

## PEC 1 : Eines HTML i CSS II

### - Instal·lació

Per començar amb la primera PEC de la assignatura, m'havia d'asegurar que tinges tot el necessari instal·lat i amb unes versions actualitzades. Tenint en compte que el VS ja el tenia instal·lat, el primer que havia de fer era actualitzar el node js desde la seva pagina oficial -> <https://nodejs.org/en/download/> i per comprovar la seva versió, utilitzem `node -v` i veiem que té la versió 20.11.1, la última versió.

```
PS C:\Users\ferri\OneDrive\Escritori\24_html_css_II\1\teoria\uoc-boilerplate-master> node -v
v20.11.1
```

Una vegada tenim el node js, solament faltava instal·lar el *npm* mitjançant `npm init --yes` i ja ho tenim tot preparat per treballar.

El següent pas ja es descarregar el repositori amb el qual em de partir com a base i començar a treballar : <https://github.com/uoc-advanced-html-css/uoc-boilerplate>

Una vegada tenim el repositori descarгат, utilitzem la comanda *npm install*, aquesta comanda utilitza el gestor de paquets npm (Node Package Manager) per descarregar i instal·lar els mòduls especificats en el fitxer `package.json` del projecte:

```
{
  "homepage": "https://github.com/uoc-advanced-html-css/uoc-boilerplate#readme",
  "devDependencies": {
    "@parcel/transformer-sass": "^2.11.0",
    "autoprefixer": "^10.4.15",
    "npm-run-all": "^4.1.5",
    "parcel": "^2.11.0",
    "postcss-preset-env": "^9.1.1",
    "posthtml-include": "^1.7.4",
    "rimraf": "^5.0.1",
    "sharp": "^0.31.1",
    "stylelint": "^16.3.1",
    "stylelint-config-recommended-scss": "^14.0.0",
    "stylelint-scss": "^6.2.1"
  }
}
```

Com podem veure a la imatge, s'ha instal·lat la llibreria Parcel, aquesta llibreria és una eina de construcció de paquets per a desenvolupament web que simplifica la compilació d'arxius JavaScript, CSS, imatges i altres recursos (s'autodenomina *zero configuration web application bundler*),. Automatitza la creació de paquets optimitzats per al web, incloent funcionalitats com minificació i gestió d'imports, permetent al desenvolupador centrar-se en la seva feina sense preocupar-se per la configuració. Podem veure en el `package.json` que Parcel s'utilitza en tots els scripts:

```
4   "private": true,
5   "description": "Boilerplate for Advanced HTML/CSS Tools UOC students",
6   "scripts": {
7     "parcel:serve": "parcel src/index.html -p 8123 --target web --open",
8     "parcel:build": "parcel build src/index.html --target web --no-source-maps --no-cache",
9     "clean": "rimraf dist .cache .cache-loader .parcel-cache",
10    "dev": "npm-run-all clean parcel:serve",
11    "build": "npm-run-all clean stylelint parcel:build",
12    "test": "echo 'Everything is working as expected' \nStart working on your project by running 'npm run dev'",
13  },
14  "web": "dist/index.html",
15  "targets": {
16    "web": {
```

El scrip `dev`, la part de `parcel:serve`, ens permet obrir un port obert amb el projecte amb el qual podem accedir desde un browser, i si fem canvis en el html o css, ho veurem reflectit automaticament, sense necessitat de tornar a fer un build del projecte.

```
S C:\Users\ferri\OneDrive\Escritorio\uoc\2024_html_css_II\1\teoria\uoc-boilerplate-master\uoc-boilerplate-master> npm run dev

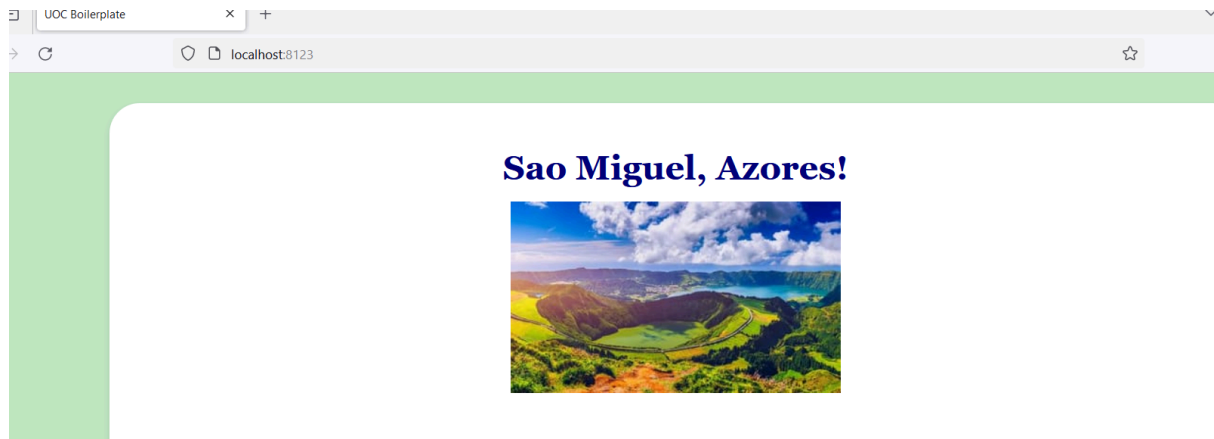
uoc-boilerplate@3.8.0 dev
npm-run-all clean parcel:serve

uoc-boilerplate@3.8.0 clean
rimraf dist .cache .cache-loader .parcel-cache

uoc-boilerplate@3.8.0 parcel:serve
parcel src/index.html -p 8123 --target web --open

server running at http://localhost:8123
Building azores.jpg...
rowserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Building azores.jpg...
rowserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Building all.css...
rowserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Built in 2.09s
```

I com podem veure, ens obri el port 8123:



- Disseny : Desktop first

Per fer aquesta primera PEC he fet un disseny molt senzill però amb un bon responsive, el cual es Desktop first.

El contingut de la pàgina consisteix amb una imatge i quatre paràgrafs amb informació de la ciutat. Aquests 4 paragraphs es mostren com un Grid quan es per una pantalla d'ordinador i es mostren com a llistat (un baix l'altre) quan es mostra per a pantalles més petites (per a mòbils).

Ordenador:



Mòbil:



I el codi CSS utilitzat per mostrar-ho en grid, on la classe `.azores-container-grid` és un div que conté els 4 paràgrafs, i com podem veure posem el parametre `display a grid` i amb 2 columnes:

```
@media (min-width: $mqMinWidth) {  
  .azores-container-grid {  
    // Positioning  
    padding: 15px;  
  
    // Box model  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    gap: 20px;  
  }  
  
  .azores-logo {  
    max-width: none;  
  }  
  
  .azores-title {  
    font-size: 36px;  
  }  
}
```

## - Dependència externa

La dependència externa que he implementat és la mateixa que la del Mòdul 1, FontAwesome, ja que en el Boiler ja estava importada com a dependència en el package.json, sols feia falta importar-la en el main.scss:

```
@import "npm:@fortawesome/fontawesome-free/css/all.css";
```

Com hem pogut veure en el disseny, hem utilitzat quatre icones, una per a cada paràgraf:



I per mostrar un exemple de codi HTML d'aquesta llibreria, aquí mostro per a les dues primeres icones:

```
<i class="fas fa-house">  
<i class="fas fa-person-hiking">
```

## - Metodologies i guies d'estil (BEM & Mark Otto)

La guia d'estil utilitzada ha sigut Mark Otto, ja que és una guia més orientada a la programació amb Bootstrap, ja que el propi Mark és el creador d'aquest, de tal manera que em vaig familiaritzant amb aquest tipus de programació.

Començo explicant els passos que he seguit per aplicar aquesta guia d'estil en HTML (en el meu cas d'ús):

- tags en minúscula
- tab + dos espais entre el tag pare i fill
- utilitzar `<!doctype html>`
- utilitzar `<html lang="es">`
- utilitzar `<meta charset="utf-8">`

- mantenir la simple, com p.e, no utilitzar funcions com onClick dins dels tags

```

1  <!DOCTYPE html>
2
3  <html lang="es">
4    <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width">
7      <link rel="stylesheet" type="text/css" href="/assets/styles/main.sc
8      <title>UOC Boilerplate</title>
9    </head>
10   <body>
11     <div class="azores-container">
12       <h1 class="azores-title">Sao Miguel, Azores!</h1>
13       <picture>
14         <source srcset="/assets/images/azores.jpg?width=350" type="imag
15         
17       <div class="azores-container-grid">
18         <p><i class="fas fa-house"></i> Sao Miguel, la més gran de les i

```

I per seguir la guia d'estil amb CSS (amb el meu cas d'ús):

- incloure un espai després de: per a cada declaració
- incloure un espai abans de la clau d'obertura dels blocs de declaració per a una millor llegibilitat
- col·locar claus de tancament dels blocs de declaració en una línia nova.
- no posar com a prefix els valors de propietat ni els paràmetres de color amb un zero inicial (p. ex., .9 en lloc de 0.9 )
- utilitzar espai entre els valors per a les propietats del color
- tots els valors hexadecimals en minúscules, per exemple, #fff. Les lletres minúscules són molt més fàcils de discernir en escanejar un document, ja que solen tenir formes més úniques.
- utilitzar valors hexadecimals abreviats quan estiguin disponibles, per exemple, #fff en comptes de #ffffff.
- evitar especificar unitats per a valors zero, p. ex., marge: 0; en lloc de marge: 0px;.
- seguir el ordre -> Positioning, Box model, Typographic, Visual i Misc

```

5 body {
6   // Positioning
7   margin: 0;
8   padding: 2rem;
9
10  // Box model
11  display: flex;
12  flex-direction: column;
13  height: 100vh;
14
15  // Typography
16  font-family: $font-family-sans-serif;
17  font-size: 16px;
18  line-height: 24px;
19
20  // Visual
21  background-color: $uoc-bg;
22  color: $uoc-corporate;
23 }
24
25 .azores-container {
26   // Positioning
27   margin: auto;
28   padding: 2rem;
29
30   // Box model
31   max-width: 1200px;
32
33   // Typography
34   text-align: center;
35
36   // Visual
37   border-radius: 2rem;
38   background-color: #fff;
39   box-shadow: 0 4px 6px -1px rgba(0 0 0 / .1), 0 2px 4px -1px rgba(0 0 0 / .06);
40 }

```

Continuant amb la metodologia, la utilitzada ha sigut la BEM. Qualsevol de les metodologies em semblava correcta per a utilitzar, pero he escollit aquesta porque em pareix una bona forma de reutilitzar codi i amb un codi molt clar de llegir i d'entendre:

```

60
61 .text {
62   // Typography
63   font-family: $font-family-serif;
64
65   // Visual
66   color: $uoc-corporate;
67 }
68
69 .text-title {
70   font-size: 24px;
71   font-weight: 700;
72 }
73

```

```

<div class="azores-container">
  <h1 class="text text-title">Sao Miguel, Azores!</h1>
  <picture>
    <source srcset="./assets/images/azores.jpg?width=350" media="(max-width: 350px)" />
    
  </picture>
  <div class="azores-container-grid">
    <p class="text"><i class="fas fa-house"></i> Sao Miguel is a natural paradise with impressive landscapes. This volcanic island offers a unique combination of natural wonders, including hot springs, dense forests, and beautiful beaches. It's the perfect place to explore its natural beauty.
    </p>
    <p class="text"><i class="fas fa-person-hiking"></i> Hiking is a popular activity in Sao Miguel, with many trails offering stunning views of the island's natural beauty.
  </div>
</div>

```

## - Stylelint

Per tal de comprovar que hem seguit la metodologia BEM, instal·lem la llibreria Stylelint amb la següent comanda:

```
npm install --save-dev stylelint-scss stylelint-config-recommended-scss
```

Una vegada tenim la llibreria instal·lada, apliquem la norma que ha de seguir el nostre codi, en aquest cas la norma per seguir la metodologia BEM, en un nou fitxer `.stylelintrc`:



Amb la comanda `npm run stylelint`, podem comprovar si hi ha alguna part del codi que no segueix la metodologia escollida, en el nostre cas no ens surt cap error, per tant, hem seguit la metodologia BEM correctament.

```
PS C:\Users\ferri\OneDrive\Escritorio\uoc\2024_html_css_II\1\teoria\uoc-boilerplate-master> npm run stylelint
> uoc-boilerplate@3.8.0 stylelint
> stylelint src/**/*.scss

PS C:\Users\ferri\OneDrive\Escritorio\uoc\2024_html_css_II\1\teoria\uoc-boilerplate-master>
```

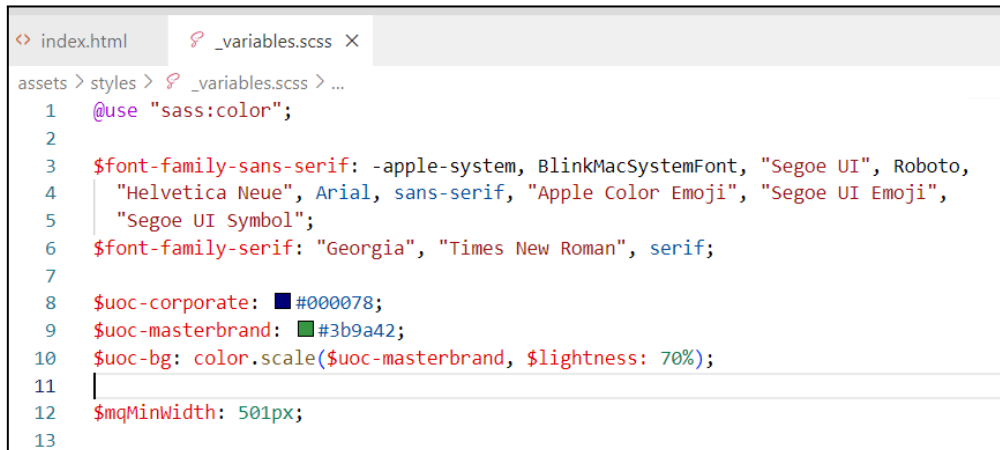


- Sass: variables, imbricació (nesting), funcions, parcials i importació

Per últim se'ns demana utilitzar els conceptes Sass:

- Variables:

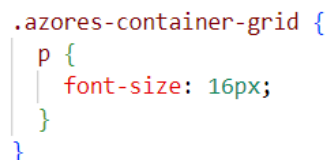
Les variables és una manera d'emmagatzemar informació que volem reutilitzar al nostre full CSS. Pots emmagatzemar coses com ara colors, piles de fonts o qualsevol valor CSS que creguis que voldràs reutilitzar.



```
<> index.html  _variables.scss X
assets > styles > _variables.scss > ...
1  @use "sass:color";
2
3  $font-family-sans-serif: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
4  "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji",
5  "Segoe UI Symbol";
6  $font-family-serif: "Georgia", "Times New Roman", serif;
7
8  $uoc-corporate: #000078;
9  $uoc-masterbrand: #3b9a42;
10 $uoc-bg: color.scale($uoc-masterbrand, $lightness: 70%);
11
12 $mqMinWidth: 501px;
13
```

- Nesting:

Sass us permetrà niar els selectors CSS d'una manera que segueixi la mateixa jerarquia visual del vostre HTML, de tal manera que serà més fàcil de llegir i entendre.



```
.azores-container-grid {
  p {
    font-size: 16px;
  }
}
```

## - Funcions:

Les funcions ens permeten definir operacions complexes en valors de SassScript que podem reutilitzar al llarg del full d'estil CSS. Faciliten l'abstracció de fórmules i comportaments comuns d'una manera llegible, en el nostre cas, hem utilitzat la funció `rgb()`.

```
// Typography
text-align: center;

// Visual
border-radius: 2rem;
background-color: #fff;
box-shadow: 0 4px 6px -1px rgba(0 0 0 / .1), 0 2px 4px -1px rgba(0 0 0 / .06);
}
```

## -Parcials:

Els fitxers Sass parcials serveixen per contenir petits fragments de CSS que pot incloure en altres fitxers Sass. Aquesta és una excel·lent manera de modularitzar el CSS i ajudar a fer que el manteniment sigui més fàcil.

```
_variables.scss X
assets > styles > _variables.scss > ...
1  @use "sass:color";
2
3  $font-family-sans-serif: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
4    "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji",
5    "Segoe UI Symbol";
6  $font-family-serif: "Georgia", "Times New Roman", serif;
7
8  $uoc-corporate: #000078;
9  $uoc-masterbrand: #3b9a42;
10 $uoc-bg: color.scale($uoc-masterbrand, $lightness: 70%);
11
12 $mqMinWidth: 501px;
13
```

- Importació:

Utilitzem el fitxer main.scss per utilitzar tots els imports necessaris del projecte, veiem que tenim les variables que després reutilitzem, el full CSS de l'índex i la llibreria FontAwesome.

```
main.scss x
assets > styles > main.scss
1  /**
2   * Use this file to import partials.
3   * Do not include SCSS code in this file.
4   */
5
6  /**
7   * Import variables
8   */
9
10 @import "variables";
11
12 /**
13  * Import npm dependencies
14  * see: https://v2.parceljs.org/features/module-resolution/
15  * see commented examples below
16  */
17
18 // @import "some-node-module";
19 // @import "@some-company/some-node-module";
20
21 /**
22  * Import layouts
23  * It's a best practice to use a different partial for each page
24  */
25
26 @import "layouts/home";
27
28 /**
29  * Do NOT include anything else in this file!
30  */
31
32 @import "npm:@fortawesome/fontawesome-free/css/all.css";
33
```

- Netlify & GitHub

Netlify: <https://dashing-marigold-047807.netlify.app/>

GitHub: [https://github.com/FerranRoigUOC/2024\\_HTML\\_CSS\\_II](https://github.com/FerranRoigUOC/2024_HTML_CSS_II)