

Introduction à CSS3

Dans le précédent chapitre, nous avons vu comment structurer les pages web de façon cohérente. Il reste maintenant à comprendre comment mettre en forme les différents éléments, les différents contenus web.

1. Feuilles de style

Les feuilles de style permettent de mettre en forme les différents éléments HTML des pages d'une application web. En effet, elles permettent de définir des propriétés telles que les couleurs, les polices de caractères, le positionnement des éléments... Elles ont également vocation à centraliser le code CSS.

Il est important de comprendre comment écrire une feuille de style avec les différentes règles et les propriétés à appliquer.

Mais avant de pouvoir écrire une feuille de style, il faut savoir ce que représentent une propriété CSS et un sélecteur.

a. Code CSS

Le code CSS, c'est entre autres lui qui permet de définir la couleur d'un texte donné, de définir une police en particulier pour les titres ou le corps de texte dans une page, ou encore la couleur de fond d'un document.

Le code CSS s'écrit généralement dans des fichiers qui portent l'extension `.css`. Dans une application web, dans l'arborescence du projet, on crée toujours un répertoire nommé **css** sous le répertoire qui correspond aux pages web de l'application, dans lequel on place les fichiers `.css`.

La structure d'un fichier CSS est simple et se compose de plusieurs règles. Une règle suit la syntaxe suivante, à savoir qu'on utilise une propriété (par exemple, la propriété `color` qui représente la couleur du texte), puis une valeur et `;` pour terminer l'instruction.

```
{  
    propriété: valeur;  
}
```

Pour mettre une ligne de code en commentaire dans un fichier CSS, on pourra utiliser `//` en début de ligne. Pour mettre plusieurs lignes de code entre parenthèses, on utilisera `/*` en début de code et `*/` à la fin du code.

b. Définition d'une règle de style

Il est important de comprendre comment écrire une feuille de style en écrivant des règles de style CSS et donc de manipuler aisément les propriétés et leurs valeurs possibles associées.

Déjà, à quoi sert une règle CSS ? Concrètement, une règle sert à appliquer des styles à une balise HTML.

Chaque règle débute par un sélecteur. Il s'agit de l'élément ou balise HTML concerné par la règle.

Par exemple, si on veut sélectionner les paragraphes de la page, on mentionnera l'élément `p` au début de la règle.

Ensuite, on ajoute des accolades ouvrantes et fermantes (`{` et `}`). Entre ces accolades, on va définir une ou plusieurs déclarations du type "propriété: valeur".

L'objectif est d'appliquer des effets graphiques aux balises sélectionnées. Chaque déclaration se termine par un

point-virgule (;).

Voici un exemple de règle CSS; c'est `h1` qui est le sélecteur. Dans cette règle, on utilise deux propriétés `text-align` et `color`. Puis, on leur affecte à chacune une valeur :

```
h1{
  text-align: right;
  color: #FFF;
}
```

c. Inclusion d'un fichier CSS

La feuille de style CSS externe, à l'inverse de la feuille de style interne, est un fichier `.css` externalisé potentiellement commun à plusieurs pages de l'application web.

Pour inclure une feuille de style ou un fichier `.css` externe dans une page, il faut utiliser la balise `<link>` entre les balises `<head></head>` d'un document HTML comme indiqué ci-dessous :

```
<html>
<head>
  <link type= « text/css » rel = « stylesheet » href= « css/myfile.css »/>
</head>
<body>
  ...
</body>
</html>
```

Il est également possible d'inclure directement du code CSS dans la balise `<head>` à l'intérieur des balises `<style></style>`. Cependant, cette mise en œuvre est à éviter dans un souci de séparation distincte du HTML et du CSS qui permet une meilleure maintenabilité de l'application.

2. Application d'un style

Pour appliquer un style donné à un élément donné, il faut au préalable savoir sélectionner cet élément.

Un élément pourra être sélectionné à partir de son identifiant, de son type ou de sa classe. Il sera également possible de combiner les différents types de sélection.

First Paragraph

a. Sélection d'un élément avec son identifiant

Il est possible de sélectionner un élément à partir de son identifiant (ou « id »).

Ainsi, la syntaxe à utiliser est alors la suivante : il faut faire précéder le nom de l'id de l'élément du symbole `#` :

```
#id{  
  
}
```

Puis, pour styliser l'élément une fois sélectionné, il faut spécifier la propriété et la valeur à l'intérieur des accolades.

Voici un exemple, avec ce code HTML et l'élément `p` qui a pour id `paragraph` :

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    <p id=«paragraph»> First paragraph </p>  
  </body>  
</html>  
  
</html>
```

Voici la feuille de style associée ou le code CSS associé ; on souhaite que la couleur du texte du paragraphe soit rouge :

```
#paragraph{  
  color: red;  
}
```

b. Sélection d'un élément d'après sa classe

Il est possible de sélectionner un élément à partir de sa classe.

Ainsi, la syntaxe à utiliser est alors la suivante ; il faut faire précéder le nom de la classe de l'élément d'un point :

```
.class{  
  
}
```

Voici un exemple, avec ce code HTML et l'élément `p` qui a pour classe `mainparagraph` :

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    <p class=«mainparagraph»>First paragraph</p>  
  </body>  
</html>  
  
</html>
```

Voici le code CSS associé ; on souhaite que la couleur du texte soit rouge :

```
.mainparagraph{
    color: red;
}
```

c. Sélection d'un élément d'après son type

Il est possible de sélectionner un élément directement à partir de son type.

Ainsi, la syntaxe à utiliser est alors la suivante :

```
type{

}
```

Voici un exemple, avec ce code HTML et l'élément de type p :

```
<html>
  <head>
    ...
  </head>
  <body>
    <p>First paragraph</p>
  </body>
</html>

</html>
```

Voici le code CSS associé ; on souhaite que la couleur du texte soit rouge :

```
P{
    color: red;
}
```

d. Autres sélecteurs ou pseudoclasses

Il existe de nombreux autres types de sélecteurs tels que les pseudoclasses pour les liens hypertextes :

- `:active`, qui désigne un lien actif, c'est-à-dire lorsque la souris pointe sur ce lien.
- `:hover`, qui désigne le survol d'un élément.
- `:link`, qui désigne un lien pas encore visité.
- `:visited`, qui désigne un lien qui a été visité.

Voici un exemple si l'on a un lien hypertexte balise `Lien1` dans notre document HTML. Si l'on souhaite que la couleur du texte devienne rouge lors de son survol par la souris, il faudra écrire dans la feuille

de style :

```
a:hover{
    color: red;
}
```

Il existe également les autres pseudoclasses telles que :

- `:first-child`, qui désigne le premier enfant d'une balise.
- `:focus`, qui désigne l'élément qui a le focus.

Il existe également des pseudo-éléments tels que :

- `:before`, qui désigne l'élément avant la balise spécifiée.
- `:after`, qui désigne l'élément après la balise spécifiée.

e. Ordre de priorité des styles

Il est intéressant de savoir quelle règle CSS s'applique à un élément quand deux règles sont en conflit.

Dans les faits, les sélecteurs CSS ont un poids et une priorité. Cela permet aux navigateurs de savoir quel style appliquer.

Voici quelques notions à connaître concernant l'ordre des priorités des styles :

- Lorsque l'on utilise l'attribut `style` directement dans un élément, via une balise HTML, alors le style défini à l'intérieur de l'élément est prioritaire sur tout le reste.
- Ensuite, priorité au style à appliquer à l'élément sélectionné à partir de son id avec le symbole `#`.
- Ensuite, priorité au style à appliquer à l'élément sélectionné à partir de sa classe avec le symbole `.`.
- Et enfin, priorité la plus basse si conflit sur l'application de style pour un élément sélectionné à partir de son type ou nom de balise (par exemple `p`, `div`...).

Cet exemple illustre l'ordre des priorités :

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .mainpragraph{
        color: orange;
      }

      #paragraph{
        color: red;
      }

      p{
        color: yellow;
      }
    </style>
  </head>
  <body>
    <div class="mainpragraph">
      <p id="paragraph">
        ...
      </p>
    </div>
  </body>
</html>
```

```
    }  
  </style>  
</head>  
<body>  
  <p id=«paragraph» class=«mainparagraph»  
style=«color:green»>First paragraph</p>  
</body>  
</html>  
  
</html>
```

Ainsi, par défaut, si on se base sur l'ordre des priorités, la couleur du texte du paragraphe sera verte. Si le style n'avait pas été précisé à l'intérieur de la balise `p`, alors la couleur du texte du paragraphe serait rouge : priorité à la règle associée à la sélection à partir de l'id.