

NSI Première (2019-2020)

HTML et CSS

1 Le langage HTML

Lorsque nous naviguons sur Internet, nous visitons différents sites et regardons des pages dans notre navigateur. Ces pages sont écrites ou traduites en langage HTML, (HyperText Markup Language), et c'est notre navigateur qui les interprète pour les afficher. Ceci signifie que c'est simplement du texte qui est envoyé du serveur au client, donc le transfert est beaucoup plus rapide que si c'était une image de la page qu'il fallait faire parvenir à travers le réseau. De plus le texte contient du code qui assure la possibilité d'une action sur la page reçue, ce qui ne serait pas le cas avec une image.

Les pages d'un site donné, ainsi que celles d'autres sites, sont reliées par des liens. Il suffit de cliquer sur un lien pour atteindre une nouvelle page. Ce système de liens ressemble à une toile d'araignée étendue sur le monde entier et s'appelle le World Wide Web.

Le HTML, à la base du fonctionnement de tous les sites Web, est un langage qui utilise des balises, plus précisément un langage de balisage d'hypertexte. Un hypertexte est un texte qui contient des éléments munis de liens permettant d'accéder à d'autres parties du texte ou à d'autres ressources. Le HTML n'est pas un langage de programmation.

Cette section est une petite introduction, juste le nécessaire pour pouvoir démarrer. Plus loin, au chapitre 6, des développements sont présentés.

Pour écrire du HTML, il suffit d'un éditeur de texte comme Notepad++. Un navigateur permet ensuite d'afficher les pages. Le HTML permet d'écrire et d'organiser le contenu des pages comme les paragraphes et les titres. Il ne gère pas la mise en forme. On utilise pour cela le CSS qui gère les emplacements, les couleurs, les tailles. Naturellement, il existe de nombreuses versions et la dernière est HTML5 (bientôt HTML6?). En résumé, nous avons le HTML pour le contenu et le CSS pour l'apparence.

Le navigateur est un programme qui interprète les langages comme HTML et CSS pour savoir ce qu'il doit afficher. Il existe de nombreux navigateurs pour ordinateur et chacun a ses particularités dans l'affichage d'une page. Des navigateurs sont aussi conçus pour les tablettes et les smartphones et là, chacun système d'exploitation a le sien, ou presque.

1.1 Création d'une page

Les balises vont par deux, comme les parenthèses, une ouvrante et une fermante. Il y a quelques exceptions, on les appelle des balises orphelines. Il faut prendre l'habitude, dès qu'on écrit une balise ouvrante, d'écrire la balise fermante correspondante. Le code HTML est écrit entre les balises `<html>` et `</html>`. La norme veut qu'on commence l'écriture du fichier par la balise `<!DOCTYPE html>`. Nous allons voir le processus de création d'une page.

Ouvrir Notepad++, dans la barre de menu, cliquer sur Encodage puis sur Encoder en UTF-8. Ceci permet en particulier de pouvoir écrire les lettres accentuées. Ensuite cliquer sur Langage puis à la lettre H sur HTML. Écrire alors les lignes suivantes :

```
<!DOCTYPE html>
<html>

</html>
```

L'éditeur reconnaît les balises et les colorie.

Enregistrer le fichier, par exemple sous le nom `page00.html`. Cliquer ensuite sur `Exécution` puis choisir le navigateur parmi ceux proposés. La page s'ouvre alors dans le navigateur et bien sûr ne contient rien. On peut aussi ouvrir un navigateur et demander à ouvrir le fichier, ou bien effectuer un double-clic sur le fichier `page00.html` qui s'ouvre alors avec le navigateur par défaut.

Entre les balises `<html>` et `</html>`, nous insérons des balises qui définissent l'entête de la page et le corps de la page (`head` et `body`). Dans l'entête nous ajoutons une balise qui définit l'encodage des caractères (`utf-8`) et une balise pour le titre de la page.

L'encodage des caractères détermine comment les caractères spéciaux seront s'affichés (accents et symboles principalement).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Titre</title>
  </head>
  <body>
  </body>
</html>
```

Enregistrer puis exécuter dans le navigateur. Le titre apparaît en haut à gauche de la fenêtre. Ce fichier modèle est le point de départ de toute page HTML.

Voyons maintenant le corps de la page en insérant des titres et des paragraphes.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Ma page 0</title>
  </head>
  <body>
    <h1>Un très gros titre</h1>
    <h6>Un tout petit titre</h6>
    <p>Un paragraphe</p>
    <p>Un paragraphe<br>sur deux lignes</p>
  </body>
</html>
```

Les balises de titres sont numérotées de `<h1>` à `<h6>`.

Remarques

L'indentation, au début de certaines lignes, décale les balises vers la droite. En langage Python, elle fait partie de la syntaxe. En HTML, elle n'est pas obligatoire et n'a aucun impact sur l'affichage de la page. Cela rend simplement le code source plus lisible.

Il est possible d'insérer des commentaires. Cela permet d'expliquer à quoi servent certaines lignes. Les commentaires ne sont pas visibles lorsque le navigateur affiche la page et ne changent rien à cet affichage. Par contre, tout le monde peut les voir avec le code HTML de la page si elle est mise en ligne sur le Web.

Pour insérer un commentaire on utilise la balise `<!-- voici un commentaire -->` à l'endroit voulu au sein du code source.

Pour voir un exemple de page complexe, il suffit d'ouvrir un navigateur, aller sur un site quelconque et demander l'enregistrement de la page. Attention, il faut demander d'enregistrer la page Web complète, sinon les images par exemple ne sont pas chargées. Dans le dossier d'enregistrement, on trouve alors un fichier avec l'extension htm ou html et un dossier du même nom que le fichier qui peut contenir des images et des fichiers de style au format css ou js.

On peut aussi afficher le code source de la page, c'est-à-dire le texte composant le fichier, à partir du navigateur, et relever toutes les balises présentes en essayant de déterminer leur rôle. En ouvrant le fichier enregistré avec Notepad++, on peut aussi supprimer toutes les balises gérant le style qui en général se termine par la balise `</style>` et observer la différence d'affichage. La partie contenant le style peut être écrite dans le fichier lui-même ou dans un fichier séparé qui est ensuite inclus dans la page.

Les noms des fichiers images sont reconnaissables grâce à leur extension png ou jpg. On peut ainsi les changer pour utiliser d'autres fichiers qui doivent se trouver dans le même dossier.

1.2 Création d'autres pages

La création de plusieurs pages se fait sur le même modèle. En général, il y a une page principale, la page d'accueil, avec des liens qui permettent de passer à d'autres pages, de l'une à l'autre, et de revenir à la page d'accueil. Les fichiers HTML sont enregistrés dans le même dossier.

Étape 1

On crée un dossier `monsite` où seront rangés tous les fichiers HTML. Les pages s'appelleront par exemple `accueil.html`, `page1.html`, `page2.html`, ... Dans le dossier `monsite`, on peut aussi créer un dossier `images` où seront enregistrées les images.

On utilise le code minimal d'une page HTML vu plus haut pour la page d'accueil. Le fichier est enregistré sous le nom `accueil.html`.

Étape 2

Voici quelques balises qui permettent d'organiser le texte et créer des pages supplémentaires :

- les balises de titres `<h1> ... </h1>`, jusqu'à `<h6> ... </h6>`;
- les balises `<p> ... </p>` pour les paragraphes;
- les balises `
` pour un retour à la ligne et `<hr>` pour une séparation horizontale, (ce sont des balises orphelines);
- les balises de mise en valeur du texte ` ... `, ` ... `, `<mark> ... </mark>`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Ma page 0</title>
  </head>
  <body>
    <h1>Un très gros titre</h1>
    <h6>Un tout petit titre</h6>
    <p><em>Un paragraphe</em></p><hr>
    <p><strong>Un paragraphe</strong><br>
      <mark>sur deux lignes</mark></p>
  </body>
</html>
```

Nous complétons le code avec des balises qui permettent d'afficher des listes.

Le code HTML est par exemple le suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page d'accueil</title>
  </head>
  <body>
    ...
  <hr>
  <h2>Les balises de listes</h2>
  <ul>
    <li>informatique</li>
    <li>mathématiques</li>
    <li>physique</li>
  </ul>
  ou bien
  <ol>
    <li>informatique</li>
    <li>mathématiques</li>
    <li>physique</li>
  </ol>
</body>
</html>
```

Étape 3 : les liens

Jusque là nous n'avons que du texte. Il est temps de passer à l'hypertexte. Comment passer d'une page à une autre ? C'est ici qu'interviennent les liens, ou hyperliens, inventés par Ted Nelson en 1965. Un lien est créé avec les balises `<a> ... `.

Un lien vers un autre site

Dans la balise ouvrante `<a>` (pour ancre ou anchor), on place l'attribut `href` suivi par l'URL de la page encadrée de guillemets, par exemple `"http://www.monlycee.fr/"`, pour la page d'accueil du site de son lycée. Disons que l'URL est l'adresse de la page.

Entre les balises `<a>` et ``, on écrit en général un texte descriptif. Cela donne le résultat :

`Site du Lycée`. Avec la plupart des navigateurs, le texte `Site du lycée` apparaît en bleu et est souligné.

Ce lien doit être inséré entre les balises `<body> ... </body>`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page d'accueil</title>
  </head>
  <body>
    ...
  <hr>
  <a href="http://monlycee.fr/">Site du lycée</a>
</body>
</html>
```

Remarque

Un lien peut afficher une infobulle quand on passe le pointeur de la souris dessus. Pour cela, on utilise l'attribut `title`. Il s'agit d'une aide ou d'une information pour le visiteur avant qu'il ne clique sur le lien.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page d'accueil</title>
  </head>
  <body>
    ...
  <hr>
    <a href="http://monLycee.fr/" title="c'est mon lycée">Site du lycée</a>
  </body>
</html>
```

Un lien vers une autre page de son site

Pour placer un lien sur la page d'accueil qui amène vers la page 1, si les deux fichiers sont enregistrés dans le même dossier, on écrit comme cible du lien le nom du fichier de la page 1. Par exemple : ``. Il s'agit alors d'un lien relatif.

Si le fichier `page1.html` est situé dans un sous-dossier `pages` par exemple, il faut alors préciser le nom du dossier : ``.

Si le fichier `page1.html` est situé dans un dossier parent, on écrit ``.

Un lien vers une ancre

Une ancre est un point de repère que l'on peut placer dans une page HTML si celle-ci est très longue. Un lien vers cette ancre permet alors d'aller directement à la partie qui nous intéresse. Pour créer une ancre, il suffit d'ajouter l'attribut `id` à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise. L'attribut `id` sert à donner un nom unique à cette balise.

Par exemple :

```
<h1 id="ancre1">Un paragraphe intéressant</h1>
```

Ensuite on crée le lien comme pour un lien vers une page, mais cette fois l'attribut `href` contient un dièse (#) suivi du nom de l'ancre.

Exemple :

```
<a href="#ancre1">Aller vers l'ancre 1</a>
```

Un lien pour envoyer un courriel

Pour que les visiteurs puissent envoyer un courriel, on utilise des liens de type `mailto`. Rien ne change au niveau de la balise, il suffit de modifier la valeur de l'attribut `href` comme ceci :

```
<a href="mailto:nom@toto.fr">Envoyez un courriel</a>
```

Un lien pour télécharger un fichier

Pour le téléchargement d'un fichier, on procède comme pour un lien vers une page Web, mais en indiquant le nom du fichier à télécharger.

Par exemple, pour télécharger "fichier.txt" qui est dans le dossier fichiers :

```
<a href="fichiers/fichier.txt">Télécharger le fichier</a>
```

1.3 Les images

Pour l'instant les pages ne contiennent que du texte et ce serait sans doute plus agréable avec quelques images.

Il existe de nombreux formats pour les fichiers d'images (PNG, JPEG, GIF, EPS, TIFF, ...).

Les formats utilisés sur le Web sont :

- le JPEG (Joint Photographic Expert Group), qui réduit le poids des photos pouvant comporter plus de 16 millions de couleurs différentes, enregistré avec l'extension .jpg ou .jpeg ;
- le PNG (Portable Network Graphics), adapté aux graphiques (ou tout ce qui n'est pas une photo), existe en version 8 bits (256 couleurs) et en version 24 bits (16 millions de couleurs) ;
- le GIF, format ancien, limité à 256 couleurs, mais avec l'avantage qu'il peut être animé.

Attention, pour éviter des problèmes, il faut prendre l'habitude d'enregistrer les fichiers avec des noms en minuscules, sans espace ni accent, par exemple : mon_image_n1.png.

La balise utilisée pour insérer une image est ``, c'est une balise de type orpheline qui est accompagnée d'un attribut obligatoire et d'un attribut recommandé.

- L'attribut obligatoire **src** permet d'indiquer le chemin de l'image que l'on veut insérer.
- L'attribut recommandé **alt** signifie "texte alternatif". C'est un texte court décrivant l'image qui sera affiché à la place de l'image si celle-ci ne peut pas être chargée.

Les images se placent souvent à l'intérieur d'un paragraphe `<p> ... </p>`.

Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page d'accueil</title>
  </head>
  <body>
    <h1>Bonjour</h1>
    <p>Voici une image de mon lycée :<br>
    
    </p>
  </body>
</html>
```

On peut ajouter une bulle d'aide comme pour les liens avec l'attribut `title`.

Si le fichier image est très volumineux, on peut en afficher seulement une miniature et on ajoute un lien sur cette miniature pour que les visiteurs puissent afficher l'image en taille originale.

```
<p>Voici une image miniature:<br>
Pour voir l'original, cliquez dessus<br>
<a href="images/monimage.jpg"></a>
</p>
```

Les figures

Les figures ne sont pas toujours des images. On utilise les balises `<figure> ... </figure>` :

```
<figure>
  texte
  texte
  ...
</figure>
```

L'intérêt est de pouvoir ajouter une légende. On utilise la balise `<figcaption>` à l'intérieur de la balise `<figure>` :

```
<figure>
  
  <figcaption>Mon lycée</figcaption>
</figure>
```

De plus une figure peut comporter plusieurs images avec une légende en dessous.

```
<figure>



<figcaption>Mes images</figcaption>
</figure>
```

1.4 Balises HTML

Les balises HTML sont très nombreuses. Une liste des plus courantes est résumée ici.

Rappelons qu'il existe deux types de balises : les balises en paires et les balises orphelines. Les balises en paires encadrent souvent un texte. Les balises orphelines servent plus souvent à insérer un élément comme une image ``, un retour à la ligne `
`, à un endroit précis de la page.

Structure générale d'une page

Ce sont les balises `<html>`, `<head>` et `<body>`.

Entête

Ces balises sont placées entre les balises `<head>` et `</head>`.

`<title>` : titre de la page.

`<meta />` : métadonnées de la page Web (charset, mots-clés, etc.).

`<link />` : lien avec une feuille de style.

`<style>` : code CSS.

`<script>` : code JavaScript.

Corps

Ces balises sont placées entre les balises `<body>` et `</body>`.

`` : mise en valeur forte; `` : mise en valeur normale; `<mark>` : mise en valeur visuelle.

`<h1>` : titre de niveau 1; `<h2>` : titre de niveau 2; `<h3>` : titre de niveau 3; `<h4>` : titre de niveau 4; `<h5>` : titre de niveau 5; `<h6>` : titre de niveau 6.

`<p>` : paragraphe.

`<a>` : lien hypertexte.

`
` : retour à la ligne; `<hr>` : séparation horizontale.

`` : image.

`<figure>` : figure; `<figcaption>` : description d'une figure.

`<audio>` : son; `<video>` : vidéo; `<source>` : format source pour les balises `<audio>` et `<video>`.

Des balises spécifiques

Pour créer des listes.

`` : liste à puces, non numérotée, `` : liste numérotée, `` : élément de la liste à puces.

Pour créer des tableaux.

`<table>` : tableau; `<caption>` : titre du tableau.

`<tr>` : ligne de tableau; `<th>` : cellule d'entête; `<td>` : cellule.

Pour créer des formulaires.

`<form>` : formulaire.

`<fieldset>` : groupe de champs; `<legend>` : titre d'un groupe de champs; `<label>` : libellé d'un champ.

`<input />` : champ de formulaire (texte, mot de passe, case à cocher, bouton, etc.).

`<textarea>` : zone de saisie à plusieurs lignes, `<select>` : liste déroulante.

Des balises génériques

Les balises HTML ont un sens, par exemple `<p>` signifie "paragraphe". Il existe des balises, appelées balises génériques qui n'ont pas de sens. On les utilise quand aucune autre ne convient.

Il y a deux balises génériques :

`<div>` : balise générique de type block;

`` : balise générique de type inline.

Ces balises sont intéressantes lorsqu'on utilise du code en langage CSS. La section suivante en propose une présentation.

2 Le langage CSS

Le CSS (Cascading Style Sheets, "feuilles de style en cascade") est utilisé pour mettre en forme des documents HTML. Le contenu du document est écrit en HTML et il est séparé des éléments de mise en forme qui sont écrits en CSS. Ainsi il est simple d'obtenir la même apparence pour toutes les pages d'un site Web.

Le CSS est un standard adopté par le World Wide Web Consortium (W3C) en 1996.

Pour cette présentation, il est nécessaire de disposer d'un éditeur de texte et d'un navigateur pour contrôler l'apparence de la page Web.

Il y a plusieurs manières de procéder. Commençons par la plus simple qui consiste à ajouter un attribut dans une balise HTML et à préciser une valeur pour une propriété de l'attribut.

2.1 Attribut dans une balise

On peut ajouter de nouveaux attributs dans des balises HTML. Par exemple, pour souligner un titre entre des balises `<h2> ...</h2>`, on ajoute l'attribut `style` dans la balise. On insère ensuite la propriété `text-decoration` et enfin on donne une valeur à cette propriété, par exemple la valeur `underline` pour souligner le texte.

Voici la syntaxe précise :

```
<html>
  <head>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h2 style="text-decoration:underline"> Un test avec CSS </h2>
  </body>
</html>
```

Pour changer la couleur d'un texte, on utilise la propriété `color`. Voici quelques exemples :

```
<html>
  <head>
    <meta charset="utf-8"/>
```



```
</head>
<body>
  <h2 style="color:blue"> Un test avec CSS en bleu</h2>
  <h2 style="color:red"> Un test avec CSS en rouge</h2>
  <h2 style="color:green"> Un test avec CSS en vert</h2>
</body>
</html>
```

Un attribut peut avoir deux propriétés séparées par un point-virgule.

```
<html>
  <head>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h2 style="text-decoration:underline;color:blue">Deux propriétés</h2>
  </body>
</html>
```

L'attribut `style` possède d'autres propriétés intéressantes.

```
<html>
  <head>
    <meta charset="utf-8"/>
  </head>
  <body>
    <p><span style="font-weight:bold">Texte en gras </span></p>
    <p><span style="font-style:italic">Texte en italique </span></p>
    <p><span style="text-decoration:overline">Texte surligné </span></p>
    <p><span style="text-decoration:line-through">Texte barré </span></p>
  </body>
</html>
```

2.2 Règle CSS dans l'entête

Dans le dernier exemple, le formatage CSS est inclus dans la balise `` avec l'attribut `style`, et s'applique donc à une partie de texte sans influence sur le reste. Mais pour pouvoir appliquer les mêmes propriétés, par exemple à tous les liens d'une page, il vaut mieux écrire une "règle" CSS, c'est-à-dire des propriétés avec leurs valeurs qui s'appliquent aux éléments de la page. Pour cela, nous ajoutons un bloc `style` dans l'entête de la page. Par exemple :

```
<html>
  <head>
    <meta charset="utf-8"/>
    <style type="text/css">
      body{background-color:rgb(120,200,230);color:blue;}
      p{color:red;}
    </style>
  </head>
  <body>
    <p><a href="#">lien</a></p>
  </body>
</html>
```

```
        </style>
    </head>
    <body>
        <p><span style="font-style:italic">Texte en italique </span></p>
    </body>
</html>
```

Nous modifions ainsi la couleur du fond et celle du texte dans le corps de la page, ainsi que celle des paragraphes. (Attention aux accolades et aux points-virgules).

Il est possible de modifier la couleur par défaut des liens ainsi que le style du texte d’ancrage des liens, quand ils sont survolés par le pointeur de la souris. On utilise `a: hover`.

```
<html>
  <head>
    <meta charset="utf-8"/>
    <style type="text/css">
      body{background-color:rgb(120,200,230);color:blue;}
      p{color:red;}
      a{color:purple;}
      a:hover{color:green;}
    </style>
  </head>
  <body>
    Bonjour
    <p> <span style="font-style:italic">Texte en italique </span></p>
    <a href="page1.html">Une page</a>
  </body>
</html>
```

Positionnement d’un texte et d’une image

Le positionnement d’un élément sur une page est un élément primordial de la mise en forme. Trois propriétés sont principalement utilisées pour positionner un élément :

- ▶ `position` avec les valeurs `absolute` qui utilise le coin supérieur gauche de la fenêtre comme point d’origine ou `relative` qui utilise la position qu’aurait l’élément s’il n’avait aucune information de positionnement ;
- ▶ `top` avec une valeur en pixels (ou pouces ou centimètres) qui donne le déplacement vertical, vers le bas si la valeur est positive, relativement au point d’origine ;
- ▶ `left` avec une valeur en pixels (ou pouces ou centimètres) qui donne le déplacement horizontal, vers la droite si la valeur est positive, relativement au point d’origine.

Tester l’exemple qui suit et effectuer des modifications sur les valeurs des positions pour bien comprendre le résultat obtenu dans un navigateur.

```
<html>
  <head>
    <meta charset="utf-8"/>
    <style type="text/css">
      body{background-color:rgb(120,200,230);color:blue;}
    </style>
  </head>
  <body>
```

```
        p{color:red;}
        a{color:purple;}
        a:hover{color:green;}
    </style>
</head>
<body>
    <h1>Bonjour</h1>
    <p> <span style="font-style:italic">Texte en italique </span></p>
    <a href="page1.html">Une page</a>
    <p style="position:relative;top:50px;left:100px;">
        Paragraphe positionné avec relative </p>
    <p style="position:absolute;top:250px;left:50px;">
        Paragraphe positionné avec absolute </p>
</body>
</html>
```

2.3 Fichier de style

Le code CSS a été écrit soit directement dans les balises du fichier HTML avec l'attribut `style`, soit dans l'entête du fichier HTML entre les balises `<style>` et `</style>`.

Une troisième possibilité est d'écrire ce code dans un fichier séparé. Le nom de ce fichier a une extension ".css". En général ce fichier est enregistré dans un dossier nommé "styles" au même niveau que la page HTML.

On commence par ajouter la ligne `<link rel="stylesheet" href="styles/style.css"/>` dans l'entête de la page, après la ligne `<meta charset="utf-8"/>`.

Créons, avec l'éditeur Notepad++, un nouveau fichier qui contient par exemple le texte :

```
body{
    color:red;
}

p{
    color:blue;
}
```

Nous nommons ce fichier "style.css" et l'enregistrons dans le dossier "styles". Nous testons ensuite avec un fichier HTML.

```
<html>
  <head>
    <meta charset="utf-8"/>
    <link rel="stylesheet" href="styles/style.css" />
  </head>
  <body>
    <h1>Bonjour</h1>
    <p>Mon paragraphe 1</p>
    <p>Mon paragraphe 2</p>
  </body>
</html>
```

Lorsqu'on ouvre la page HTML, le corps de la page est bien écrit en rouge et les paragraphes sont en bleu.

Cette méthode est intéressante pour appliquer le même style aux différentes pages d'un site. De plus, pour faire une modification, il suffit de modifier une fois le fichier "style.css", ce qui est bien plus simple que corriger toutes les pages d'un site une par une.

Le contenu du fichier est donc composé d'une suite de noms de balises avec des propriétés et des valeurs. Si deux balises ont le même style, par exemple si les titres <h1> et <h2> sont en bleu, on peut écrire dans le fichier CSS :

```
h1, h2{
  color:blue;
}
```

Remarque : dans un fichier CSS, un commentaire est écrit entre les symboles /* et */.

L'attribut class

Pour ne pas appliquer le même style à tous les paragraphes par exemple, nous pouvons utiliser l'attribut `class` qui permet de donner un nom à une balise particulière :

```
<p class="paragraphe1">Mon paragraphe 1</p>
```

Dans le fichier CSS, on écrit alors, en n'oubliant pas le point au début de la ligne :

```
.paragraphe1{
  color:green;
  background-color:rgb(200,200,230);
}
```

Ainsi, seules les balises nommées "paragraphe1" sont affichées en vert sur un fond gris bleu.

Pour modifier seulement une partie du texte, on applique la même méthode en utilisant les balises ` ...`.

Par exemple :

```
<p class="paragraphe1">Mon paragraphe<span class="num">1</span></p>
```

et on écrit dans le fichier CSS :

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

2.4 Commandes CSS

Voici quelques commandes permettant de gérer le texte.

Taille du texte et police

Pour une taille absolue en pixels, `font-size: 22px;`

Pour une taille relative, `font-size: large;` (la taille est prédéfinie), ou alors en pourcentage `font-size: 1.5em;` (la taille est multipliée par 1,5).

On peut choisir la famille de police, (Arial, Roman, etc), avec `font-family` et utiliser ensuite d'autres commandes comme `font-style: italic;` et `font-weight: bold;`.

Alignement

Le texte peut être aligné à gauche, à droite, centré, ou justifié avec la propriété `text-align` : `text-align:left;` ou `text-align:right;` ou `text-align:center;` ou `text-align:justify;` (le texte remplit les lignes).

Le texte peut "habiller" une image. On utilise pour cela la propriété `float` :

```
<p>
```

Un texte dans un paragraphe qui habille une

image flottante avec un code dans un fichier css </p>

Le code à écrire dans le fichier CSS est :

```
.imageflottante{  
    float:left;  
}
```

De nombreuses autres commandes sont disponibles pour l'utilisation de couleurs, de bordures et d'ombres ou d'apparence dynamique. Il est intéressant d'étudier le contenu à l'intérieur des balises <style> et </style> d'une page d'accueil simple d'un site. On peut copier ce contenu dans un fichier "style2.css", le modifier et l'utiliser.

Attention : les propriétés CSS sont héritées en cascade, c'est-à-dire que si on donne un style à un élément, tous les sous-éléments auront le même style. C'est le phénomène de l'héritage. Donc les balises qui se trouvent à l'intérieur d'une autre balise "héritent" de ses propriétés.