

Mise en forme du HTML avec CSS

1. Introduction

Après avoir découvert comment le HTML structure l'information, il est nécessaire de séparer le contenu de sa présentation. C'est précisément le rôle de CSS (Cascading Style Sheets) : définir l'apparence des pages – couleurs, tailles, marges, mises en page – tout en laissant au HTML le soin de porter le sens. Cette séparation facilite la maintenance, la réutilisation des styles et l'accessibilité.

2. CSS et la couche présentation

CSS, normalisé pour la première fois en 1996, propose un vocabulaire de propriétés (par exemple "color", "font-size", "margin") auxquelles on assigne des valeurs ("blue", "16px", etc.). Ces propriétés s'appliquent à des éléments ciblés grâce à des sélecteurs. L'ensemble sélecteur–propriétés–valeurs constitue une règle de style.

Les principaux éléments d'un style sont :

- ▼ Les **sélecteurs** : déterminent à quel élément un style doit être appliqué.
- ▼ Les **propriétés** : déterminent le style qui va être appliqué à un élément.
- ▼ Les **valeurs** : déterminent le comportement

3. Anatomie d'une règle CSS

Une règle CSS associe un "sélecteur" à un bloc d'instructions : chaque instruction est une paire propriété : valeur terminée par un point-virgule. Le sélecteur désigne le ou les éléments visés ; le navigateur calcule ensuite la présentation effective en tenant compte de l'ordre des règles et de leur spécificité.

```
p {  
  color : blue ;  
  font-size : 16px ;  
}
```

Règle de définition du style de l'élément p

4. Où définir les styles ?

On peut définir des styles de trois manières :

1. en ligne, via l'attribut "style" directement sur une balise HTML ;
2. dans l'en-tête du document, au sein d'un élément <style> ;
3. dans un fichier externe .css lié à la page. La troisième approche est recommandée pour mutualiser les styles et réduire la duplication.

Lier un fichier HTML avec un fichier CSS est réalisé grâce à l'élément `link` :

```
<link rel='stylesheet' href='fichiercss.css'/>
```

5. Comment définir/cibler les styles ?

Plusieurs familles de sélecteurs existent : les sélecteurs de balises (p, h1, ul), l'identifiant unique (#entete), les classes réutilisables (.important), ainsi que des conteneurs génériques comme <div> (bloc) et (inline) qui n'ajoutent pas de sens, mais facilitent l'application de styles.

Les styles peuvent :

- ▼ redéfinir le style d'une **balise existante**.
- ▼ définir le style d'un élément identifié par l'attribut **id**.
- ▼ être définis grâce aux éléments **div** et **span**.
- ▼ être définis grâce à des **classes**.

5.1. Redéfinir les styles existants

Il est courant de redéfinir l'aspect des balises HTML de base pour harmoniser la charte graphique. Une règle sur "h1" affecte tous les titres de niveau 1 ; des règles sur "p" contrôlent la typographie générale des paragraphes (couleurs, interlignage, alignement, etc.).

```
h1 {  
  color: blueviolet;  
  font-family: 'Franklin Gothic Medium', serif;  
  font-size: 18px;  
}  
  
/* Commentaire : la mise en forme ci-dessous  
s'applique à tous les paragraphes */  
p {  
  color: rgb(14, 153, 44);  
  text-decoration: underline;  
}
```

Exemple de redéfinition du style des éléments h1 et p

5.2. Lier un document HTML à une feuille CSS

Pour appliquer des styles externes, on relie la page HTML au fichier CSS via l'élément `<link>` placé dans la section `<head>`. Cela permet de partager la même feuille de style entre plusieurs pages.

```
<!doctype html>
<html>
  <head>
    <title>Ici le titre de la page dans l'onglet du navigateur</title>
    <!-- Commentaire : la ligne ci-dessous
         définit le jeu de caractères employés -->
    <meta charset=utf-8></meta>
    <link rel='stylesheet' href='exempleBase.css' />
  </head>
  <body>
    <h1>L'entête de cette page Web est mis en avant</h1>
    <p>Permet de définir un paragraphe.</p>
  </body>
</html>
```

Exemple de d'utilisation de la feuille de style "exempleBase.css" dans une page HTML

5.3. Cibler par identifiant (id)

Un identifiant permet de styliser un élément unique de la page. En CSS, on le cible avec le préfixe `#`. Attention : un id ne doit apparaître qu'une seule fois dans un document.

L'identifiant d'une balise peut être utilisé pour mettre en forme un élément.

```
<ul id='listel'>
  <li>premier</li>
  <li>deuxième</li>
  <li>troisième</li>
</ul>
```

Le CSS y fera référence grâce au symbole `#` :

```
#listel {
  font-size : 16px;
  color : blue;
}
```

Exemple d'utilisation d'un identifiant pour appliquer un style à un élément

5.4. Définir des classes réutilisables

Les classes (préfixe `.`) sont idéales pour partager une même mise en forme entre plusieurs éléments. Elles favorisent la cohérence visuelle et limitent la duplication de règles.

Il est possible de définir des "classes" de mise en forme.
Le nom de la classe est précédé du signe . (point) :

```
.important {  
  font-size : 16px;  
  color : red;  
}
```

Le HTML y fera référence grâce à l'attribut class :

```
<p class='important'>Voici une liste d'éléments :</p>  
<ul>  
  <li>premier</li>  
  <li class='important'>deuxième</li>  
  <li>troisième</li>  
</ul>
```

Exemple d'utilisation d'une classe pour appliquer un style à un élément

5.5. Les conteneurs <div> et

<div> regroupe des contenus en bloc, tandis que cible une portion en ligne au sein d'un texte. On peut ainsi appliquer des styles différents au conteneur et à un fragment interne.

On définit 2 classes dans le fichier CSS :

```
.encart {  
  padding: 1rem;  
  background: #f6f6f6;  
}  
.accent {  
  color: #c00;  
  font-weight: bold;  
}
```

Le HTML y fera référence en utilisant div et span :

```
<div class="encart">  
  Ceci est un bloc contenant du texte et un <span class="accent">élément en ligne</span>.  
</div>
```

Exemple d'utilisation de div et span pour appliquer un style

6. Cascade et spécificité : la priorité des styles

Quand plusieurs règles s'appliquent au même élément, le navigateur suit des règles de priorité : la spécificité d'abord (id > classe > balise), puis l'ordre d'apparition (la règle la plus récente l'emporte). Cette logique est au cœur de la "cascade" et explique pourquoi une déclaration peut sembler ignorée si une autre plus spécifique s'applique.

- ▼ Les classes sont prioritaires par rapport aux styles standards définis à l'aide des sélecteurs simples (p, h1, ...).
- ▼ L'id est prioritaire sur la classe.
- ▼ Si deux classes de même nom sont définies avec des propriétés différentes, c'est la dernière classe définie qui sera prise en compte.
- ▼ Pour en savoir plus : <https://algocool.fr/priorite-selecteurs-css3/>

7. Conclusion

CSS complète le HTML en contrôlant finement la présentation des pages. En maîtrisant l'anatomie d'une règle, les différents modes d'inclusion (inline, interne, externe) et les mécanismes de sélecteurs et de spécificité, on dispose des bases indispensables pour mettre en œuvre une charte graphique robuste, réutilisable et accessible.

8. Crédits / Sources

Ce document a été réalisé d'après un travail de Fabrice Mourlin (UPEC)

Références :

- w3schools.com – Référence CSS
- MDN Web Docs – Guides et références CSS
- Xul.fr – Structure HTML5 et rappels CSS
- Wikipedia – Feuilles de style en cascade