

Intégrez le CSS dans la page HTML

Liez le fichier CSS au fichier HTML

À partir du moment où vous créez un fichier `.css` pour appliquer du style à votre page web (écrite dans un fichier `.html`), il vous faut lier ces deux fichiers. Ainsi, les propriétés CSS que vous ajoutez vont pouvoir s'appliquer aux balises HTML auxquelles vous souhaitez qu'elles s'appliquent.

Pour lier les fichiers `.css` et `.html`, vous allez rajouter une ligne dans le fichier `.html` pour indiquer au navigateur d'aller chercher la **feuille de style** (*stylesheet* en anglais) afin d'afficher la page web avec les propriétés de style qu'on lui a appliquées.

Cette ligne à rajouter dans le fichier `.html` s'ouvre avec la balise orpheline `<link>` et on la place à l'intérieur de la balise `<head> </head>` :

```
<head>
<meta charset="utf-8">
<title>Ma page</title>
<link href="style.css" rel="stylesheet">
</head>
```

Lorsque vous cliquerez sur votre fichier HTML pour l'ouvrir dans le navigateur, ce dernier aura l'instruction d'appeler le fichier CSS associé pour afficher les propriétés de style de la page web.

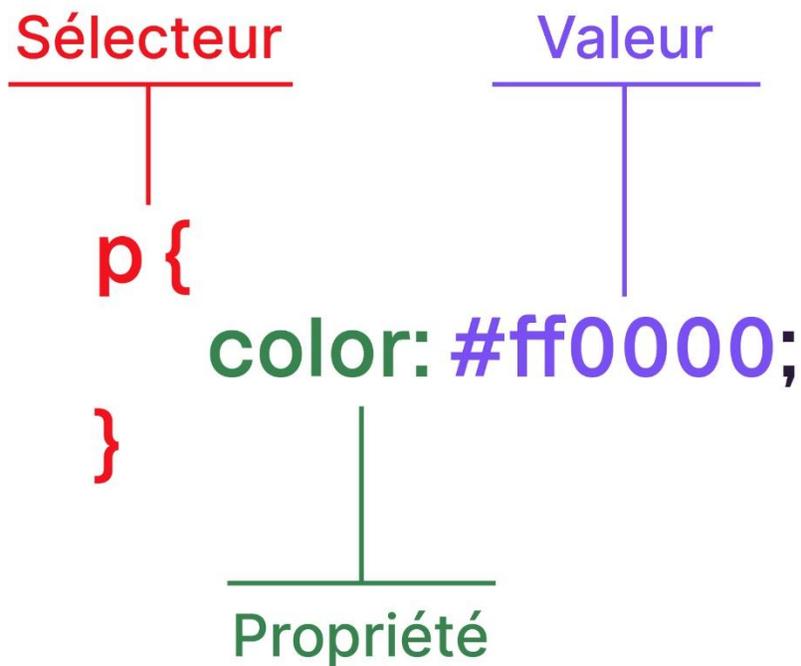
Appliquez une propriété CSS à une balise HTML

Dans la vidéo d'introduction, nous avons utilisé cette syntaxe dans notre fichier CSS :

```
h1 {
  color: blue;
}
```

Dans un code CSS comme celui-ci, on trouve en fait trois éléments différents :

1. **Le sélecteur** : ici on écrit les noms des balises HTML dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes `<p>` , je dois écrire `p` (sans les chevrons).
2. **La ou les propriétés CSS** : les effets de style sont listés via des propriétés. Par exemple, `color` permet d'indiquer la couleur du texte, `font-size` permet d'indiquer la taille du texte, etc. Il existe BEAUCOUP de propriétés CSS ! Mais rassurez-vous, vous n'avez pas besoin de les connaître toutes par cœur.
3. **...et leurs valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété `color` , il faut indiquer le nom de la couleur. Pour `font-size`, il faut indiquer quelle taille on veut, etc.



On écrit le nom de la balise. Puis entre accolades `{ }` on écrit les propriétés et leurs valeurs. (On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades).

Chaque propriété est suivie du symbole `:` puis de la valeur correspondante.

Chaque ligne se termine par `;`.

Voici un exemple ! Pour changer la couleur des paragraphes, on écrirait :

```
p {
  color: blue;
```

```
}
```

Ce code signifie en français :

“Je veux que tous les paragraphes soient écrits en bleu.”

Au fait on peut faire des commentaires en CSS, comme on en fait en HTML ?

Oui ! Tapez /* suivi de votre commentaire, puis */ :

```
p {  
  
  /* J'en profite pour  
  ajouter un commentaire  
  sur plusieurs lignes */  
  
  color: blue;  
}
```

Appliquez une propriété CSS à plusieurs balises HTML à la fois

Ce code CSS signifie que nos titres de niveau 1 et nos paragraphes doivent s'afficher en bleu :

```
h1 {  
  
  color: blue;  
}  
  
p {  
  
  color: blue;  
}
```

Ok mais c'est un peu répétitif, non ? On va devoir tout écrire à la main comme ça ?

Heureusement, non. Si les deux balises ont la même style, il suffit de combiner la déclaration en séparant les noms des balises par une virgule :

```
h1, p  
{
```

```
color: blue;
```

Et voilà ce que ça donne :

Titre principal

Ceci est le contenu de mon premier paragraphe

Ceci est le contenu de mon deuxième paragraphe

Voilà mon sous-titre h2

Les éléments h1 et p sont en bleu mais le sous-titre h2 reste en noir

Ce que l'on vient de voir implique que TOUS les paragraphes seront écrits en bleu.

Comment faire pour que certains paragraphes soient écrits d'une couleur différente ?

Pour appliquer un style à un seul élément (par exemple à un seul paragraphe parmi tous les paragraphes de notre code), on peut - **en théorie** - utiliser deux attributs :

1. L'attribut `class` ;
2. Ou l'attribut `id`.

Mais pour ce faire - **en pratique** - on va surtout utiliser l'attribut `class`.

Les attributs `class` et `id` fonctionnent selon la même méthode mais on ne les utilise pas pour les mêmes raisons :

- pour appliquer un style à *un seul élément parmi d'autres*, on utilise un attribut `class`.
- pour appliquer un style à *un élément unique en son genre*, on utilise un attribut `id`.

En CSS, on peut appliquer du style à un élément (ou plus) avec l'attribut `class`. Par contre `id` ne peut s'utiliser que pour un seul élément, pas plus.

Voyons comment appliquer la méthode dans un exemple dont l'usage est pertinent avec l'attribut `class`

Appliquez un style à un élément isolé avec l'attribut `class`

Étape 1 : marquez un élément avec l'attribut `class` dans le fichier `.html`

`class` est un attribut que l'on peut mettre sur n'importe quelle balise HTML :

```
<h1 class="voici"> </h1>

<p class="des"> </p>

<img class="classes">
```

Oui, mais que met-on comme valeur à l'attribut `class`?

En fait, vous devez écrire un nom qui sert à identifier la balise. Ce que vous voulez, du moment que le nom commence par une lettre. Souvenez-vous qu'il s'agit d'un moyen de **sélectionner** votre élément pour ensuite lui appliquer du style.

Par exemple, je vais associer `ma-classe` à mon deuxième paragraphe dans le HTML :

```
<body>

  <h1>Titre principal</h1>

  <p>Ceci est le contenu de mon premier paragraphe</p>

  <p class="ma-classe">Ceci est le contenu de mon deuxième paragraphe</p>

  <h2>Voilà mon sous-titre h2</h2>

</body>
```

On vient de créer un attribut `class` nommé `ma-classe` dans le fichier `.html` pour marquer un élément auquel on veut appliquer un style en particulier. Maintenant, il faut appeler l'élément que l'on a marqué dans le fichier `.css` pour pouvoir lui appliquer un style.

Étape 2 : appelez le nom de l'attribut `class` via un `.` dans le fichier `.css`

Pour faire cela, on va indiquer dans le fichier `.css` le nom de notre classe précédée d'un point `.` :

```
.ma-classe {  
  
  color: #663399;  
  
}
```

D'ailleurs, on en profite pour utiliser la notation hexadécimale pour indiquer la couleur, ce qui nous permet d'être plus précis (pas de panique, on verra cela plus en détail dans le prochain chapitre).

Testez le résultat : seul votre paragraphe appelé `ma-classe` va prendre la couleur qu'on lui a attribué dans le fichier `.css` :

Titre principal

Ceci est le contenu de mon premier paragraphe

Ceci est le contenu de mon deuxième paragraphe

Voilà mon sous-titre h2

La classe "ma-classe" associée à la couleur violette pour le deuxième paragraphe

Et comment on fait ça avec l'attribut `id` alors ? Et en quoi c'est différent ?

Appliquez une propriété CSS à un élément unique avec l'attribut `id`

L'attribut `id` fonctionne selon la même méthode que `class`, mais il y a une différence de taille : `id` ne peut être utilisé **qu'une fois** dans le code.

Quel intérêt alors ?

En CSS, il n'y en a pas vraiment ; cela vous sera surtout utile si vous faites du [JavaScript](#) plus tard, pour reconnaître certaines balises.

En HTML, il y en a un car souvenez-vous on l'avait utilisé dans le chapitre sur le liens hypertextes pour créer des ancres.

En pratique donc, on n'utilisera en CSS un `id` que sur un élément qui est unique dans la page, comme le logo par exemple.

Étape 1 : marquez un élément unique avec l'attribut `id` dans le fichier `.html`

Exemple avec le logo, élément unique, que l'on va "marquer" grâce à l'attribut `id` dans le fichier `.html`:

```

```

Étape 2 : appelez l'attribut `id` via un `#` dans le fichier `.css`

Avec l'attribut `id`, on définit les propriétés de style de cet élément unique dans le fichier `.css` en l'appelant par un dièse `#`:

```
#logo {  
  
/* Indiquez les propriétés CSS ici */  
  
}
```

Appliquez plusieurs propriétés CSS d'un coup à un élément

En CSS, il est tout à fait possible de cumuler les styles.

Pour l'occasion, j'en profite pour vous montrer une nouvelle propriété `font-size`, qui permet d'indiquer la taille d'un texte (que nous verrons plus en détail dans le chapitre suivant).

Si on veut par exemple que tous nos textes soient de la taille `30px`, on peut :

- déclarer le style :

```
.ma-classe {  
  
color: #663399;  
  
}  
  
.grand-texte {  
  
font-size: 30px;  
  
}
```

- et l'utiliser sur `p` (mais aussi sur vos autres éléments ; il est possible de cumuler plusieurs classes au sein d'une même balise). Par exemple :

```
<body>
  <h1>Titre principal</h1>
  <p>Ceci est le contenu de mon premier paragraphe</p>
  <p class="ma-classe grand-texte">Ceci est le contenu de mon deuxième paragraphe</p>
  <h2 class="grand-texte">Voilà mon sous-titre h2</h2>
</body>
```

Ce qui donne :

Titre principal

Ceci est le contenu de mon premier paragraphe

Ceci est le contenu de mon deuxième paragraphe

Voilà mon sous-titre h2

De la même manière, on peut ajouter plusieurs propriétés sur un même sélecteur :

```
balise1
{
  propriete1: valeur1;
  propriete2: valeur2;
  /* ... */
}
```

Ok super ! Et comment je fais si je veux appliquer un style à quelque chose qui n'est pas entouré par une balise, par exemple un mot ou un groupe de mots qu'il y a dans une phrase ?

Bonne question ! Si vous avez besoin de donner un attribut à certains mots pour leur donner un style mais que ces derniers ne sont pas entourés par des balises (un attribut s'applique à une balise en entier, pas juste une partie), alors vous allez utiliser des balises universelles !

Exploitez les balises universelles

Imaginons, je veux modifier uniquement 990 espèces d'oiseaux dans le paragraphe suivant :

```
<p>Il existe plus de 990 espèces d'oiseaux répertoriées dans toute l'Europe.</p>
```

Cela serait facile à faire s'il y avait une balise autour de "990 espèces d'oiseaux", mais malheureusement, il n'y en a pas. Par chance, on a inventé... la balise-qui-ne-sert-à-rien.

En fait, on a inventé deux balises dites "universelles", qui n'ont aucune signification particulière (elles n'indiquent pas que le mot est important, par exemple). Il y a une petite différence (mais significative) entre ces deux balises :

1. ` `;

2. `<div> </div>`.

```
<span> </span>
```

C'est une balise de type "**inline**", c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte pour sélectionner certains mots uniquement. Les balises `` et `` sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe, et c'est celle dont nous allons nous servir pour colorer "990 espèces d'oiseaux".

```
<div> </div>
```

C'est une balise de type "**block**", qui entoure un bloc de texte. Les balises `<p>`, `<h1>` , etc., sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau "bloc", dans la page, et provoquent donc obligatoirement un retour à la ligne. `<div>` est une balise fréquemment utilisée dans la construction d'une mise en page, comme nous le verrons plus tard.

Pour le moment donc, nous allons plutôt utiliser la balise ``.

En résumé

- CSS est un autre langage qui vient compléter le HTML. Son rôle est de mettre en forme votre page web.

- Pour écrire le code CSS, on crée un fichier séparé portant l'extension `.css` comme `style.css`.
- Pour lier les fichiers CSS et HTML, on rajoute une ligne dans la balise `<head> </head>` du fichier HTML : `<link href="style.css" rel="stylesheet">`
- En CSS, on sélectionne les portions de la page HTML qu'on veut modifier, et on change leur présentation avec des propriétés CSS :

```

balise1

{

    propriete1: valeur1;

    propriete2: valeur2;

}

```

- Il existe plusieurs façons de sélectionner la portion de page que l'on veut mettre en forme. Par exemple, on peut viser :
 - toutes les balises d'un même type, en écrivant simplement leur nom (`h1` par exemple) ;
 - certaines balises spécifiques, auxquelles on a donné des noms à l'aide des attributs `class` ou `id` (`.nom-classe` ou `#nom-id`) ;
 - uniquement les balises qui se trouvent à l'intérieur d'autres balises (`h3,em`).