

PEC 1 - Herramientas HTML y CSS

Url Github: https://github.com/FerranRoigUOC/HTML_CSS_II_PEC1

Url Netlify: <https://roaring-monstera-f5ccb9.netlify.app/>

Per realitzar aquesta primera pràctica de la assignatura, el primer que he fet, tal com hem vist en la teoria, es inicialitzar el projecte amb npm: *npm init -yes* . El que ens fa aquesta comanda es crear un fitxer “package.json”, que serveix per a quan fem un build o un start del projecte, s'utilitzi la configuració d'aquest fitxer.

```
{
  "name": "my-project",
  "source": "src/index.html",
  "browserslist": "> 0.5%, last 2 versions, not dead",
  > Debug
  "scripts": {
    "start": "npm-run-all clean parcel:dev",
    "build": "npm-run-all clean parcel:build",
    "parcel:dev": "parcel",
    "parcel:build": "parcel build",
    "clean": "rimraf dist .parcel-cache"
  },
  "devDependencies": {
    "npm-run-all": "^4.1.5",
    "parcel": "latest",
    "rimraf": "^3.0.2"
  },
  "dependencies": {
    "lazysizes": "^5.3.2"
  }
}
```

Aquesta és la configuració que li he donat; el nom del projecte, el root del projecte, amb el “browserslist” indiquem que volem que el nostre projecte també soporti versions de navegadors antics, els scripts que s'executaran amb el *npm run build* i *npm run start*, on indiquem que faci una neteja de les compilacions anteriors i s'iniciï el parcel (el qual ens permet fer canvis en el codi i es vegin reflectats automàticament, al localhost:1234 en aquest cas). També definim les dependències que ens ajuden al desenvolupament i les dependències funcionals, que en el nostre cas hem utilitzat la llibreria lazysizes, que després explicarem per a que serveix.

A nivell de programació, s'ha realitzat una pàgina web senzilla però força completa. S'ha desenvolupat una pàgina web responsive, estructurant-la amb header, body i footer. Utilitzant un menú en el header, amb la propietat de position: sticky en versió

mòbil, el qual li dona un toc més amigable per a l'usuari quan fa scroll cap abaix, i utilitzant el flex-box i el gridview per estructurar els elements del body. Per tal de començar a relacionar els fitxers HTML amb els fitxers de Javascript, s'ha desenvolupat una funcionalitat molt senzilla sobre obrir un fitxer HTML quan es clica un element en concret en el fitxer de JS.

Com a extra, s'ha instal·lat una llibreria que millora una mica el rendiment del projecte. Aquesta llibreria és diu lazysizes

<https://www.npmjs.com/package/lazysizes>, i serveix per inicialitzar tant imatges com vídeos abans de que es carregi la web per complet.

Per tal d'utilitzar aquesta llibreria, l'he tingut d'instal·lar: *npm install lazysizes --save*, i importar-la als fitxers HTML on volia fer ús d'aquesta:

```
<script src="lazysizes.min.js" async=""></script>
```

I finalment, he tingut de canviar una mica el codi en els elements d'imatges i vídeos, especificant que han d'utilitzar la classe *lazyload*, tal com ens explica les instruccions d'aquesta llibreria:

```
<!-- non-responsive: -->  

```

```
<!-- iframe example -->  
<iframe frameborder="0"  
    class="lazyload"  
    allowfullscreen=""  
    data-src="//www.youtube.com/embed/ZfV-aYdU4uE">  
</iframe>
```

Finalment, una vegada finalitzat el projecte, s'ha penjat el projecte al repositori Github i Netlify, en les urls que s'han indicat al principi de la documentació.