



Mise en place de la partie centrale

Modification de `index.html`

Nous allons mettre en place la structure de la partie centrale :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Café Florette | Accueil</title>
    <link rel="stylesheet" href="css/styles.css" />
    <link
      href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,700&..."
      rel="stylesheet"
    />
  </head>
  <body>
    <header class="header">
      <nav>
        <h1><a href="./index.html">CAFE FLORETTE</a></h1>
        <ul>
          <li><a class="active" href="./index.html">Accueil</a></li>
          <li><a href="./where.html">Nous trouver</a></li>
          <li><a href="./contact.html">Contact</a></li>
        </ul>
      </nav>
    </header>
    <section class="vp">
      <div class="vp-content">
        <h1>Toute l'expertise française dans un café</h1>
        <div class="separator"></div>
        <p>
          Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperna
          nemo cupiditate earum eum porro unde sint ducimus vero voluptas,
          perferendis quasi quia accusamus omnis rerum. Neque sapiente quid
          distinctio dolorem?
        </p>
        <a href="./where.html" class="btn btn-primary my-10">DECOUVRIR</a>
```

```
    </div>
  </section>
</body>
</html>
```

Nous mettons un élément sémantique `section` pour notre partie centrale sur lequel nous plaçons une classe `vp`.

A l'intérieur nous plaçons le titre principal dans un `h1`.

Ensuite nous créons une `div` pour créer une séparateur.

Après, nous mettons un paragraphe de texte qui contiendrait normalement la description du lieu.

Enfin, nous plaçons un lien que nous allons transformer en bouton avec plusieurs classes.

Modification de `styles.css`

Nous allons maintenant créer les règles `CSS` :

```
/* General */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

html,
body {
  font-family: "Open Sans", sans-serif;
}

h1,
h2,
h3 {
  margin-bottom: 20px;
}

a {
  color: #222;
  text-decoration: none;
}

p {
  margin-bottom: 10px;
```

```
}

.my-10 {
  margin-top: 10px;
  margin-bottom: 10px;
}

.separator {
  margin: 20px auto;
  height: 10px;
  width: 100px;
  background: #111;
}

.btn {
  display: inline-block;
  font-size: 14px;
  padding: 15px 25px;
  border-radius: 30px;
  font-weight: 700;
}

.btn-primary {
  background: white;
  color: #222;
}

.btn-primary:hover {
  background: #222;
  color: white;
}

/* Header */

.header {
  background: black;
  overflow: auto;
  line-height: 1.7em;
}

.header nav h1 {
  float: left;
  padding: 20px;
  margin-bottom: 0;
}
```

```
.header nav h1 a {
  color: white;
}

.header nav ul {
  float: right;
  list-style: none;
}

.header nav ul li {
  float: left;
}

.header nav ul li a {
  color: white;
  padding: 20px;
  display: block;
}

.header nav ul li a:hover,
.active {
  background: #444;
}

/* value proposition */
.vp {
  height: 700px;
  background: url("https://dyma-images.s3.fr-par.scw.cloud/html-css/project
    center center;
  background-size: cover;
  text-align: center;
}

.vp-content {
  max-width: 600px;
  padding-top: 150px;
  color: white;
  margin: auto;
}

.vp-content h1 {
  font-size: 50px;
}
```

```
.vp-content p {  
  font-size: 18px;  
  letter-spacing: 0.5px;  
  line-height: 1.9em;  
}
```

Description des classes génériques

Nous allons commencer par décrire les classes génériques que nous mettons en place.

La classe `my-10` signifie `margin y`, donc verticales (en haut et en bas), de `10 px`. Donner un tel nom permet de savoir facilement ce que fait cette classe générique.

La classe `separator` permet de créer un séparateur qui est seulement un rectangle de `100px` par `10px` avec un arrière-plan noir et des marges verticales (en haut et en bas) de `20px`. Nous le centrons horizontalement en passant `auto` en deuxième valeur de la propriété `margin`.

La classe `btn` permet de créer un bouton en donnant au texte contenu du `padding` en haut en en bas de `15px` et à droite et à gauche de `25px`, elle permet également d'arrondir les angles avec `border-radius`. Elle dispose l'élément en `inline-block` pour pouvoir disposer plusieurs boutons sur la même ligne tout en respectant le `padding`. Enfin, elle donne au texte une taille de `14px` et une graisse de `700`.

La classe `btn-primary` s'utilisera toujours avec la classe `btn` et permet de donner une couleur blanche à l'arrière-plan et une couleur noire au texte.

La classe `btn-primary:hover` permet d'utiliser une pseudo-classe pour inverser la couleur de l'arrière-plan et la couleur du texte lorsque l'utilisateur passe son curseur sur le bouton.

Description des classes pour le contenu central

Nous allons ensuite décrire les classes mises en place pour le contenu central.

La classe `vp` permet de créer un bloc de `700px` de hauteur pour positionner une image en arrière-plan en utilisant une `url`. Nous centrons l'image verticalement et horizontalement avec `center center`. Nous utilisons `cover` pour recouvrir l'espace disponible.

La classe `vp-content` permet de mettre le texte en blanc, de le centrer horizontalement avec `margin: auto`. Elle permet également de mettre `150px` de `padding` au-dessus. Enfin, elle définit une largeur maximale à `600px`.

Etape 3 du projet

Vous pouvez retrouver l'étape 3 du projet ici :

<https://codesandbox.io/embed/htmlcss-c4-l3-pbtil>

