

OHMYFOOD

PRÉSENTATION

HTML

- Création du code HTML
- Une page d'accueil
- Quatres pages de menu

SASS

- Création du code CSS via SASS
- Mise en place du code via :
 - Nesting
 - BEM
 - Mixins
 - Variables
 - Loop
 - Media queries
 - Keyframes

BEM

Utilisation de la méthodologie BEM via un nommage des classes qui élimine les risques de conflits CSS.

- **Blocks – Elements – Modifiers**

<> index.html > html > body > div.bloc_page > div.loader

```
31      <!-- HEADER -->
32
33      <header>
34
35          <div class="box_title">
36              <h1>ohmyfood</h1>
37          </div>
38
39          <div class="search_box">
40              <div class="search_box__icon">
41                  <i class="fas fa-map-marker-alt"></i>
42              </div>
43              <input class="search_box__place_search" type="text" placeholder="Paris, Bellevue" />
44          </div>
45
46          <div class="title_label">
47              <span class="title_label__1">Réservez le menu qui vous convient</span>
48              <span class="title_label__2">Découvrez des restaurants d'exception, sélectionnés par nos soins</span>
49
50              <a href="#restaurants" class="title_label__button_explorer">
51                  <span>Explorer nos restaurants</span>
52              </a>
53          </div>
54      </header>
```











VARIABLES

Création de variables dans le but de pouvoir par la suite modifier les éléments qui sont identiques comme par exemple le code couleur prévu dans la maquette.

A voir directement dans le code

`_variables.scss`

sass > utils >  _variables.scss > ...

```
1  $color-primary:  #9356dc;
2  $color-secondary:  #ff79da;
3  $color-third:  #99e2d0;
4  $color-linear: linear-gradient(to top,  #9356dc,  #b35cdc,  #cf64dc,  #e86edb,  #ff79da);
5
6  $font-family1: "Shrikhand";
7  $font-family2: "Roboto";
8  $font-family3: "Roboto-Black";
9  $font-family4: "Roboto-Light";
10 $font-family5: "Roboto-Bold";
11
12 $box-shadow: 0px 2px 10px 1px  #c4c4c4;
13 $box-shadow-plus: 3px 7px 6px 0px  #c4c4c4;
14
15 $border-radius1: 15px 15px 15px 15px;
16
```

LOOP

Création de loop @each et @for dans le but de réduire le code et permettre une création du css plus rapide, en évitant les répétitions.

A voir directement dans le code

`_loop.scss`

sass > utils >  _loop.scss > ...



```
1
2
3  @for $i from 1 through 4 {
4    |   #image_background_#{ $i } {
5    |   |   background-image: url('../..images/restaurants/' + $i + '.jpg');
6    |   }
7  }
8
9  $menu-items: 1, 2, 3 ,4, 5, 6, 7, 8, 9, 10;
10
11  @each $item in $menu-items {
12    |   .small_box-#{ $item } {
13    |   |   opacity: 0;
14    |   |   animation: fadein 1s;
15    |   |   animation-delay: #{$item*400}ms;
16    |   |   animation-fill-mode: forwards;
17    |   }
18  }
```


MEDIA QUERIES

Création d'un fichier scss media pour le responsive directement dans le code pour permettre une intégration plus simple.

A voir directement dans le code

`_media.scss`

```
sass > utils >  _media.scss >  $phone
```

```
1  $phone: 375px;
```

```
2  $tablet: 768px;
```

```
3  $desktop: 1024px;
```

```
4
```

```
5  $media-down: "only screen and (min-width: #{ $phone })";
```

```
6  $media-medium: "only screen and (min-width: #{ $tablet })";
```


```
7  $media-up: "only screen and (min-width: #{ $desktop })";
```

KEYFRAMES

Création d'un fichier Keyframes pour inclure des animations directement dans le code afin de permettre une intégration plus simple.

A voir directement dans le code

`_keyframes.scss`

```
sass > utils >  _keyframes.scss >  @keyframes rotate
```

```
1  @keyframes fadein {  
2    from {  
3      opacity:0;  
4    }  
5    to {  
6      opacity:1;  
7    }  
8  }  
9  
10 @keyframes rotate {  
11   from{  
12     -webkit-transform: rotate(0deg);  
13   }  
14   to{  
15     -webkit-transform: rotate(360deg);  
16   }  
17 }  
18  
19 @keyframes loading {  
20   0% {  
21     opacity: 1;  
22   }  
23   50% {  
24     opacity: 0.5;  
25   }  
26   100% {  
27     opacity: 0;  
28     display: none;  
29   }  
30 }
```

MIXINS

Création de mixins dans le but de pouvoir par la suite modifier les éléments (button, form...) qui sont identiques.

A voir directement dans le code

`_mixins.scss`

Vérification W3C HTML/CSS

Code vérifier et valider sans erreurs.

- <https://validator.w3.org/>

index + 4 pages menu

- <https://jigsaw.w3.org/css-validator/>

style + menu