

Partie 3 : Les CSS 3

Chapitre 3-1 Intégrer les styles CSS

1. Le rôle des CSS

Rappelons que les **CSS**, *Cascading Style Sheets*, permettent de mettre en forme le contenu des éléments HTML et s'occupent aussi de la mise en page des sites web. Avec les CSS, nous avons donc deux objectifs bien définis :

- La mise en forme qui concerne le formatage des textes, avec par exemple l'application d'une couleur, le changement de casse des caractères, la mise en évidence des paragraphes de texte avec la mise en forme de l'interligne ou le retrait de première ligne...
- La mise en page concerne l'agencement des blocs de contenu HTML de la page, avec une barre de navigation, un en-tête, un pied de page, une zone d'affichage des contenus...

Les CSS permettent de modifier dynamiquement la mise en page selon le média de diffusion. C'est-à-dire que la mise en page va être différente suivant que vous consultez le site sur l'écran d'un ordinateur, d'une tablette ou d'un smartphone.

Rappelons que les CSS3 sont divisées en modules indépendants et qu'elles sont élaborés avec leur propre vitesse.

Il est maintenant bien acquis, et depuis longtemps déjà, qu'il faut séparer les CSS de la structure du HTML. Nous allons donc voir dans ce chapitre où placer les règles CSS qui constituent une feuille de style.

2. Les styles intégrés dans un élément HTML

La première possibilité d'intégrer des styles CSS est de définir une règle directement dans l'élément HTML concerné. Dans ce cas, cette règle ne s'applique qu'à cet élément et à nul autre.

Pour cela, nous devons utiliser l'attribut `style` dans l'élément HTML souhaité et indiquer la propriété et la valeur voulues.

Dans cet exemple, nous appliquons de l'italique au premier titre `<h2>` :

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Ma page web</title>
</head>
<body>
  <h2 style="font-style: italic">Tristique Cursus Commodo Ligula</h2>
  <p>Cras justo odio, dapibus ac facilisis...</p>
  <h2>Tortor Ullamcorper Etiam Nibh</h2>
  <p>Nullam id dolor id nibh ultricies...</p>
</body>
</html>
```

Voici l'affichage obtenu :

Tristique Cursus Commodo Ligula

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Nullam id dolor id nibh ultricies vehicula ut id elit. Aenean lacinia bibendum nulla sed consectetur. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vitae elit libero, a pharetra augue.

Tortor Ullamcorper Etiam Nibh

Nullam id dolor id nibh ultricies vehicula ut id elit. Etiam porta sem malesuada magna mollis euismod. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam porta sem malesuada magna mollis euismod. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.

Ainsi, vous voyez que la portée de cette règle CSS est des plus réduites, puisqu'elle ne s'applique qu'à l'élément HTML dans lequel elle est définie. Elle ne peut pas être utilisée ailleurs.

3. Les styles définis dans la page

La deuxième possibilité est de définir les règles CSS dans une page HTML. Dans ce cas, les règles ne peuvent s'appliquer qu'aux éléments HTML contenus dans cette page HTML et nulle part ailleurs, dans aucun autre fichier HTML. Les règles CSS se définissent dans l'élément `<head>` dans un élément `<style>`.

Voici un exemple :

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Ma page web</title>
    <style>
        .titre-article {
            font-style: italic;
            text-transform: uppercase;
        }
    </style>
</head>
<body>
    <h2 class="titre-article">Tristique Cursus Commodo Ligula</h2>
    <p>Cras justo odio, dapibus ac facilisis...</p>
    <h2 class="titre-article">Tortor Ullamcorper Etiam Nibh</h2>
    <p>Nullam id dolor id nibh ultricies...</p>
</body>
</html>
```

Dans l'élément `<style>`, nous n'avons qu'une seule règle CSS et elle est nommée `.titre-article`. Cette règle définit un texte en italique et passe en majuscules tout le texte sur lequel elle sera appliquée.

Dans la page HTML, dans les deux éléments `<h2>`, nous appliquons cette règle avec l'attribut `class`.

Voici l'affichage obtenu :

TRISTIQUE CURSUS COMMODO LIGULA

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Nullam id dolor id nibh ultricies vehicula ut id elit. Aenean lacinia bibendum nulla sed consectetur. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vitae elit libero, a pharetra augue.

TORTOR ULLAMCORPER ETIAM NIBH

Nullam id dolor id nibh ultricies vehicula ut id elit. Etiam porta sem malesuada magna mollis euismod. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam porta sem malesuada magna mollis euismod. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.

Vous concevez donc parfaitement que la portée des règles CSS est plus importante que précédemment, puisqu'elles peuvent s'appliquer à tous les éléments HTML présents dans la page web.

4. Les styles définis dans un fichier .css

La troisième possibilité est de déclarer les règles CSS dans un fichier séparé des pages HTML. Ce fichier aura alors comme extension **.css**. Ce fichier ne contiendra rien d'autre que les règles CSS voulues. Ensuite, nous devrons lier ce fichier CSS aux pages HTML voulues pour appliquer les règles CSS aux éléments HTML voulus.

Voici un exemple très simple d'un fichier CSS, nommé **styles.css**, qui ne contient que deux règles CSS :

```
.titre-article {  
    margin-left: 20px;  
    padding-left: 10px;  
    border-left: 5px solid #000;  
}  
  
.nom {  
    font-style: italic;  
}
```

Voici le fichier HTML :

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Ma page web</title>
    <link href="styles.css" rel="stylesheet">
</head>
<body>
    <h2 class="titre-article">Tristique Cursus Commodo Ligula</h2>
    <p>Cras justo odio ... id nibh <span class="nom">Ultricies
    vehicula</span> ut id elit...</p>
    <h2 class="titre-article">Tortor Ullamcorper Etiam Nibh</h2>
    <p>Nullam id dolor ... <span class="nom">Etiam porta</span>
    sem malesuada magna mollis euismod...</p>
</body>
</html>
```

La liaison entre le fichier HTML et le fichier CSS se fait dans l'élément `<head>`. C'est l'élément `<link>` qui est utilisé :

- L'attribut `href="styles.css"` indique le chemin d'accès au fichier CSS.
- L'attribut `rel="stylesheet"` précise le type de relation. Ici, il s'agit d'une relation à une feuille de style.
- L'attribut `type="text/css"` est facultatif car le type les feuilles de style est par défaut CSS.

L'application des styles se fait de nouveau avec l'attribut `class` dans les éléments HTML voulus.

Voici l'affichage obtenu :

Tristique Cursus Commodo Ligula

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Nullam id dolor id nibh *Ultricies vehicula* ut id elit. Aenean lacinia bibendum nulla sed consectetur. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vitae elit libero, a pharetra augue.

Tortor Ullamcorper Etiam Nibh

Nullam id dolor id nibh ultricies vehicula ut id elit. *Etiam porta* sem malesuada magna mollis euismod. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam porta sem malesuada magna mollis euismod. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.

Vous pouvez bien sûr lier plusieurs fichiers CSS à une même page HTML et un même fichier CSS peut être lié à plusieurs pages HTML. Vous comprenez bien que la portée des styles CSS est ici la plus importante.

5. Les styles importés

La dernière possibilité d'intégrer une feuille de style CSS est d'utiliser la règle `@import`. Attention, notez que cette règle a été implémentée dans les CSS 2. Il ne s'agit donc pas d'une règle HTML ni d'un élément HTML.

Cette règle permet d'importer un fichier .css dans un autre fichier .css. Voici sa syntaxe qui se place usuellement au début d'un fichier .css :

```
@import url("autres-styles.css")
```

La règle `@import` utilise la propriété `url()` pour indiquer le chemin d'accès au fichier .css qui doit être importé.

Vous pouvez aussi utiliser la règle `@import` dans un fichier .html, dans l'élément `<head>`. Voici la syntaxe à utiliser :

```
<style>
  @import url("autres-styles.css");
</style>
```

Mais pour des raisons de performance, cette méthode est moins utilisée.

Chapitre 3-2

Définir les styles CSS

1. La structure d'une règle de style

1.1 La terminologie des CSS

Nous parlerons de CSS, de style, de règle, de déclaration, de propriété et de valeur. Il convient de bien définir cette terminologie afin d'utiliser les bons termes.

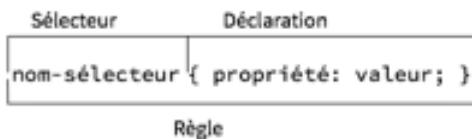
Les **CSS**, pour *Cascading Style Sheets*, sont une technologie développée par le W3C qui permet de mettre en forme et de mettre en page les pages des sites web structurés en HTML.

Un **style** est une mise en forme, ou une mise en page utilisant les CSS, qui est mémorisée et qui, ensuite, peut être appliquée à un ou plusieurs éléments HTML.

Une **règle** CSS permet la création d'un style. Cette règle est créée avec une syntaxe précise utilisant un **sélecteur** et une **déclaration**. Cette dernière est constituée de **propriétés** et de **valeurs**.

1.2 Définir une règle de style

Un style CSS est bâti avec une règle. Cette règle est constituée de plusieurs parties. Voici un petit schéma illustrant une règle CSS :



- La **règle** est constituée d'un sélecteur et d'une déclaration.
- Le **sélecteur** indique la portée du style, c'est-à-dire sur quel élément HTML peut s'appliquer le style créé. Il existe beaucoup de sélecteurs, nous les étudierons dans un prochain chapitre.
- La **déclaration** est indiquée entre accolades.

C'est dans cette déclaration que sont indiquées la ou les **propriétés** CSS utilisées. Chaque propriété utilise une ou plusieurs **valeurs**. La propriété est séparée de la ou des valeurs par le caractère deux-points :. Chaque ligne dans la déclaration se termine par le caractère point-virgule ;.

Pour plus de visibilité et de lisibilité, il est d'usage d'aller à la ligne après l'accolade ouvrante et avant l'accolade fermante. Ainsi, chaque couple propriété/valeur se trouve sur une seule ligne.

Voici une règle CSS n'utilisant aucun espace ni retour à la ligne. La visibilité n'est pas au rendez-vous :

```
nom-selecteur{propriete1:valeur1;propriete2:valeur2;propriete3: valeur3;}
```

Voici une règle CSS utilisant plusieurs lignes dans sa déclaration :

```
nom-selecteur {  
    propriete1: valeur1 ;  
    propriete2: valeur2 ;  
    propriete3: valeur3 ;  
}
```

Chapitre 4

Le texte

1. L'objectif

Bootstrap propose une série de règles CSS pour la mise en forme typographique du texte. Nous allons passer en revue toutes ces règles CSS.

2. L'uniformisation des affichages

2.1 Le rendu des navigateurs

Comme nous l'avons déjà évoqué précédemment, les navigateurs n'ont pas tous le même rendu pour toutes les propriétés des CSS. Il existe encore des différences plus ou moins importantes. Il convient donc d'uniformiser ces rendus.

C'est l'objectif des styles **Reboot** de Bootstrap. Sachez que **Reboot** est basé sur la feuille de styles **Normalize** de Nicolas Gallagher (<https://necolas.github.io/normalize.css/>).

2.2 Le corps de la page

Avec **Reboot**, les éléments `<html>` et `<body>` se voient attribuer plusieurs règles par défaut.

Les polices de caractères ciblent les principaux navigateurs pour un affichage optimal. Bootstrap utilise **San Francisco** pour iOS, **Segoe UI** pour Windows, **Roboto** pour Android, et **Helvetica Neue** et **Arial** pour les autres OS. La taille de caractères est basée sur 16 pixels, avec la valeur 1 rem.

L'élément `<body>` utilise un interligne `line-height: 1.5`, une couleur de caractères sombre `color: #212529`, un alignement du texte `text-align: left` et une couleur d'arrière-plan blanche `background-color: #fff`.

Pour la mise en page, tous les éléments utilisent la propriété `box-sizing: border-box`.

Bien sûr, tous les éléments inclus vont hériter de ces propriétés globales, sauf s'ils redéfinissent leurs propres valeurs.

2.3 Les marges des en-têtes, des paragraphes, des listes et du texte préformaté

Les en-têtes `<h1>` à `<h6>` et les paragraphes `<p>` n'ont pas de marge supérieure. La marge inférieure des en-têtes a pour valeur `margin-bottom: .5rem` et celle des paragraphes est définie à `1rem`.

Les listes ``, `` et `<dl>` n'ont pas de marge supérieure, mais une marge inférieure définie à `margin-bottom: 1rem`. Les listes imbriquées n'ont pas de marge inférieure.

Le texte préformaté `<pre>` n'a pas de marge supérieure, mais une marge inférieure définie à `margin-bottom: 1rem`.

2.4 D'autres éléments

Dans les tableaux, les titres, l'alignement du texte, les bordures et le remplissage sont uniformisés. Les formulaires ont de nombreuses propriétés impactées par le **Reboot**. D'autres éléments, comme `<address>`, `<blockquote>`, `<abbr>` et `<summary>`, ont aussi des propriétés ajustées.

3. Les règles CSS pour les en-têtes

Les exemples à télécharger sont dans le dossier **04-Typographie- 01**.

3.1 Les sélecteurs d'éléments des en-têtes

■ Remarque

Bootstrap propose une mise en forme de tous les en-têtes, de `<h1>` à `<h6>`.

Voici les règles CSS communes aux en-têtes :

```
h6, h5, h4, h3, h2, h1 {  
  margin-top: 0;  
  margin-bottom: 0.5rem;  
  font-weight: 500;  
  line-height: 1.2;  
}
```

Voici les tailles de caractères utilisées :

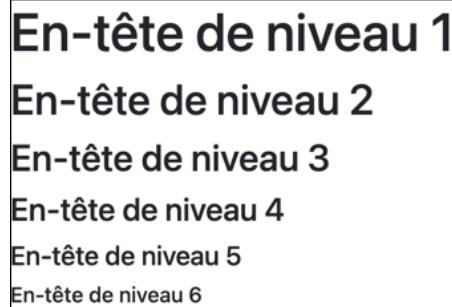
```
h1 {  
  font-size: 2.5rem;  
}  
h2 {  
  font-size: 2rem;  
}  
h3 {  
  font-size: 1.75rem;  
}  
h4 {  
  font-size: 1.5rem;  
}
```

```
h5 {  
  font-size: 1.25rem;  
}  
h6 {  
  font-size: 1rem;  
}
```

Voici un exemple simple d'utilisation des en-têtes `<h1>` à `<h6>` :

```
<h1>En-tête de niveau 1</h1>  
<h2>En-tête de niveau 2</h2>  
<h3>En-tête de niveau 3</h3>  
<h4>En-tête de niveau 4</h4>  
<h5>En-tête de niveau 5</h5>  
<h6>En-tête de niveau 6</h6>
```

Voici l'affichage obtenu :



En-tête de niveau 1
En-tête de niveau 2
En-tête de niveau 3
En-tête de niveau 4
En-tête de niveau 5
En-tête de niveau 6

3.2 Les sélecteurs de classes des en-têtes

Bootstrap propose des classes nommées `.h1`, `.h2`, `.h3`, `.h4`, `.h5` et `.h6`, qui utilisent les mêmes propriétés que les en-têtes vus précédemment.

Ces classes peuvent s'utiliser dans des paragraphes par exemple :

```
<p class="h1">Paragraphe avec la classe <code>.h1</code></p>  
<p class="h2">Paragraphe avec la classe <code>.h2</code></p>  
<p class="h3">Paragraphe avec la classe <code>.h3</code></p>  
<p class="h4">Paragraphe avec la classe <code>.h4</code></p>  
<p class="h5">Paragraphe avec la classe <code>.h5</code></p>  
<p class="h6">Paragraphe avec la classe <code>.h6</code></p>
```

Voici l'affichage obtenu :

```
Paragraphe avec la classe .h1
Paragraphe avec la classe .h2
Paragraphe avec la classe .h3
Paragraphe avec la classe .h4
Paragraphe avec la classe .h5
Paragraphe avec la classe .h6
```

3.3 L'affichage de grands en-têtes

Bootstrap propose des classes spéciales pour afficher les en-têtes avec une grande taille de caractères. Dans les éléments `<h1>` à `<h6>`, il faut ajouter les classes `.display-1` à `.display-6`.

Voici le code de l'exemple :

```
<h1 class="display-1">Affichage de niveau 1</h1>
<h2 class="display-2">Affichage de niveau 2</h2>
<h3 class="display-3">Affichage de niveau 3</h3>
<h4 class="display-4">Affichage de niveau 4</h4>
```

Voici l'affichage obtenu :

```
Affichage de niveau 1
Affichage de niveau 2
Affichage de niveau 3
Affichage de niveau 4
```

4. Les paragraphes

Bootstrap propose une mise en forme spéciale pour mettre en évidence certains paragraphes avec la classe `.lead`.

■ Remarque

*L'exemple à télécharger est dans le dossier **04-Typographie-02**.*

Voici le code de l'exemple :

```
<p>Etiam porta sem malesuada...</p>
<p class="lead">Etiam porta sem malesuada...</p>
<p>Etiam porta sem malesuada...</p>
```

Voici l'affichage obtenu :

```
Etiam porta sem malesuada magna mollis euismod. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
Etiam porta sem malesuada magna mollis euismod. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
Etiam porta sem malesuada magna mollis euismod. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
```

Les paragraphes `<p>` utilisent les propriétés de **Reboot** évoquées précédemment :

```
p {
  margin-top: 0;
  margin-bottom: 1rem;
}
```

Dans notre exemple, le deuxième paragraphe utilise la classe `.lead`, dont voici les propriétés :

```
.lead {
  font-size: 1.25rem;
  font-weight: 300;
}
```

5. Les mises en évidence

Bootstrap propose des mises en forme spécifiques pour les mises en évidence de textes dans des paragraphes.

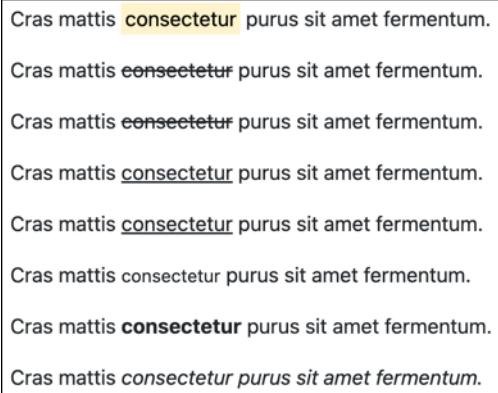
■ Remarque

L'exemple à télécharger est dans le dossier **04-Typographie-03**.

Voici le code de l'exemple :

```
<p>Cras mattis <mark>consectetur</mark> purus sit amet fermentum.</p>
<p>Cras mattis <del>consectetur</del> purus sit amet fermentum.</p>
<p>Cras mattis <s>consectetur</s> purus sit amet fermentum.</p>
<p>Cras mattis <ins>consectetur</ins> purus sit amet fermentum.</p>
<p>Cras mattis <u>consectetur</u> purus sit amet fermentum.</p>
<p>Cras mattis <small>consectetur</small> purus sit amet fermentum.</p>
<p>Cras mattis <strong>consectetur</strong> purus sit amet fermentum.</p>
<p>Cras mattis <em>consectetur</em> purus sit amet fermentum.</p>
```

Voici l'affichage obtenu :



The screenshot shows a list of text snippets, each with a different Bootstrap styling class applied to a word or phrase. The classes are: **mark** (highlighted in yellow), **del** (strikethrough), **s** (text-decoration: underline), **ins** (text-decoration: underline), **u** (text-decoration: underline), **small** (smaller font), **strong** (bold), and **em** (italic). The text content is "Cras mattis consectetur purus sit amet fermentum." repeated eight times.

Cras mattis **consectetur** purus sit amet fermentum.
Cras mattis ~~consectetur~~ purus sit amet fermentum.
Cras mattis ~~consectetur~~ purus sit amet fermentum.
Cras mattis consectetur purus sit amet fermentum.
Cras mattis consectetur purus sit amet fermentum.
Cras mattis consectetur purus sit amet fermentum.
Cras mattis **consectetur** purus sit amet fermentum.
Cras mattis *consectetur* purus sit amet fermentum.

Servez des outils d'analyse de code de vos navigateurs pour lister toutes les propriétés et les valeurs utilisées dans ces classes.