Feina d'aula 5.1
display i media queries

la botiga de l'examen, que es vegi bé també al mòbil

SESSIÓ	CARPETA	MATERIAL	DESPRÉS
Dj 05/11 · 2 h	**El repositori de l'examen**	**VS Code i les DevTools**	**Flexbox · dv 06/11**

Per què aquesta feina. La web de l'examen es veia bé a l'ordinador. Obre-la ara com si fos un mòbil: l'article de 400 px no hi cap, el menú té pics i la foto surt per fora. Avui aprendràs les dues eines que ho arreglen: `display`, que diu **com es col·loca cada caixa**, i les **media queries**, que donen un CSS diferent segons l'amplada de la pantalla.

QUAN	BLOC	QUÈ HAS DE SABER FER EN ACABAR
1a hora	A · `display`	Distingir block i inline, i fer un menú amb caixes clicables
2a hora	B · Media queries	Fer que la botiga canviï de forma entre el mòbil i l'ordinador
Últims 10 min	C · Lliurament	Commit, push i dues captures al Classroom

La carpeta. Treballa sobre el **teu repositori de l'examen**, `llinatge1_llinatge2_nom_examen1`: obre'l amb VS Code i Live Server, com sempre. Si no el tens complet, baixa del Classroom `feina_5_1_inici.zip`: és la web resolta de l'examen. Descomprimeix-la dins del teu repositori.

BLOC A · DISPLAY

1a hora

Cada element HTML és una caixa, i el navegador la col·loca d'una de dues maneres. Això és el `display` per defecte de cada element.

Tres `<p>` : block

primer

segon

tercer

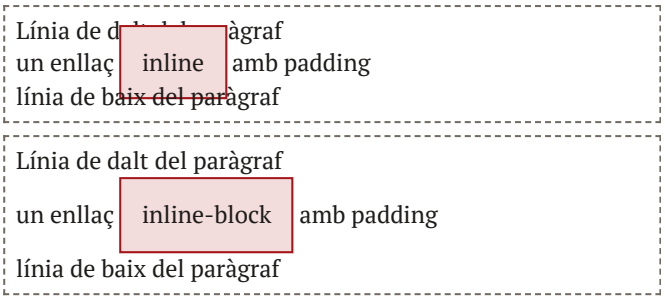
Tres `<a>` dins d'un paràgraf: inline

Text del paràgraf primer segon tercer i el text continua a la mateixa línia, com les paraules.

DISPLAY	COM ES COL·LOCA	WIDTH I HEIGHT	MARGIN I PADDING	PER DEFECTE, A…
`block`	Comença una línia nova i ocupa **tota l'amplada** del pare	Sí	Sí, per tots els costats	`div p h1…h6 ul ol li header nav main article footer form table`
`inline`	Va **seguit**, dins del text, i fa el que fa el seu contingut	**No**: no en fa cas	Als costats sí; a dalt i a baix no aparta res	`a strong em span code label`
`inline-block`	Va seguit, com un inline…	…però sí, com un block	Sí, per tots els costats	`img input button` s'hi assemblen
`none`	No es veu i **no ocupa lloc**: com si no hi fos	No és el mateix que `visibility: hidden`, que el fa invisible però li guarda el lloc		

El parany: el padding d'un inline no aparta les línies

Si a un `a` (inline) li poses `padding: 10px`, el fons creix, però les línies de dalt i de baix no s'aparten: se superposen. Amb `inline-block`, el padding sí que empeny. Mira-ho a la dreta: és el mateix enllaç amb el mateix padding.



EXERCICI A1 · BLOCK O INLINE?

Escriu el `display` per defecte. Després, pensa què passa si en poses dos seguits a l'HTML.

ELEMENT	DISPLAY	SI EN POSES DOS SEGUITS…
`<h2>`		
`<a>`		
`<li>`		
`<strong>`		
`<article>`		
`<span>`		

EXERCICI A2 · 💻 EL MENÚ, AMB CAIXES CLICABLES

Ara el menú és una llista amb pics, i només pots clicar damunt de les lletres. Afegeix això al teu `css/estils.css`:

```
nav ul {
  list-style: none;      /* fora els pics */
  margin: 0;
  padding: 0;            /* fora el sagnat de la llista */
}
nav a {
  display: block;        /* l'enllaç passa a ser una caixa */
  padding: 10px 16px;
}
```

Prova-ho: ara pots clicar a qualsevol lloc de la franja, no només damunt del text. Esborra el `display: block` i torna-ho a provar.

Per què hem posat `display: block` a l'enllaç i no l'hem deixat com estava?

EXERCICI A3 · 💻 NONE O HIDDEN, A L'INSPECTOR

Obre l'inspector (F12), selecciona el primer `article` i, a la pestanya *Styles*, afegeix-li `display: none`. Després canvia-ho per `visibility: hidden`. Aquests canvis no es guarden: en recarregar la pàgina, tot torna a ser com abans. Què passa amb el segon article en cada cas?

REGLA D'OR A

block = una caixa per línia, que accepta totes les mides. **inline** = va amb el text i no fa cas del `width`, del `height` ni dels marges de dalt i de baix. Si vols caixes seguides *i* amb mides, fes servir **inline-block**.

Saps per què el segell de «Nou» de l'examen, que és un `span` (inline), acceptava el padding com una caixa? Perquè un element amb `position: absolute` passa a comportar-se com un block.

BLOC B · MEDIA QUERIES

2a hora

B0 · Mira la teva web com un mòbil

A les DevTools (F12), prem **Ctrl + Shift + M**: és el **mode dispositiu**. A dalt, tria una amplada de 375 px, que és un mòbil normal. Si arrossegues la vora de la pantalla, veuràs la web amb qualsevol amplada.

Recordes el `<meta name="viewport">` de la plantilla, que comptava a l'examen? És el que diu al mòbil «fes servir la teva amplada de debò». Sense ell, el mòbil fa veure que és una pantalla de 980 px, ho dibuixa tot i ho encongeix: la lletra surt minúscula.

A la dreta hi ha la web de l'examen en un mòbil: la foto i l'article surten per la dreta, i el menú té pics.



Abans: la web de l'examen a 375 px

B1 · Com s'escriu una media query

```
@media (min-width: 768px) {
  /* les regles d'aquí dins només s'apliquen
     si la finestra fa 768 px d'ample o més */
  h1 { font-size: 3.2rem; }
}
```

- **Mòbil primer.** Fora de cap media query escrius el CSS del mòbil. Dins d'una `@media (min-width: …)` afegeixes només **el que canvia** quan la pantalla és més ampla.
- **Els punts de tall** (*breakpoints*) habituals són 576 px (mòbil gran), 768 px (tauleta), 992 px (ordinador) i 1200 px (pantalla gran). Són els de Bootstrap, que veurem al capítol 6.
- Una media query **no té més força** que una regla normal: si dues regles empaten, guanya l'última. Per això les media queries van **al final** del CSS.
- `max-width` és el contrari: «fins a tants píxels». Si fas mòbil primer, gairebé no el necessites.

EXERCICI B1 · QUINA MIDA TÉ?

```
h1 { font-size: 2.6rem; }
@media (min-width: 768px)  { h1 { font-size: 3.2rem; } }
@media (min-width: 1200px) { h1 { font-size: 4rem; } }
@media (max-width: 500px)  { .eslogan { display: none; } }
```

L'html té `font-size: 62.5%`. Omple la taula.

AMPLADA DE LA FINESTRA	MIDA DEL H1	ES VEU L'ESLÒGAN?
375 px (mòbil)		
800 px (tauleta)		
1300 px (ordinador)		

I si les dues media queries del h1 estiguessin a l'inrevés (primer la de 1200 i després la de 768), quina mida tindria a 1300 px?

EXERCICI B2 · 💻 LA BOTIGA, MÒBIL PRIMER

1. El mòbil. A la regla de l'`article`, **esborra el** `width: 400px`: com que és un block, ja ocupa tota l'amplada. Perquè la foto no surti per fora, fes que ocupi l'amplada de l'article. Separa els articles del menú:

```
article img {
  width: 100%;
}
main {
  padding: 20px 16px 0;
}
```

2. Un eslògan que només es veu a l'ordinador. Afegeix-lo a l'HTML, sota el `h1` del `header`, a les dues pàgines. Al CSS, amaga'l: aquesta regla és la del mòbil.

```
<p class="eslogan">Venda i lloguer de bicicletes a Sóller</p>

.eslogan {
  display: none;
}
```

3. L'ordinador. A **baix de tot** del CSS, el que canvia a partir de 768 px:

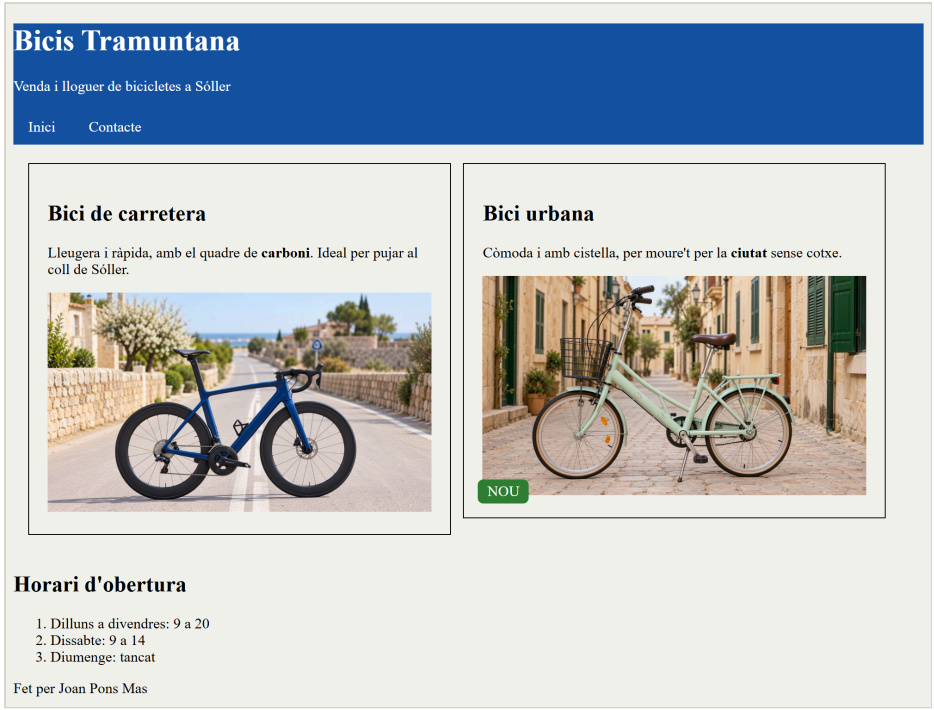
```
@media (min-width: 768px) {
  .eslogan {
    display: block;          /* ara sí que hi cap */
  }
  nav li {
    display: inline-block;   /* els enllaços del menú, un al costat de l'altre */
  }
  article {
    display: inline-block;   /* els dos articles, un al costat de l'altre */
    width: 48%;
    margin-right: 1%;
    vertical-align: top;     /* que s'alineïn per dalt */
  }
}
```

4. Prova-ho. Al mode dispositiu, arrossega l'amplada a poc a poc: a 767 px els articles van un a sota de l'altre i, a 768 px, salten un al costat de l'altre.

Per què hem posat 48 % i no 50 %? Prova-ho amb 50 %.



Després, a 375 px



Després, a 1000 px

REGLA D'OR B

Escriu primer el CSS del **mòbil**. Després, a **baix de tot**, una `@media (min-width: 768px)` amb **només el que canvia**. I prova sempre la web amb el mode dispositiu (Ctrl + Shift + M) abans de dir que està acabada.

BLOC C · LLIURAMENT

- [ ] El menú no té pics, i cada enllaç és una franja clicable.
- [ ] A 375 px, res surt per la dreta: ni els articles ni les fotos.
- [ ] A 768 px o més, el menú va en una línia, surt l'eslògan i els dos articles van un al costat de l'altre.
- [ ] La media query és a baix de tot del CSS i només té el que canvia.
- [ ] Has fet `git add .`, `git commit -m "Feina 5.1: display i media queries"` i `git push`.
- [ ] Has lliurat al Classroom dues captures del mode dispositiu: una a 375 px i una a 1000 px.

Demà, Flexbox: els dos articles un al costat de l'altre, sense el truc del 48 %.