FORMATIONS

CSS

DÉBUTANT

FORMATION CSS3 ENRICHIE

Le style et la beauté du web

CHAPITRE 1 — Introduction au CSS

CSS (Cascading Style Sheets) est le langage qui permet de styliser vos pages HTML. Sans CSS, votre site serait simplement du texte noir sur fond blanc, sans aucune mise en forme.

> 💡 **Comprendre CSS**
>
> CSS fonctionne comme un styliste pour votre HTML. Le HTML crée la structure (les éléments), et le CSS leur donne du style (couleurs, tailles, positions). C'est comme habiller un mannequin : HTML = le corps, CSS = les vêtements.

Les 3 façons d'ajouter du CSS

1. CSS INLINE (dans la balise HTML)

C'est la méthode la plus simple mais la moins recommandée pour de gros projets.

📄 **exemple-inline.html**

```
<!DOCTYPE html>
<html>
<body>

  <!-- Le style est directement dans la balise -->
  <h1 style="color: blue; font-size: 36px;">Mon titre bleu</h1>

  <p style="color: red; background-color: yellow;">
    Paragraphe rouge sur fond jaune
  </p>

</body>
</html>
```

2. CSS INTERNE (dans la balise <style>)

Le CSS est placé dans le <head> de la page HTML. Pratique pour une seule page.

📄 **exemple-interne.html**

```
<!DOCTYPE html>
<html>
<head>
  <style>
    /* Les règles CSS sont ici */
    h1 {
      color: blue;
      font-size: 36px;
      text-align: center;
    }

    p {
      color: #333;
      line-height: 1.6;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h1>Mon titre stylé</h1>
  <p>Mon paragraphe avec du style.</p>
</body>
</html>
```

3. CSS EXTERNE (fichier séparé) ⭐ RECOMMANDÉ

Le CSS est dans un fichier .css séparé. C'est la MEILLEURE méthode pour les vrais projets.

📄 **index.html**

```
<!DOCTYPE html>
<html>
<head>
  <!-- Lien vers le fichier CSS externe -->
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Mon titre</h1>
  <p>Mon paragraphe</p>
```

```
</body>
</html>
```

📄 **style.css**

```
/* Fichier CSS séparé - Plus propre et réutilisable */

h1 {
  color: blue;
  font-size: 36px;
}

p {
  color: #333;
  font-size: 18px;
}
```

✅ **Pourquoi CSS externe ?**

• Réutilisable sur plusieurs pages • Code plus propre et organisé • Facile à maintenir • Le navigateur le met en cache (plus rapide) • Séparation structure/style (bonne pratique)

CHAPITRE 2 — Les Sélecteurs CSS

Les sélecteurs CSS permettent de cibler précisément les éléments HTML que vous voulez styliser. C'est comme pointer du doigt l'élément à habiller.

1. Sélecteur d'élément (balise)

Cible TOUS les éléments d'un type donné.

📄 **HTML**

```
<h1>Titre 1</h1>
<h1>Titre 2</h1>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
```

📄 **CSS**

```
/* Tous les <h1> seront bleus */
h1 {
  color: blue;
}

/* Tous les <p> seront gris */
p {
  color: #666;
}
```

2. Sélecteur de classe (.nomClasse)

Cible les éléments qui ont une classe spécifique. Réutilisable !

📄 **HTML**

```
<p class="important">Texte important</p>
<p>Texte normal</p>
<p class="important">Autre texte important</p>
```

```
<div class="card">Carte 1</div>
<div class="card">Carte 2</div>
```

📄 **CSS**

```
/* Tous les éléments avec class="important" */
.important {
  color: red;
  font-weight: bold;
}

/* Tous les éléments avec class="card" */
.card {
  background: #f0f0f0;
  padding: 20px;
  border-radius: 10px;
}
```

3. Sélecteur d'ID (#nomID)

Cible UN seul élément unique. L'ID doit être unique sur la page !

📄 **HTML**

```
<header id="main-header">En-tête principal</header>
<div id="content">Contenu</div>
<footer id="main-footer">Pied de page</footer>
```

📄 **CSS**

```
/* L'élément avec id="main-header" */
#main-header {
  background: #333;
  color: white;
```

```
  padding: 20px;
}

#content {
  min-height: 500px;
}
```

⚠️ Classe vs ID

• CLASSE (.class) : Réutilisable, pour plusieurs éléments • ID (#id) : Unique, pour UN SEUL élément • En général, utilisez plutôt les CLASSES (plus flexible)

CHAPITRE 3 — FLEXBOX (Layout moderne)

Flexbox est la méthode MODERNE pour créer des mises en page. Elle permet d'aligner et distribuer l'espace entre les éléments facilement.

Concept de base

Flexbox fonctionne avec un CONTENEUR (parent) et des ITEMS (enfants). Le conteneur contrôle comment les enfants sont disposés.

📄 **HTML - Structure**

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

📄 **CSS - Flexbox basique**

```
/* Le CONTENEUR devient flex */
.container {
  display: flex;

  /* Les items seront alignés en ligne (→) */
  /* Par défaut flex-direction: row; */
}

/* Style des items */
.item {
  background: #3498db;
  color: white;
  padding: 20px;
  margin: 10px;
  border-radius: 5px;
}
```

RÉSULTAT : Les 3 items seront côte à côte →

[Item 1] [Item 2] [Item 3]

justify-content (alignement horizontal)

📄 **CSS - Différentes options**

```
/* Items au CENTRE */
.container {
  display: flex;
  justify-content: center;
}
/* Résultat :     [Item 1] [Item 2] [Item 3]     */

/* Items ESPACÉS (space-between) */
.container {
  display: flex;
  justify-content: space-between;
}
/* Résultat : [Item 1]          [Item 2]          [Item 3] */

/* Items ESPACÉS UNIFORMÉMENT (space-around) */
.container {
  display: flex;
  justify-content: space-around;
}
/* Résultat :   [Item 1]    [Item 2]    [Item 3]   */

/* Items à DROITE */
.container {
  display: flex;
  justify-content: flex-end;
}
/* Résultat :                    [Item 1] [Item 2] [Item 3] */
```

align-items (alignement vertical)

📄 **CSS**

```
/* Items centrés VERTICALEMENT */
.container {
  display: flex;
  height: 300px; /* Important d'avoir une hauteur */
  align-items: center;
}

/* Autres options : */
/* align-items: flex-start;  → Haut */
/* align-items: flex-end;    → Bas */
/* align-items: stretch;     → Items prennent toute la hauteur */
```

Exemple pratique : Navigation

📄 **HTML complet**

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
    }

    /* Navigation avec Flexbox */
    .navbar {
      display: flex;
      justify-content: space-between;
      align-items: center;
      background: #333;
      padding: 15px 30px;
    }

    .logo {
      color: white;
      font-size: 24px;
```

```
      font-weight: bold;
    }

    .nav-links {
      display: flex;
      gap: 20px; /* Espace entre les liens */
    }

    .nav-links a {
      color: white;
      text-decoration: none;
      padding: 10px 15px;
      border-radius: 5px;
      transition: background 0.3s;
    }

    .nav-links a:hover {
      background: #555;
    }
  </style>
</head>
<body>

  <nav class="navbar">
    <div class="logo">MonSite</div>
    <div class="nav-links">
      <a href="#">Accueil</a>
      <a href="#">Services</a>
      <a href="#">Portfolio</a>
      <a href="#">Contact</a>
    </div>
  </nav>

</body>
</html>
```

💡 Astuce Flexbox

gap: 20px; est super pratique ! Au lieu d'ajouter des margins sur chaque item, gap crée automatiquement l'espace entre tous les items. Disponible depuis 2021 dans tous les navigateurs modernes.

✨ Formation CSS ENRICHIE créée avec succès ! ✨