UNIVERSITAT ROVIRA i VIRGILI

Repte CSS

Maquetació portfoli personal.

Un cop es té tota l'estructura HTML ja es pot crear el fitxer style.css i començar a maquetar. És a dir, començar a donar-li forma a la web. Per a fer-ho utilitzarem CSS. Primer seria recomanable començar a donar estructura general a la web i veure si es necessitaria algun element contenidor extra (div, section, main, etc) per si es vol aplicar Flexbox o Grid.

Part 1

Crear barra de menú transversal, un document CSS i donar estils a la pàgina de Currículum Vitae (cv.html)

Introducció

En aquesta part es començarà a treballar amb CSS. Cada pàgina HTML ha de tenir enllaçat un document CSS (típicament se li diu style.css) on s'hi escriuran totes les propietats i valors CSS per tal de donar-li els estils visuals necessaris per a que el resultat final s'ajusti al disseny proposat.

El CSS serveix per donar estils als elements HTML. És molt comú que en una web hi hagi elements repetits i és aquí on entra en joc CSS. Si a tots els elements que tenen el mateix estil se'ls hi assigna la mateixa classe CSS, només caldrà definir els estils una vegada al document style.css, i l'estil es replicarà per tots els elements que comparteixin la mateixa classe. Per exemple, si volem que tots els títols tinguin la mateixa mida i color, podem crear una classe titol i assignar-la a tots els títols.

style.css

```
.titol {
  font-size: 2rem;
  color: red;
}
```

index.html

<h2 class=”titol”>Els nostre serveis</h2>
…
<h2 class=”titol”>El nostre equip</h2>

Tasca

Crear un fitxer style.css i enllaçarlo a la pàgina cv.html

A document cv.html, identificar tots els elements que comparteixen el mateix estil, i assignar-los el mateix nom de classe.

Al document style.css, escriure les propietats i valors corresponents per tal d'ajustar el resultat final al disseny proposat (maquetació)

Crear la capçalera de la web amb la barra de menú i els enllaços cap a les altres pàgines.

A tenir en compte

En una web amb moltes pàgines, el document CSS pot arribar a ser molt gran, per tant és important tenir el codi ben estructurat i amb comentaris, per saber a quina part correspon cada grup de propietats.

D'igual manera és important utilitzar noms de classe descriptives. Per exemple, si tenim un formulari, és millor utilitzar noms de classe com “contacte” o “formulari” que no pas noms genèrics “element” o “part”.

En aquest punt no cal saber les mesures ni colors exactes del disseny, només cal arribar a tenir un resultat aproximat al disseny proposat. Més endavant es treballarà en base a Figma per tal d'ajustar el disseny exactament a la proposta.

La capçalera de la web és un element que es repetirà a totes les pàgines: index, cv i contacte (opcionalment). En aquest tipus d'elements, per evitar repetir feina, és

recomanable crear-lo i donar-li estils primer en una pàgina i, quan l'estructura HTML ja estigui acabada (etiquetes i classes), llavors replicar-lo per la resta de pàgines.

D'altra manera si, per exemple, canviem una classe del menú a cv, també l'haurem de canviar a index i contacte. Per evitar aquesta feina duplicada es recomana acabar-lo primer en una pàgina i després replicar-lo.

Treballant amb frameworks es pot evitar aquesta tasca de duplicar elements, ja que estan pensats per optimitzar el flux de treball, però en aquest curs més introductori encara hem fer aquesta feina per cada pàgina.

Recursos recomanats

Consultar documentació adjunta de CSS, exemples de codi del repositori Github del curs: https://github.com/Aleix-Marti/curs-frontend/tree/main/CSS

W3schools: Introducció i referències de CSS
https://www.w3schools.com/css/default.asp
https://www.w3schools.com/cssref/default.asp

MDN: Introducció i referències de CSS
https://developer.mozilla.org/es/docs/Web/CSS
https://developer.mozilla.org/es/docs/Learn/CSS

Part 2

Maquetació de la pàgina principal

Introducció

En aquesta part es començarà a treballar amb Figma. Figma és una eina de disseny i prototipat, que permet als programadors consultar en detall totes les parts del disseny i inclús veure'n algunes de les propietats de CSS. D'aquesta manera es poden ajustar mides, espaiats, tipografies, colors… amb l'objectiu d'aconseguir una maquetació fidel al disseny proposat.

Tot i ser una eina molt potent a nivell de disseny, només s'utilitzarà per consultar el disseny proposat i mirar d'ajustar la maquetació per tal d'obtenir un resultat final el més similar possible al disseny proposat.

Cal tenir en compte que el disseny proposat és només orientatiu. Si algú prefereix crear el seu propi disseny, o aplicar algunes modificacions sobre la proposta, és totalment lliure de fer-ho.

Tasca

Maquetar la pagina d'inici (index.html) per tal d'aconseguir un resultat el més similar possible al disseny proposat.

Enllaçar el fitxer style.css a index.html

Escriure el codi CSS necessari per obtenir un resultat el més similar possible al disseny proposat.

A tenir en compte

És força comú que al plantejament inicial del fitxer index.html se li hagin d'aplicar modificacions. No només afegir classes CSS, sinó inclús canviar l'estructura d'alguns elements.

Per exemple, per utlitzar Flexbox, cal tenir un element que actuï com a contenidor. D'una proposta inicial on s'ha identificat una secció destacada amb una imatge i un text es pot tenir quelcom així:

```
<img src=”imatge.jgp”>
<p>lorem ipsum dolor sit amet...</p>
```

Però si volem tractar aquest conjunt d'elements amb propietats Flexbox, caldrà posar-los a dins d'un altre element que actuï de contenidor. Per exemple:

```
<div class=”destacat”>
  <img src=”imatge.jgp”>
  <p>lorem ipsum dolor sit amet...</p>
</div>
```

D'aquesta manera ja podrem “atacar” aquesta secció desde la nostra classe CSS amb les propietats de Flexbox:

```
.destacat {
  display: flex;
  flex-direction: row;
  justify-content: space-evenly;
}
```

Figma permet veure els atributs o propietats CSS dels elements del disseny. És molt útil sobretot per ajustar tamanys de lletra, espaiats, colors i descarregar recursos (icones i imatges).

Recursos recomanats

Disseny proposat
https://www.figma.com/file/093xOOr3uJr9ClxYrepqMD/Untitled

Vídeos ús de variables amb CSS (CSS avançat)
https://wordpress.tv/2018/11/11/aleix-marti-variables-nativas-css-front-end-con-super poderes

Joc per aprendre Flexbox
https://flexboxfroggy.com/#ca

Joc per aprendre Flexbox
http://www.flexboxdefense.com

W3schools: Flexbox
https://www.w3schools.com/css/css3_flexbox.asp

Part 3 (opcional)

Si s'ha implementar la pàgina de contacte, maquetar el formulari de contacte. Es pot agafar de referència qualsevol formulari de contacte que es vulgui agafar com a inspiració.

Part 4 (opcional)

Maquetació responsive de la pàgina web

Introducció

Grid CSS no és una tècnica específica per aplicar responsive, però ben combinat amb media queries pot ser útil ja que, igual que Flexbox, treballa de forma molt fluïda i s'adapta força bé al canvis de mida de la pantalla, gairebé automàticament.

De totes maneres, hi ha alguns elements que és complicat que s'adaptin automàticament, tot i utilitzar Flexbox o Grid. En aquests casos serà necessari fer ús de les media queries.

Hi ha dos formes de plantejar la maquetació responsive: de petit a gran (mobile first) o de gran a petit. És cert que quan es té certa experiència és més fàcil adoptar la primera forma, però d'entrada sembla més complicat. Es pot utilitzar la forma que sigui més còmoda per cadascú, només cal tenir en compte a escriure bé les media queries i entendre els seus objectius.

Per exemple, agafant la mida de tall de 768px, es vol que el color d'un text sigui blau per dispositius mòbils i tauleta vertical, i de color vermell per tota la resta de pantalles més grans. Es pot plantejar de 2 maneres. El resultat serà el mateix, però cal posar la propietat inicial fora de la media query, i la propietat amb el canvi a dins.

Mobile first: min-width (o a partir de la mida indicada)

```
.text {
  color: blue;
  font-size: 1rem;
}

@media only screen and (min-width: 768px) {
  .text {
    color: red;
    font-size: 1.2rem;
  }
}
```

Desktop first: max-width (o fins a la mida indicada)

```
.text {
  color: red;
  font-size: 1.2rem;
}

@media only screen and (max-width: 768px) {
  .text {
    color: blue;
    font-size: 1rem;
  }
}
```

Tasca

Escriure el codi CSS necessari i modificar HTML si cal, per aconseguir que la pàgina web es vegi bé en diferents mides de pantalla.

Maquetar la capçalera de la web i el peu (header i footer) i replicar-ho a totes les pàgines.

A tenir en compte

Si es treballa amb media queries, aplicar els talls per les mides següents:

600px (mòbil)
768px (tauleta vertical)
992px (tauleta horitzontal)
1200px (escriptori)

Recursos recomanats

W3schools: RWD
https://www.w3schools.com/css/css_rwd_intro.asp

W3schools: apartat Typical Device Breakpoints
https://www.w3schools.com/css/css_rwd_mediaqueries.asp

Documentació de Google Chrome Developer Tools.

https://developer.chrome.com/docs/devtools/device-mode/

La web CSS Grid Garden per familiaritzar-se amb Grid jugant.
https://cssgridgarden.com/#ca

Article CSS Grid: A guide to getting started.
https://blog.logrocket.com/css-grid-getting-started/

Web Learn CSS Grid.
https://learncssgrid.com/

Col·lecció de Codepen amb exemples de Grid
https://codepen.io/collection/XRgWPe/

Exemple de Codepen amb una tarjeta responsive amb media queries i Grid CSS
https://codepen.io/Aleix/pen/BazMJYp

Vídeo de trucs de CSS avançat.
https://www.youtube.com/watch?v=D1-hKNZgFBA&t=218s&ab_channel=WordPressTarragona