

## Appliquer des styles grâce au CSS

---

Maintenant que vous connaissez les bases du HTML, vous pouvez utiliser différentes balises pour formater le texte d'une page Web. Vous pouvez y insérer des liens et des images.

Toutefois, la plupart des effets cosmétiques (appelés effets de styles) sont appliqués à l'aide du CSS. CSS signifie « Cascading Style Sheets », ou feuilles de style en cascade. Bien rares sont les sites Web bâtis exclusivement en HTML; presque tous utilisent maintenant le CSS.

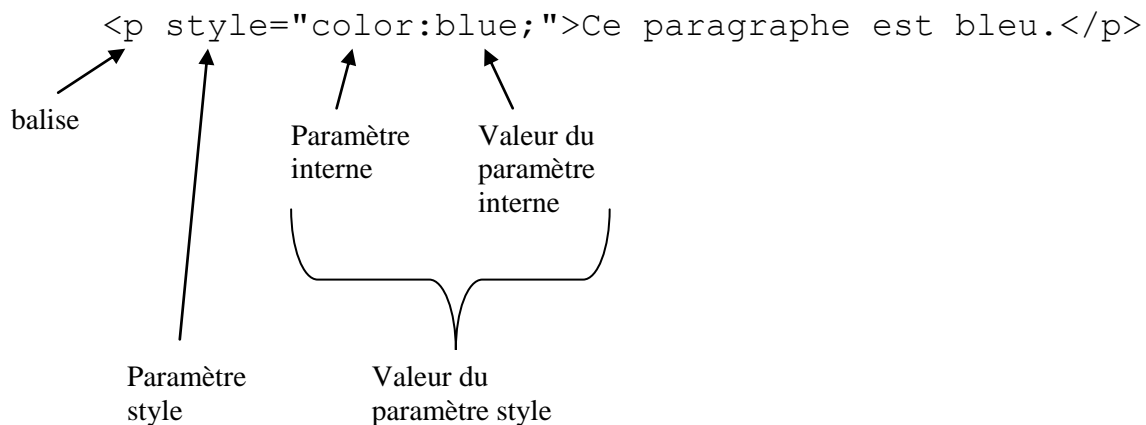
Une feuille de style est une page qui contient la définition de plusieurs styles. Les pages HTML de tout un site peuvent ensuite y faire référence. De cette façon, en modifiant les styles d'une feuille CSS, on modifie d'un seul coup l'apparence de plusieurs pages Web, sans toucher au HTML de ces dernières.

L'aspect « en cascade » signifie qu'une feuille de style peut hériter des propriétés d'une autre feuille de style et en ajouter. Nous n'exploiterons toutefois pas cette possibilité avancée dans le cadre de ce cours.

### *Le CSS directement dans le corps du HTML (et la balise <div>)*

On peut utiliser le paramètre « style » dans la très grande majorité des balises. Ce paramètre, comme tous les autres, doit être écrit dans la balise, après le nom de celle-ci, mais à l'intérieur des <>. Elle sera suivie, comme tout paramètre, d'un symbole = et d'une valeur entre guillemets. Par contre, sa valeur sera en fait composée d'un ou plusieurs « paramètres internes », avec un nom, un « : » et une valeur, séparés par des « ; ».

Ouf! C'est assez compliqué à suivre, mais tout va s'éclaircir avec un exemple :



Le paramètre « style » peut être utilisé dans plusieurs balises différentes. Par exemple, ici, la même chose avec la balise <a> (lien) :

```
<a href="Page2.html" style="color:blue;">Allez à la page 2</a>
```

On peut énumérer plusieurs éléments de style entre les guillemets. Il suffit de les séparer par des points-virgules (il y en avait d'ailleurs un après le mot « blue » dans les exemples précédents pour vous préparer à cette éventualité!) Par exemple :

```
<p style="color:blue; border-width:1px; border-color: black;
border-style: solid; background-color:yellow; font-size:150%">Ce
paragraphe a beaucoup de style</p>
```

On combine ici plusieurs éléments de style dans la même balise :

- color : couleur du texte (bleu)
- border-width : épaisseur de la bordure autour du paragraphe (1 px = 1 pixel)
- border-color : couleur de la bordure (noir)
- border-style : style de la bordure (solid = lignes pleines)
- background-color : couleur de fond (jaune)
- font-size : taille de la police (150% de plus que son élément parent, ici 150% de plus que la taille par défaut)

Il existe des tas d'autres propriétés de style que l'on peut modifier. Voici un site qui les énumère toutes – il est malheureusement en anglais, mais est très clair et donne des exemples pour chacune d'elles : <http://www.w3schools.com/cssref/default.asp>

Ceci nous permet de modifier le style de tout élément de notre page avec une grande précision. On utilisera souvent une balise <div> strictement pour appliquer un style. La balise <div> ne sert qu'à diviser la page en morceaux et ne provoque aucun effet visible lorsqu'utilisée seule. On peut toutefois y mettre un paramètre style :

```
<div style="color:blue; border-width:1px; border-color: black;
border-style: solid; background-color:yellow; font-size:150%">Ce
paragraphe a beaucoup de style</div>
```

C'est à peu près équivalent à utiliser <p>, à la différence qu'on peut mettre un <div> à l'intérieur d'un <div> (on ne le fera pas dans ce cours, mais soyez avertis que c'est d'usage courant).

### *Le CSS au début de la page*

Appliquer le style à une balise est toutefois lourd et difficile à modifier. En effet, si on veut plusieurs paragraphes avec le même style un peu partout dans la page, il faudra réécrire tout le texte du paramètre style chaque fois.

C'est pourquoi on va souvent définir les styles une seule fois au début de la page :

```
<head>
```

```
<style>
body {background-color:lightgrey;}
p    {color:blue;}
a    {color:green;}
</style>
```

```
</head>
```

```

<body>

<p>Ceci est un paragraphe</p>
<a href="http://google.ca">Ceci est un lien</a>

</body>

```

Ici, la balise `<style>` est utilisée. Elle est refermée plus loin avec la balise `</style>` et est placée à l'intérieur de la section `<head>`.

Entre `<style>` et `</style>`, on décidera quels styles appliquer à toutes les balises choisies, tout au long de la page. Par exemple, ici, on définit une couleur de fond à utiliser pour la balise `body`, une couleur de texte à utiliser pour toutes les balises `<p>` et une autre couleur pour toutes les balises `<a>`. Tout ceci sera valide pour toute la page.

### *Des classes pour des styles différents*

Mais que faire si on veut définir deux styles de paragraphes différents et alterner d'un à l'autre tout au long du document? Si on le définit une fois dans une balise `<style>`, il s'appliquera à tous les paragraphes. Doit-on alors le définir chaque fois dans la balise `<p>`?

Eh non! On peut définir des **classes** pour chaque type de balise. Ceci se fera toujours dans la balise `<style>`, elle-même dans la balise `<head>`. Bref, on fera comme dans l'exemple précédent, mais cette fois, on nommera les classes avec le nom de la balise, suivi d'un point, suivi du nom de la classe. Par exemple :

```

<head>

<style>
body      {background-color:lightgrey;}
p         {color:blue;}
p.erreur {color:white;background-color:red}
a         {color:green;}
</style>

</head>

```

On a défini un style pour les `<p>` (texte en bleu) et on a défini une classe spéciale de `p`, qui s'appelle « erreur », pour laquelle le style est « texte en blanc sur fond rouge ». Ensuite, dans le corps de notre page, on utilisera le paramètre « class » dans certaines balises `<p>` pour leur appliquer le style de la classe « erreur » :

```

<body>

<p>Ceci est un paragraphe</p>
<a href="http://google.ca">Ceci est un lien</a>
<p class="erreur">Ceci est une erreur</p>
<p>Ceci est un autre paragraphe</p>
<p class="erreur">Ceci est une autre erreur</p>

</body>

```

Il faut toutefois faire attention : les `<p class="erreur">` recevront à la fois les éléments de style des `<p>` et ceux des `p.erreur`! Lorsque ce sont les mêmes éléments (comme c'est le cas ici, `color` est `blue` pour les `<p>` et `white` pour les `p.erreur`), le `p.erreur` l'emporte. Mais s'ils comportent des éléments différents, les deux s'appliquent.

Vous aurez remarqué qu'on a défini le style en faisant `p.erreur`, pour « classe erreur d'une balise `<p>` ». Il est aussi possible de définir un style pouvant s'appliquer à n'importe quelle balise en faisant quelque chose comme :

```
<style>
.erreur {color:white;background-color:red}
</style>
```

On garde le point, mais on ne met rien avant.

### *Les styles dans des fichiers à part*

Il est très courant de définir tous les styles dans un fichier à part et d'y faire référence dans plusieurs fichiers HTML. Ceci permet d'éviter de copier la balise `<style>` et tout son contenu au début de chaque fichier et ça simplifie aussi grandement la maintenance : on n'a qu'à modifier le contenu du fichier de styles pour changer l'apparence d'un tas de pages d'un seul coup!

Le fichier de styles ne doit pas contenir de code HTML, uniquement des définitions de styles. En fait, on n'y mettra que ce que l'on mettrait à l'intérieur d'une balise `<style></style>`, mais sans y inclure la balise elle-même. On nommera le fichier comme on veut, mais avec l'extension « `.css` » (et non pas « `.html` »!) Pour que tout fonctionne bien, le fichier `.css` doit être dans le même dossier que les fichiers `.html` qui l'utilisent.

Par exemple, on pourrait créer un fichier appelé `messtyles.css` et y placer le contenu suivant :

```
body      {background-color:lightgrey;}
p         {color:blue;}
.erreur   {color:white;background-color:red}
a         {color:green;}
```

(C'est la même chose que ce qu'on avait mis dans la balise `<style>` tout à l'heure.)

Ensuite, dans notre fichier `.html`, on enlèvera la balise `<style>` et tout son contenu, qui n'est plus nécessaire, et on y mettra plutôt une référence à notre fichier `.css` :

```
<head>
<link rel="stylesheet" href="messtyles.css">
</head>

<body>

<p>Ceci est un paragraphe</p>
<a href="http://google.ca">Ceci est un lien</a>
```

```

<p class="erreur">Ceci est une erreur</p>
<p>Ceci est un autre paragraphe</p>
<p class="erreur">Ceci est une autre erreur</p>

</body>

```

(La section <body> est exactement pareille comme tantôt!)

On utilisera donc la balise <link> pour faire référence à notre fichier de styles. La balise <link> n'a pas besoin d'être fermée et doit se trouver dans la section <head>. Elle a besoin au minimum de deux paramètres : rel (la relation entre le document courant et celui auquel on fait référence, ici « stylesheet » puisque c'est une feuille de styles), et href (le chemin vers le fichier, exactement comme dans une balise <a>).

Essayez de modifier le contenu du fichier .css, puis rechargez votre page .html dans votre navigateur. Vous devriez voir les changements s'appliquer!

## **Un atelier**

---

Vous pouvez vous placer en équipes de deux pour réaliser cet atelier.

Vous devez créer un fichier .css contenant différents styles et un fichier .html qui les utilise.

Vous aurez besoin de définir quatre styles :

- Un style s'appliquant au corps de la page :
  - o Couleur de fond beige.
- Un style s'appliquant aux liens :
  - o Le texte en orange
  - o Le texte en gras (on parle ici de « poids de la police », donc vérifiez de ce côté-là dans les propriétés)
- Un style s'appliquant à certains paragraphes :
  - o Le texte en jaune sur fond bleu
  - o Le texte semi-transparent (opacité à 50% ou à 0.5)
- Un style s'appliquant à certains autres paragraphes :
  - o Le texte en rouge sur fond rose
  - o Le texte complètement opaque (donc normal, pas transparent)
  - o Le texte avec une bordure en pointillés noirs de 2 pixels d'épaisseur

Dans votre document .html, alternez des paragraphes d'un style puis de l'autre au moins deux fois (A-B-A-B), et placez un lien vers une page Web de votre choix quelque part.

Remettez vos deux fichiers (.css et .html) dans la boîte de remise prévue à cet effet. N'oubliez pas de nommer vos fichiers avec les noms des deux équipiers si vous n'êtes pas seul à faire le travail. Un seul membre de l'équipe doit les remettre.