



MÀSTER UNIVERSITARI DE
DESENVOLUPAMENT DE LLOCS I APLICACIONS WEB

HTML I CSS

PEC 3

Gerard Fauchs Masgrau

Índex

1	Pregunta 1. Sobre layouts CSS	2
1.1	Expliqueu les diferències entre els elements flexbox i els elements grid.	2
1.2	Expliqueu quins avantatges aporta emprar elements flexbox en contrast amb els mètodes emprats anteriorment, com float.	3
1.2.1	CSS Float	3
1.2.2	CSS Flexbox	4
2	Pregunta 2. Sobre disseny responsive.	5
2.1	Expliqui en breus paraules que és el disseny responsive (responsive design).	5
2.2	Describeu detalladament l'estructura emprada en CSS per a aplicar disseny responsive. . .	5
2.3	Explica què podem fer per a facilitar que la pàgina sigui responsive abans de fer el pas d'afegir media queries.	5
2.3.1	Multicol	6
2.3.2	Flexbox	6
2.3.3	Cuadrículas CSS	6
2.3.4	Imatges responsive	6
2.3.5	Tipografia responsive	6
2.3.6	La metaetiqueta viewport	7
3	GitHub Pages	8

1 Pregunta 1. Sobre layouts CSS

1.1 Expliqueu les diferències entre els elements flexbox i els elements grid.

La diferència bàsica entre CSS Grid Layout i CSS Flexbox Layout és que Flexbox es va crear per a dissenys d'una dimensió, en una fila o una columna. En canvi CSS Grid Layout es va pensar per al disseny bidimensional, en diverses files i columnes al mateix temps. No obstant això, les dues especificacions comparteixen algunes característiques comunes.

Hi ha una pregunta molt simple per a decidir si utilitzar grid o flexbox:

- **Necessito controlar el disseny únicament per files o per columnes?** Usa flexbox.
- **Necessito controlar el disseny per files i per columnes?** Usa grid.

Flexbox treballa des del contingut. Un cas d'ús ideal per a flexbox és quan tens un conjunt d'ítems i vols espaiar-los uniformement en un contenidor. Deixes que la grandària del contingut decideixi quant espai ocupa cada ítem. Si els ítems s'emboliquen (wrap) en una nova línia, calcularan el seu espaiat basant-se en la seva grandària i l'espai disponible en aquesta línia.

Grid funciona des del layout cap a dintre. Quan uses CSS Grid Layout creas un disseny i després col·loques elements en ell, o permetes que les regles d'auto-placement col·loquin els elements en les cel·les de la quadrícula d'acord amb aquesta quadrícula estricta. Existeix la possibilitat de crear pistes que responguin a la grandària del contingut, no obstant això, també canviaran tota la pista.

La propietat flex-wrap de CSS especifica si els elements “fills” són obligats a romandre en una mateixa línia o poden fluir en diverses línies. **És una propietat opcional Flexbox.**

Grid també pot lliscar elements opcionalment (si permetem l' auto filling) en el sentit que els elements poden omplir una línia i moure's a la nova (o autocol·locar-se). Però en fer-ho, cauran al llarg de les mateixes línies del grid.

Es pot pensar en flexbox com a un element “unidimensional”.

Flexbox pot fer files i columnes en el sentit que permet que els elements flueixin i no hi ha manera de controlar on acabaran ja que els mateixos simplement empenyen al llarg d'un sol eix i després es mouen en forma concorde al codi.

De fet els elements es desplacen al llarg d'un pla unidimensional i és per aquesta única dimensió que podem moure els elements, **com alinear-los al llarg d'una línia de base, que és una cosa que el Grid no pot fer.**

El Grid és “bidimensional” en el sentit que podem (si volem) declarar la grandària de les files i columnes i després col·locar explícitament les coses en totes dues files i columnes com vulguem.

El Grid es defineix majorment en l'element pare. En el flexbox, la major part del disseny (més enllà del bàsic) es realitza en els fills.

El Grid és millor en la superposició dels elements.

Aconseguir que els elements se solapin en el flexbox requereix estudiar el css tradicional, com els marges negatius, les transformacions o el posicionament absolut per a sortir del comportament del flexbox. Amb el Grid, podem superposar els elements en les línies, o fins i tot dins de les mateixes cel·les.

El Grid és més ferm.

Mentre que la flexió dels elements flexbox és a vegades la seva força, la forma en què es gestionen les dimensions és bastant complicada. És una combinació d'amplària, amplària mínima, amplària màxima, base de flexió, creixement de flexió i contracció de flexió, sense esmentar el contingut interior i coses com els espais en blanc, així com els altres elements de la mateixa fila.

El Grid té interessants característiques per a “ocupar els espais”, com les unitats fraccionàries, i la capacitat que el contingut trenqui les quadrícules, encara que, en termes generals, estem configurant línies de quadrícula i col·locant elements dins d'elles que es despleguen en el seu lloc.

Flexbox pot “empènyer” els elements. És una característica bastant única de flexbox que pot, per exemple, posar `margin-right: auto;` a un element i, si hi ha espai, aquest element empenyerà tota la resta tan lluny com pugui.

1.2 Expliqueu quins avantatges aporta emprar elements flexbox en contrast amb els mètodes emprats anteriorment, com float.

1.2.1 CSS Float

La float propietat CSS permet al desenvolupador incorporar columnes de tipus taula en un disseny HTML sense l'ús de taules.

Si no fos per la propietat flotant de CSS, els dissenys de CSS no serien possibles si no s'utilitza un posicionament relatiu i absolut, cosa que seria desordenat i faria que el disseny no es pogués mantenir. El propòsit de la float propietat CSS és, en general, empènyer un element de nivell de bloc cap a l'esquerra o cap a la dreta, traient-lo del flux en relació amb altres elements de bloc. Això permet que el contingut que flueix de manera natural s'envolti sobre l'element flotat.

En general, un element flotant hauria de tenir una amplada establerta explícitament (tret que sigui un element substituït, com una imatge). D'aquesta manera es garanteix que el comportament flotant es comporta com s'esperava i ajuda a evitar problemes en determinats navegadors.

Per tal d'aconseguir un disseny de disseny per a navegadors que no entenen la flexbox ni la grid, fem servir float. S'utilitza en elements de bloc, és a dir, que tenen `display: block`. Ha valors com `left`, `right`, `none` i `inherit`.

Es pot fer flotar tants elements com vulgueu, un darrere l'altre, de manera que tots s'adaptin a tota l'amplada disponible. Tot i això, aquesta estratègia té alguns inconvenients:

- és difícil gestionar les altures dels elements del contenidor.
- Si es necessita, el centratge vertical pot resultar dolorosament difícil de gestionar.
- en funció del contingut que tingueu dins dels elements (juntament amb les propietats CSS de cada contenidor intern), el navegador pot ajustar elements a la següent línia gratuïta i trencar el disseny.

1.2.2 CSS Flexbox

El mòdul de caixa flexible s'ha dissenyat com un model de presentació unidimensional i com un mètode que pot proporcionar distribució de l'espai entre els elements de la interfície i potents funcions d'alineació. Quan s'utilitza flexbox, es pot aconseguir qualsevol disseny desitjat sense utilitzar display: block/inline-block, float o position per controlar el disseny, perquè flexbox té funcions natives per a aquestes propietats.

Flexbox funciona en una relació "pare-fill", que significa que es display: flex s'aplica als elements pares (flex-containers), amb les declaracions adequades que s'utilitzen per controlar el comportament dels elements fills (flex-items). Un flex-items també pot ser un flex-container al mateix temps, per crear més jerarquia i subdivisió del disseny de flexió, sovint en la direcció ortogonal. Per aconseguir el disseny desitjat, podem fer ús de propietats com ara justify-content i align-items per distribuir els elements de contingut al llarg dels eixos flexibles. La direcció de l'eix de flexió principal també es pot modificar mitjançant la direcció de flexió per ajudar a tornar a transmetre el contingut més enllà de certs punts d'interrupció si cal.

El mòdul de caixa flexible, conegut amb el nom de Flexbox , és un model de disseny més recent capaç de distribuir espai i alinear poderosament els elements d'un contenidor (la caixa) d'una manera unidimensional.

A més de que l'enfocament flotant és molt popular i és generalment adoptat pels marcs de quadrícula populars, Flexbox presenta una sèrie de beneficis respecte al float:

- alineació vertical i igualtat d'alçada per als elements del contenidor a cada fila embolicada.
- el contenidor (quadre) pot augmentar / disminuir en funció de l'espai disponible i es pot determinar si es tracta d'una columna o d'una fila.
- independència de la font: és a dir, que l'ordre dels articles no importa, només han d'estar dins de la caixa.

2 Pregunta 2. Sobre disseny responsive.

2.1 Expliqui en breus paraules que és el disseny responsive (responsive design).

En els primers dies del disseny web, les pàgines es dissenyaven per omplir una mida de pantalla en particular. Si l'usuari tenia una pantalla més gran o més petita que la del dissenyador, els resultats esperats anaven des barres de desplaçament no desitjades fins longituds de línia excessivament llargues i un mal ús de l'espai. A mesura que van estar disponibles mides de pantalla més diverses, va aparèixer el concepte de disseny web responsiu (RWD, responsive web design), Un conjunt de pràctiques que permet a les pàgines web alterar el seu disseny i aparença per adaptar-se a diferents amplituds de pantalla, resolucions, etc. És una idea que va canviar la forma en què vam dissenyar per a una web multidispositiu.

2.2 Descriu detalladament l'estructura emprada en CSS per a aplicar disseny responsive.

El disseny responsiu només va poder sorgir gràcies a la consulta als mitjans.

Les consultes als mitjans ens permeten executar una sèrie de proves (per exemple, si la pantalla d'usuari és més gran que un ample o una resolució determinats) i aplicar CSS selectivament per dissenyar la pàgina de manera que resulti adequada a les necessitats de l'usuari.

Per exemple, la consulta als mitjans següent explora si la pàgina web que es mostra ho fa com un mitjà de pantalla (per tant, no és un document imprès) i si la finestra té al menys 800 píxels d'ample. El CSS per al selector `.container` només s'aplicarà si les dues condicions són certes.

```
@media screen and (min-width: 800px) {  
  .container {  
    margin: 1em 2em;  
  }  
}
```

Pots afegir múltiples consultes als mitjans dins d'un full d'estil, i ajustar tot el teu disseny o només parts d'ell perquè s'adapti millor a les diferents grandàries de pantalla. Els punts en els quals s'introdueix una consulta als mitjans i es canvia el disseny es coneixen com punts d'interrupció.

Un enfocament comú quan es fan servir les consultes als mitjans és crear un disseny senzill d'una sola columna per a dispositius de pantalla estreta (per exemple, telèfons mòbils), després implementar un disseny en columnes per a pantalles més grans quan se sap que hi ha prou ample de pantalla per manejar-lo. Això es descriu sovint com disseny primer mòbil.

2.3 Explica què podem fer per a facilitar que la pàgina sigui responsive abans de fer el pas d'afegir mitjans queries.

El terme disseny responsiu va ser encunyat per Ethan Marcotte en 2010, i descrivia l'ús combinat de tres tècniques.

- La primera era la idea de les xarxes fluides.

- La segona tècnica era la idea de les imatges fluïdes. Usant una tècnica molt simple d'establir la propietat de `max-width` al 100%.
- El tercer component clau era la consulta als media.

Quadrícules flexibles

Els llocs responsive no sols canvien el seu disseny entre punts d'interrupció, sinó que es construeixen sobre quadrícules flexibles. Una quadrícula flexible significa que no has de centrar-te en totes les grandàries de dispositiu possibles i construir per a ells un disseny en píxels perfecte.

2.3.1 Multicol

El més antic d'aquests mètodes de disseny de pàgines web és multicol. Quan especifiques un atribut `column-count`, això indica en quantes columnes desitges dividir el teu contingut. El navegador llavors calcula la grandària d'aquestes columnes, que canviarà d'acord amb la grandària de la pantalla.

Si en lloc d'això estableixes l'atribut `column-width`, especifiques un ample mínim. El navegador crea tantes columnes d'aquest ample com càpiguen còmodament en el contenidor, i reparteix l'espai entre totes les columnes. Per tant, el nombre de columnes canvia segons la quantitat d'espai que hi ha.

2.3.2 Flexbox

En el mètode Flexbox, els elements flexibles s'encongeixen i distribueixen l'espai entre els elements segons l'espai que hi ha en el seu contenidor, segons el seu comportament inicial. En canviar els valors de `flex-grow` i `flex-shrink`, pots indicar com desitges que es comportin els elements quan al seu al voltant hi ha més o menys espai.

2.3.3 Cuadrículas CSS

En el disseny de quadrícules amb CSS, la unitat `fr` permet la distribució de l'espai disponible en les traces de la quadrícula.

2.3.4 Imatges responsive

L'enfocament més simple per a les imatges responsive és el que es descriu en els primers articles de Marcotte sobre disseny responsive. Bàsicament, prendre una imatge que tingui la grandària més gran que puguis necessitar, i reduir-la. Aquest continua sent un enfocament utilitzat avui dia.

2.3.5 Tipografia responsive

Aquest concepte descriu essencialment el fet de canviar la grandària de lletra segons l'espai de pantalla que reflecteixen les consultes a mitjana.

Un enfocament interessant és utilitzar les unitats de finestra gràfica vw per a habilitar la tipografia responsive. 1vw és igual a l'u per cent de l'ample de la finestra gràfica, cosa que significa que si configures la grandària del tipus de lletra amb vw, sempre estarà en relació amb la grandària de la finestra gràfica.

2.3.6 La metaetiqueta viewport

Aquesta metaetiqueta informa els navegadors dels dispositius mòbils que han d'establir l'ample de la finestra gràfica a l'ample del dispositiu i escalar el document al 100% d'aquesta grandària, de manera que el document es mostrarà a la grandària optimitzada per a aquests dispositius mòbils.

3 GitHub Pages

He publicat la pàgina web al servidor de GitHub Pages, l'adreça web és la següent.

<https://gerardfauchs.github.io/web/>.