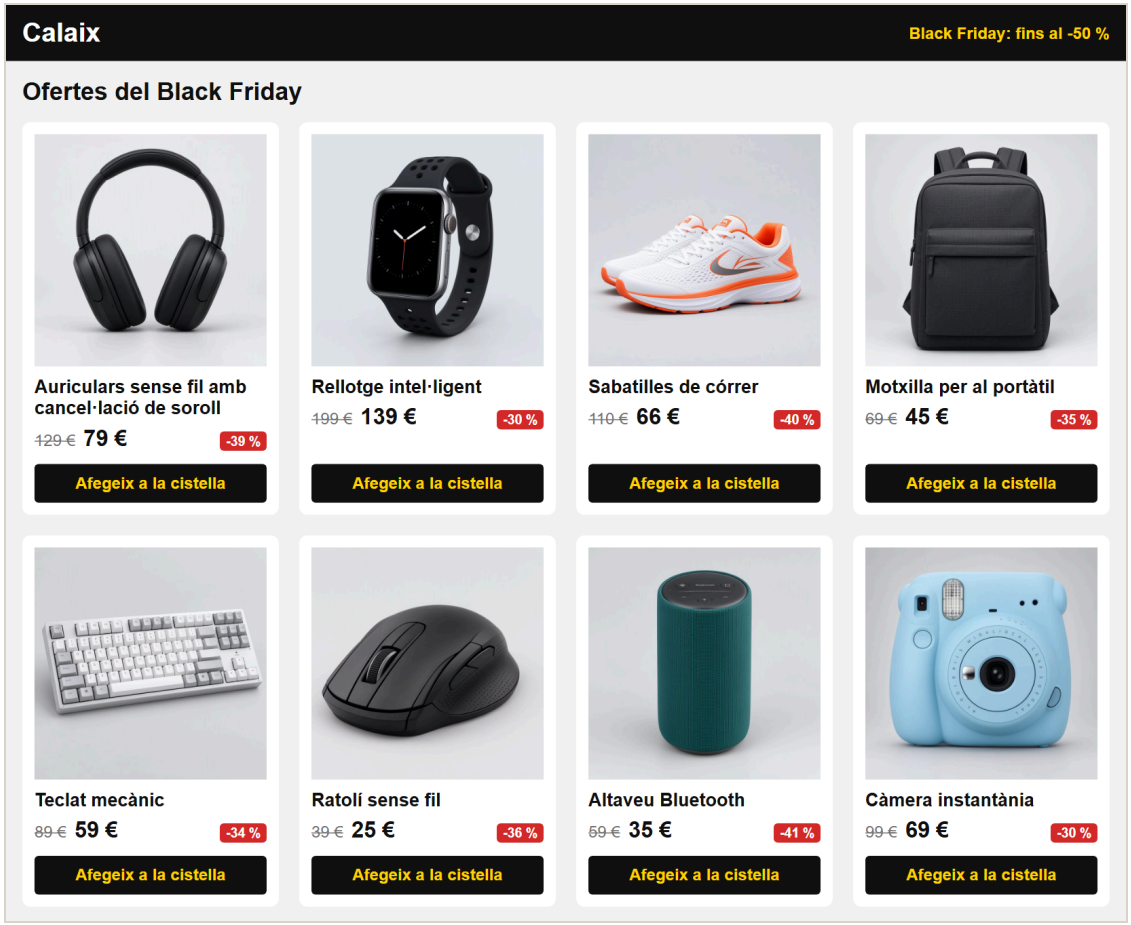
Les tres amplades, i flex i grid iguals



375 PX · 1



700 PX · 2



1100 PX · 4

Abans d’acabar

- [ ] Les 12 fotos, amb el seu `alt`, a `flex.html` i a `grid.html`.
- [ ] 1, 2, 3 i 4 per fila, igual a les dues versions.
- [ ] Els botons d’una fila, tots a la mateixa altura; el descompte, a la dreta.
- [ ] `grid.css` sense cap `width` a la `.producte` ni cap `calc()`.
- [ ] `git commit -m "5.3: Black Friday amb Flexbox i Grid"` i `git push`
- [ ] Dues captures al Classroom: a 375 px i a 1100 px.

El que has de saber sí o sí

flex-direction	column: els fills, l’un sota l’altre.
margin auto	en un ítem flex, empeny fins a la vora.
display: grid	al pare: una quadrícula.
grid-template-columns	quantes columnes, i de quina mida.
fr	una part del que queda després dels gaps.
repeat(4, 1fr)	1fr 1fr 1fr 1fr.