Les bases du CSS

Ibrahim ALAME

28 novembre 2024

1 Présentation du chapitre

1.1 Objectifs CSS

- Dans ce chapitre nous allons réaliser notre premier projet !
- Nous allons faire un site avec plusieurs pages de présentation d'un café fictif à Paris.
- Nous nous emploierons à vous montrer les meilleures pratiques de développement et nous mettrons en oeuvre tout ce que nous avons appris sur les langages HTML et CSS .

1.2 Création du projet

1. Commencez par créer un dossier, par exemple cafe .
2. Dans ce dossier créez un dossier images qui contiendra les images de notre site. Ensuite ouvrez le dossier cafe dans VS Code .
3. Créez un dossier css au même niveau que le dossier images.

1.2.1 Création du fichier styles.css

Créez un fichier `styles.css` dans le dossier css.

Il est très courant dans un projet front-end de créer un fichier `styles.css` pour contenir le CSS commun à toutes les pages.

1.2.2 Création du fichier index.html

Créez un fichier au premier niveau et appelez le `index.html`.

Dans les applications et les sites Web , `index.html` est le point d'entrée par convention. C'est-à-dire que c'est le premier fichier envoyé au navigateur.

Création des deux autres pages

Notre projet comporte deux autres pages : une pour le formulaire de contact, et l'autre pour la localisation.

Nous allons donc créer un fichier `where.html` et un fichier `contact.html`. Modification d'`index.html`

Dans le fichier `index.html` nous allons mettre :

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
```

```
    <title>Café Florette | Accueil</title>
    <link rel="stylesheet" href="css/styles.css" />
</head>
<body>

</body>
</html>
```

Nous utilisons le générateur Emmet en faisant ! .

Ensuite nous importons notre fichier `styles.css` en donnant un chemin relatif à partir du dossier courant (le dossier où se situe `index.html`).

1.2.3 Modification de styles.css

Dans le fichier `styles.css` nous allons mettre :

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

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

La première règle utilise le sélecteur universel * pour enlever les marges et le remplissage par défaut, notamment appliqués sur body . La deuxième règle sélectionne tous les titres de niveau `h1,h2` et `h3` et définit une marge de 20px vers le bas.

1.3 Mise en place du header

1.3.1 Modification de index.html

Nous allons commencer par mettre en place la structure de la partie header de notre page :

```
<!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&display=swap"
      rel="stylesheet"
    />
  </head>
  <body>
    <header>
      <nav>
        <h1><a href="./index.html">CAFE FLORETTE</a></h1>
        <ul>
          <li><a 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>
  </body>
</html>
```

- Nous ajoutons un lien pour charger la police Open Sans depuis une API de Google .
- Nous avons sélectionné trois graisses : 300, 400 et 700.
- Dans la partie body nous créons un élément sémantique header dans lequel nous plaçons un élément sémantique nav car il contiendra les liens pour naviguer sur les pages de notre site.
- Dans cet élément nav , nous plaçons le titre de la page h1 qui est cliquable car nous y imbriquons un élément a .
- Ainsi, les utilisateurs seront redirigés sur la page d'accueil lorsqu'ils cliquerons sur le titre.
- Ensuite, nous créons une liste non ordonnée ul , dans laquelle nous plaçons trois list items . Chaque li contient un lien dans un élément a pour pouvoir naviguer vers les autres pages du site.

1.3.2 Raccourci Emmet

Nous allons voir un nouveau raccourci Emmet qui permet de grouper des éléments pour appliquer par exemple * .

Ici nous voulons créer trois fois des li dans lesquels nous avons des a : `ul>(li>a)*3`

1.3.3 Modification de styles.css

Nous allons ajouter quelques règles CSS dans notre fichier styles.css :

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

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

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

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

1. Nous commençons par utiliser la police Open Sans chargée depuis Google pour tous nos éléments. Nous définissons une famille de polices de repli dans le cas où la police ne pourrait être chargée.
2. Ensuite, nous remplaçons les règles CSS par défaut pour les éléments a : à savoir le soulignage et le bleu.
3. Nous enlevons le soulignage en faisant text-decoration : none ; et nous changeons la couleur avec la propriété color .

1.3.4 Modification de index.html

Nous allons maintenant attaquer le style de notre header :

```html
<!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&display=swap"
      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>
  </body>
</html>
```

Pour cela, nous ajoutons deux classes qui nous permettront de définir des règles CSS .

1. Une première classe header qui nous permettra spécifiquement de cibler ce header (pour rappel, nous pouvons avoir plusieurs éléments sémantiques header dans une même page).
2. Une seconde classe, active , qui nous permettra de définir un style particulier pour le lien de l'accueil. Cela permettra d'indiquer à l'utilisateur qu'il a sélectionné le lien "Accueil" et qu'il est sur la page correspondante.

1.3.5 Modification de styles.css

Nous allons maintenant passer aux règles de style :

```
* {
  margin: 0;
}

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

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

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

/* Header */
```

```css
.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;
}
```

Nous allons reprendre ensemble les éléments essentiels.

- **Pourquoi avons-nous dû utiliser overflow-auto sur le parent des éléments positionnés en float ?**

  Le container header ne contient que des éléments qui sont positionnés en float .

Or, pour rappel, lorsque nous plaçons un élément en float , il est retiré du flux normal des éléments.

Du coup, l'élément header fait comme si il n'y avait pas d'éléments et il n'a donc pas de hauteur (pour rappel les éléments de bloc prennent la hauteur de leur contenu).

L'utilisation de overflow : auto permet la création d'un contexte de formatage de blocs et oblige l'élément à prendre en compte les éléments positionnés en float .

Ne vous attardez pas trop sur ce point car ce n'est pas l'approche la plus moderne. Nous verrons comment il est possible de réaliser la même chose avec le CSS le plus moderne lorsque nous verrons les boites flexibles et les grilles dans les chapitres suivants.

Sachez que c'était la meilleure manière de faire sans ces deux modules CSS plus modernes. Il est encore possible que vous la rencontriez sur des sites développés il y a quelques temps.

- **Pourquoi devons nous définir une line-height ?**

  Par défaut, les navigateurs prennent 1.2 (ce qui signifie que la valeur de font-size de l'élément est multiplié par 1.2).

  Donc comme un h1 dans une nav a une taille de 1.5em par défaut sa line-height sera de 1.2*1.5*16px donc 28.8 px . Car pour rappel, dans un navigateur, 1em vaut 16px par défaut.

  Et comme les éléments ont une taille de 1 em soit 16px , les liens auront une line-height de 1.2*1*16 px soit 19.2 px .

  Pour donner à nos éléments la même line-height , nous la fixons donc à 1.7em ce qui permet que les clics sur les liens puissent se faire sur toute la hauteur du header.

  Nous ne détaillerons pas plus le comportement des éléments en float car ce n'est pas utile pour vous. Nous verrons très en détails les grilles et les boites flexibles à la place.

1.4 Mise en place de la partie centrale

1.4.1 Modification de index.html

```
<!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&display=swap"
      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. Aspernatur
          nemo cupiditate earum eum porro unde sint ducimus vero voluptas,
          perferendis quasi quia accusamus omnis rerum. Neque sapiente quidem
          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 .

1. A l'intérieur nous plaçons le titre principal dans un h1 .
2. Ensuite nous créons une div pour créer un séparateur.
3. Après, nous mettons un paragraphe de texte qui contiendrait normalement la description du lieu.
4. Enfin, nous plaçons un lien que nous allons transformer en bouton avec plusieurs classes.

1.4.2 Modification de styles.css

Nous allons maintenant créer les règles CSS :

```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;
}
```

```css
.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/projects/project1/cafe1.jpg"
    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;
}
```

1.4.3 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 1 00px 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.

1.4.4 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 .