

AP1 : prise en main d'un site WEB existant, ajout de contenu et développement personnel

B1.3 Développer la présence en ligne

B1.6 Organiser son développement personnel

Objet du document : Apprentissage des éléments principaux de HTML 5 et CSS 3 afin de pouvoir réaliser un site web

Prérequis : Editeur de texte Notepad++ ou VS code Navigateur Firefox et Chrome

Chapitre 8 : Mettez en place le CSS

Petit rappel : à quoi sert CSS ?

CSS ? C'est lui qui vous permet de choisir la couleur de votre texte.

Lui qui vous permet de sélectionner la police utilisée sur votre site.

Lui encore qui permet de définir la taille du texte, les bordures, le fond...

Et aussi, c'est lui qui permet de faire la mise en page de votre site. Vous pourrez dire : je veux que mon menu soit à gauche et occupe telle largeur, que l'en-tête de mon site soit calé en haut et qu'il soit toujours visible, etc.

Souvenez-vous de ce petit comparatif que nous avons vu dès le premier chapitre (figure suivante).

HTML
(pas de CSS)



HTML + CSS



La même page HTML, sans et avec CSS (www.csszengarden.com)

Grâce au HTML, nous avons pu rédiger le contenu de notre site, mais il est brut de décoffrage. Le CSS vient compléter ce code pour mettre en forme tout cela et donner au contenu l'apparence que l'on souhaite.

Où écrit-on le CSS ?

Vous avez le choix car on peut écrire du code en langage CSS à trois endroits différents :

- dans un fichier `.css` (*méthode la plus recommandée*) ;
- dans l'en-tête `<head>` du fichier HTML ;
- directement dans les balises du fichier HTML *via* un attribut `style` (*méthode la moins recommandée*).

Je vous cite les trois méthodes, mais sachez d'ores et déjà que la première... est la meilleure et la seule que vous devez utiliser !

Dans un fichier `.css` (recommandé)

Comme je viens de vous le dire, on écrit le plus souvent le code CSS dans un fichier spécial ayant l'extension `.css` (contrairement aux fichiers HTML qui ont l'extension `.html`). C'est la méthode la plus pratique et la plus souple. Cela nous évite de tout mélanger dans un même fichier. J'utiliserai cette technique dans toute la suite de ce cours.

Commençons à pratiquer dès maintenant ! Nous allons partir du fichier HTML suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" />
    <title>Premiers tests du CSS</title>
  </head>

  <body>
    <h1>Mon super site</h1>

    <p>Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez
encore un peu !</p>
  </body>
</html>
```

Vous noterez le contenu de la ligne 5, `<link rel="stylesheet" href="css/style.css" />` : c'est elle qui indique que ce fichier HTML est associé à un fichier appelé `style.css` et chargé de la mise en forme.

Attention : un nom de fichier doit être écrit en minuscule, sans espace ni caractères spéciaux !

Ce fichier est rangé dans le répertoire `<css>` ! Ce répertoire contiendra toutes vos feuilles de style !

Enregistrez ce fichier sous le nom que vous voulez (par exemple, `page.html`). Pour le moment, rien d'extraordinaire à part la nouvelle balise que nous avons ajoutée.

Maintenant, créez un *nouveau* fichier vide dans votre éditeur de texte (par exemple Sublime Text) et copiez-y ce bout de code CSS (rassurez-vous, je vous expliquerai tout à l'heure ce qu'il veut dire) :

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

Pour obtenir la coloration du code dans Sublime Text, enregistrez bien votre fichier avec l'extension `.css` d'abord.

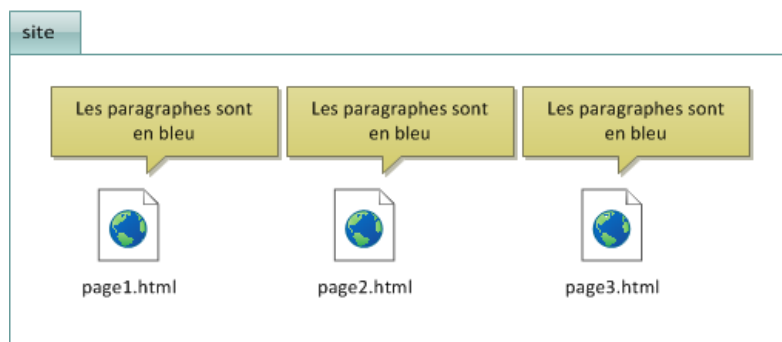
Enregistrez le fichier en lui donnant un nom qui se termine par `.css` , comme `style.css` . Placez ce fichier `.css` dans le même dossier que votre fichier `.html` .

Dans votre explorateur de fichiers, vous devriez les voir apparaître côte à côte. D'un côté le `.html` , de l'autre le `.css`

Ouvrez maintenant votre fichier `page.html` dans votre navigateur pour le tester, comme vous le faites d'habitude. Regardez, c'est magique : vos paragraphes sont écrits en bleu

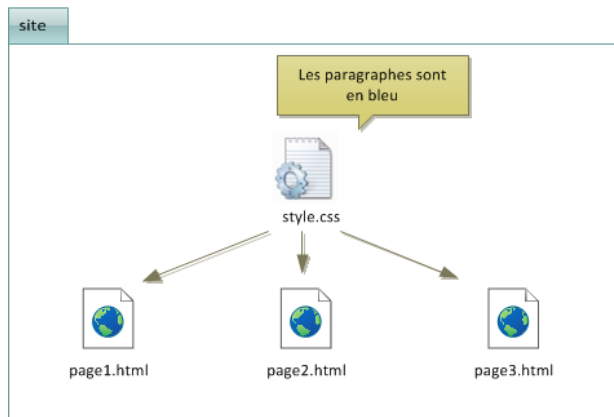
Pour le moment, vous faites vos tests sur un seul fichier HTML. Cependant, votre site sera plus tard constitué de plusieurs pages HTML, on est d'accord ?

Imaginez : si vous placez le code CSS directement dans le fichier HTML, il faudra copier ce code dans tous les fichiers HTML de votre site ! Et si demain vous changez d'avis, par exemple si vous voulez que vos paragraphes soient écrits en rouge et non en bleu, il faudra modifier chaque fichier HTML un à un, comme le montre la figure suivante.



Le code CSS est répété dans chaque fichier HTML

Si vous travaillez avec un fichier CSS externe, vous n'aurez besoin d'écrire cette instruction qu'une seule fois pour tout votre site, comme le montre la figure suivante.



Le code CSS est donné une fois pour toutes dans un fichier CSS

Appliquer un style : sélectionner une balise

Maintenant que nous savons où placer le code CSS, intéressons-nous de plus près à ce code. Je vous ai donné, sans vous l'expliquer, un premier bout de code CSS :

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

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

- **des noms de balises** : on écrit les noms des balises dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes `<p>` , je dois écrire `p` ;
- **des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Il y a par exemple la propriété `color` qui permet d'indiquer la couleur du texte, `font-size` qui permet d'indiquer la taille du texte, etc. Il y a beaucoup de propriétés CSS et, comme je vous l'ai dit, je ne vous obligerai pas à les connaître toutes par cœur ;
- **les 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.

Schématiquement, une feuille de style CSS ressemble donc à cela :

```
balise1
{
    propriete1: valeur1;
    propriete2: valeur2;
    propriete3: valeur3;
}

balise2
{
    propriete1: valeur1;
    propriete2: valeur2;
    propriete3: valeur3;
    propriete4: valeur4;
}
```

```
balise3
{
    propriete1: valeur1;
}
```

Vous repérez dans cet extrait de code les balises, propriétés et valeurs dont je viens de vous parler.

Comme vous le voyez, on écrit le nom de la balise (par exemple `h1`) et on ouvre des accolades pour, à l'intérieur, mettre les propriétés et valeurs que l'on souhaite. On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades. Chaque propriété est suivie du symbole « deux-points » (:) puis de la valeur correspondante. Enfin, chaque ligne se termine par un point-virgule (;).

Je vous apprendrai de nombreuses propriétés dans les chapitres suivants. Pour le moment, dans les exemples, on va juste changer la couleur pour s'entraîner.

Le code CSS que nous avons utilisé jusqu'ici :

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

... signifie donc en français : « *Je veux que tous mes paragraphes soient écrits en bleu.* ».

Essayez de changer le nom de la balise affectée par le code CSS. Par exemple, si j'écris `h1` , c'est le titre qui sera écrit en bleu. Modifiez votre fichier `style.css` comme ceci :

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

Maintenant, ouvrez à nouveau votre page HTML (souvenez-vous, c'est la page HTML qu'on ouvre dans le navigateur, pas le fichier CSS !) : vous devriez voir son titre s'afficher en bleu

Appliquer un style à plusieurs balises

Prenons le code CSS suivant :

```
h1
{
    color: blue;
}

em
{
    color: blue;
}
```

Il signifie que nos titres `<h1>` et nos textes importants `` doivent s'afficher en bleu. Par contre, c'est un peu répétitif, vous ne trouvez pas ?

Heureusement, il existe un moyen en CSS d'aller plus vite si les deux balises doivent avoir la même présentation. Il suffit de combiner la déclaration en séparant les noms des balises par une virgule, comme ceci :

```
h1, em
```

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

Cela signifie : « *Je veux que le texte de mes `<h1>` et `` soit écrit en bleu* ».

Vous pouvez indiquer autant de balises à la suite que vous le désirez.

Des commentaires dans du CSS

Comme en HTML, il est possible de mettre des commentaires. Les commentaires ne seront pas affichés, ils servent simplement à indiquer des informations pour vous, par exemple pour vous y retrouver dans un looong fichier CSS.

D'ailleurs, vous allez vous en rendre compte, en général le fichier HTML est assez court et la feuille CSS assez longue (si elle contient tous les éléments de style de votre site, c'est un peu normal). Notez qu'il est possible de créer plusieurs fichiers CSS pour votre site si vous ressentez le besoin de séparer un peu votre code CSS (en fonction des différentes sections de votre site, par exemple).

... De quoi on parlait déjà ? Ah oui, les commentaires en CSS.

Donc, pour faire un commentaire, c'est facile ! Tapez `/*` , suivi de votre commentaire, puis `*/` pour terminer votre commentaire.

Vos commentaires peuvent être écrits sur une ou plusieurs lignes. Par exemple :

```
/*
style.css
-----

Par nom de l'auteur
*/

p
{
    color: blue; /* Les paragraphes seront en bleu */
}
```

Il est possible que j'utilise les commentaires dans la suite du cours, pour vous donner des explications à l'intérieur même des fichiers `.css` .

Appliquer un style : class et id

Ce que je vous ai montré jusqu'ici a quand même un défaut : cela implique par exemple que TOUS les paragraphes possèdent la même présentation (ici, ils seront donc tous écrits en bleu).

Comment faire pour que certains paragraphes seulement soient écrits d'une manière différente ? On pourrait placer le code CSS dans un attribut `style` sur la balise que l'on vise (c'est la technique que je vous ai présentée un peu plus tôt) mais, comme je vous l'ai dit, ce n'est pas recommandé (il vaut mieux utiliser un fichier CSS externe).

Pour résoudre le problème, on peut utiliser ces attributs spéciaux *qui fonctionnent sur toutes les balises* :

- l'attribut `class` ;

- l'attribut `id` .

Que les choses soient claires dès le début : les attributs `class` et `id` sont quasiment identiques. Il y a seulement une petite différence que je vous dévoilerai plus bas. Pour le moment, et pour faire simple, on ne va s'intéresser qu'à l'attribut `class` .

Comme je viens de vous le dire, c'est un attribut que l'on peut mettre sur n'importe quelle balise, aussi bien titre que paragraphe, image, etc.

```
<h1 class=""> </h1>
<p class=""> </p>
<img class="" />
```

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.

Par exemple, je vais associer la classe `introduction` à mon premier paragraphe (ligne 12) :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" />
    <title>Premiers tests du CSS</title>
  </head>

  <body>
    <h1>Mon super site</h1>

    <p class="introduction">Bonjour et bienvenue sur mon site !</p>
    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez
encore un peu !</p>
  </body>
</html>
```

Maintenant que c'est fait, votre paragraphe est identifié. Il a un nom : `introduction` . Vous allez pouvoir réutiliser ce nom dans le fichier CSS pour dire : « *Je veux que seules les balises qui ont comme nom 'introduction' soient affichées en bleu* ».

Pour faire cela en CSS, indiquez le nom de votre classe en commençant par un point, comme ci-dessous :

```
.introduction
{
  color: blue;
}
```

Testez le résultat : seul votre paragraphe appelé `introduction` va s'afficher en bleu

Et l'attribut `id` , alors ?

Lui, il fonctionne exactement de la même manière que `class` , à un détail près : il ne peut être utilisé *qu'une fois* dans le code.

Quel intérêt ? Il y en a assez peu pour tout vous dire ; cela vous sera utile si vous faites du JavaScript plus tard, pour reconnaître certaines balises. D'ailleurs, nous avons déjà vu l'attribut `id` dans le chapitre sur les liens (pour réaliser des ancres). En pratique, nous ne mettrons des `id` que sur des éléments qui sont uniques dans la page, comme par exemple le logo :

```

```

Si vous utilisez des `id` , lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l' `id` par un dièse (#) :

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

Je ne vous propose pas de le tester, cela fonctionne exactement comme `class` .

Si vous vous emmêlez les pinceaux entre `class` et `id` , retenez que deux balises peuvent avoir le même nom avec l'attribut `class` . Un nom d' `id` doit en revanche être unique dans la page HTML.

Les balises universelles

Il arrivera parfois que vous ayez besoin d'appliquer une `class` (ou un `id`) à certains mots qui, à l'origine, ne sont pas entourés par des balises.

En effet, le problème de `class` , c'est qu'il s'agit d'un attribut. Vous ne pouvez donc en mettre que sur une balise. Si, par exemple, je veux modifier uniquement « bienvenue » dans le paragraphe suivant :

```
<p>Bonjour et bienvenue sur mon site !</p>
```

Cela serait facile à faire s'il y avait une balise autour de « bienvenue » , 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 différence minime (mais significative !) entre ces deux balises :

- `` `` : 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 « bienvenue » ;
- `<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'un design, comme nous le verrons plus tard.

Pour le moment donc, nous allons utiliser plutôt la balise `` . On la met autour de « bienvenue », on lui ajoute une classe (du nom qu'on veut), on crée le CSS et c'est gagné !

```
<p>Bonjour et <span class="salutations">bienvenue</span> sur mon site !</p>
.salutations
{
    color: blue;
}
```

Appliquer un style : les sélecteurs avancés

En CSS, le plus difficile est de savoir cibler le texte dont on veut changer la forme. Pour cibler (on dit « sélectionner ») les éléments de la page à modifier, on utilise ce qu'on appelle des **sélecteurs**. Vous en avez déjà utilisé quelques-uns un peu plus tôt dans ce chapitre, résumons-les pour commencer.

Les sélecteurs que vous connaissez déjà

Ces sélecteurs, que nous avons vus précédemment, sont de loin les plus couramment utilisés. Il faut les connaître par cœur. Commençons par la base de la base :

```
p
{
}
```

... signifie « Je veux toucher tous les paragraphes ». Après, c'est à vous de dire ce que vous faites à ces paragraphes (vous les écrivez en bleu, par exemple).

Nous avons aussi vu :

```
h1, em
{
}
```

... qui signifie « Tous les titres et tous les textes importants ». Nous avons sélectionné deux balises d'un coup.

Et enfin, nous avons vu comment sélectionner des balises précises à qui nous avons donné un nom, grâce aux attributs `class` et `id` :

```
.class
{
}

#id
{
}
```

Vous savez quoi ? Il existe des dizaines d'autres façons de cibler des balises en CSS ! Nous n'allons pas toutes les voir car il y en a beaucoup et certaines sont complexes, mais voici déjà de quoi vous permettre d'être plus efficace en CSS !

* : sélecteur universel

```
*  
{  
  
}
```

Sélectionne toutes les balises sans exception. On l'appelle le *sélecteur universel*.

A B : une balise contenue dans une autre

```
h3 em  
{  
  
}
```

Sélectionne toutes les balises `` situées à l'intérieur d'une balise `<h3>` . Notez qu'il n'y a pas de virgule entre les deux noms de balises.

Exemple de code HTML correspondant :

```
<h3>Titre avec <em>texte important</em></h3>
```

Sélectionne la première balise `<p>` située après un titre `<h3>` .

Je ne vous ai présenté ici qu'une partie des sélecteurs CSS, mais sachez qu'il en existe beaucoup d'autres. Si vous voulez une liste complète, vous pouvez vous renseigner directement à la source : sur le [site du W3C](#) ! C'est très complet.

Sachez que nous découvrirons certains de ces autres sélecteurs dans la suite de ce cours !

EN RÉSUMÉ : NOTEZ ICI LES POINTS IMPORTANTS DU CHAPITRE