NSI première - IHM sur le web

HTML et CSS : cours

HTML et CSS : rappels

Le couple HTML + CSS sont les deux langages utilisés pour présenter et mettre en forme les documents sur le web.

HTML

HTML est un *langage de balise* et non un *langage de programmation.*

HTML permet de *décrire* le contenu d'un document et d'en donner *la structure.*

HTML n'étant pas un langage de programmation, il n'existe pas de variables ou de conditions, fonctions etc. en HTML.

Voici la définition accessible sur Wikipédia :

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias, dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

Langage de balise

Le principe des balises et d'encapsuler un contenu entre une balise ouvrante et une balise fermante.

Par exemple :

```
<div>
    Le contenu de cette balise div
</div>
```

L'action de ces balises porte alors sur la partie du texte qu'elle contient.

Les balises peuvent être imbriquées.

De la même manière qu'une série de parenthèse comme ceci : `[( ])` ne convient pas, les balises doivent être correctement ordonnées :

```
<div>
    <p>
        Un paragraphe
    </div>
</p>
```

Cet exemple ne convient pas ! Comment le rectifier ?

On étudiera, en terminale, un algorithme permettant de repérer (et parfois résoudre) ces problèmes de balises incorrectement ordonnées.

C'est ce qu'on appelle un `parser` de document.

Structure des pages web

L'ordre dans lequel sont décrits les balises permet de construire une structure de la page.

Tout en haut, à la racine de cette arbre, on trouve la balise `<html>`.

L'ensemble du document est compris entre `<html>` et `</html>`

Dans ce contenu on trouve alors deux parties :

- `<head>` qui contient la description de l'entête (nom de l'onglet, propriétés parfois les fichiers importés etc.)
- `<body>` qui contient le corps de la page.

```
<html>
    <head>
    La description du document
    </head>

    <body>
    Ce qui sera affiché dans la page
    </body>
</html>
```

Intéressons-nous au document lui même.

Les balises qu'il contient ont toutes un rôle :

- `<div>` décrire un bloc qui sera aligné *verticalement* avec les autres blocs ;
- `<span>` décrire un bloc qui sera aligné *horizontalement* avec les autres blocs ;
- `<p>` décrire un paragraphe ;
- `<h1>`, `<h2>` etc. décrire un titre, un sous-titre etc. ;
- `<form>` contient un formulairaire permettant à l'utilsateur d'envoyer des informations au serveur ;
- `<button>` un bouton (qui l'eut cru ?) ;
- `<ol>` `<ul>` listes numérotées ou non ;
- `<li>` un élément d'une liste
- `<table>` un tableau… dont la structure est aussi séparée entre entête `<thead>` et contenu `<tbody>`
- `<tr>` table *row*, une ligne et `<tr>` table *data*, une cellule de la ligne.
- `<a>` un lien ;
- `<img>` une image

etc.

Remarquons que ces balises visent toutes à décrire la structure de la page et non sa mise en forme.

Remarquons aussi, que pour décrire un tableau contenant mille lignes, on doit produire 1000 éléments `<tr>` et autant d'éléments `<td>` que de colonnes… et ce pour chaque ligne.

Deux constats immédiats :

1. La mise en forme du document doit pouvoir être faite de manière automatique, si possible en décrivant ce qu'on souhaite faire. C'est le rôle de CSS
2. La construction de cette structure respecte une organisation simple. Dans la majorité des cas, il est possible de la générer automatiquement ! C'est en partie le rôle de *javascript* que nous aborderons plus tard.

Quelques mots sur Javascript

Comme Python, C ou Golang, Javascript est un *langage de programmation.*

Javascript n'est pas le seul langage qui permette d'écrire du code HTML, tous les langages modernes peuvent être utilisés pour générer une page web automatiquement et structurer des données.

Javascript est par contre le seul qui puisse être *exécuté dans le navigateur du client.*

En quoi est-ce intéressant ? Cela permet de réaliser les calculs *sur la machine du client* et non sur le serveur directement. Aussi on limite la charge du serveur !

D'une structure à son arbre

```
<html>
    <head>
        <title>Le titre dans l'onglet du navigateur</title>
    </head>
    <body>
        <div>
            <h1>Un titre</h1>
            <p>Un paragraphe</p>
        </div>
        <div>
        <table>
            <caption>Alien football stars</caption>
            <tr>
                <th scope="col">Player</th>
                <th scope="col">Gloobles</th>
                <th scope="col">Za'taak</th>
            </tr>
            <tr>
                <th scope="row">TR-7</th>
                <td>7</td>
                <td>4,569</td>
            </tr>
            <tr>
                <th scope="row">Khiresh Odo</th>
                <td>7</td>
                <td>7,223</td>
            </tr>
            <tr>
                <th scope="row">Mia Oolong</th>
                <td>9</td>
                <td>6,219</td>
            </tr>
        </table>
        </div>
        <div>
            <ul>
                <li>Milk</li>
                <li>Cheese
                    <ul>
                        <li>Blue cheese</li>
                        <li>Feta</li>
                    </ul>
                </li>
            </ul>
        </div>
    </body>
</html>
```

Construire l'arbre décrivant la structure de cette page. Attention… il est imposant !

Pour vérifier.

Propriétés d'une balise

Il est possible d'inclure des propriétés DANS la description d'une balise.

Par exemple :

```
<div class="ma-classe">
    Une div avec la classe "ma-classe"
    Plusieurs balises peuvent porter la même classe.
```

```
</div>
<p id="mon-id">
    Un paragraphe avec l'id "mon-id"
    Une seule balise par id.
</p>
```

CSS

Si HTML décrit la structure du document, CSS en réalise la mise en forme.

C'est CSS qui permet d'appliquer des couleurs, des changements de fontes, de taille etc.

CSS pour "Cascading Style Sheets" ou "feuilles de style en cascadre"

Le principe de CSS est d'appliquer une mise en forme à un ou des éléments sélectionnés.

Ainsi :

```
p {
    color: red;
}
```

Va :

1. sélectionner **tous** les *paragraphes* contenus dans des balises `<p>`
2. Leur appliquer les commandes entre les `{  }`

   Dans notre exemple, `color` désigne la couleur du texte, qui sera rouge.

Anatomie d'une règle CSS

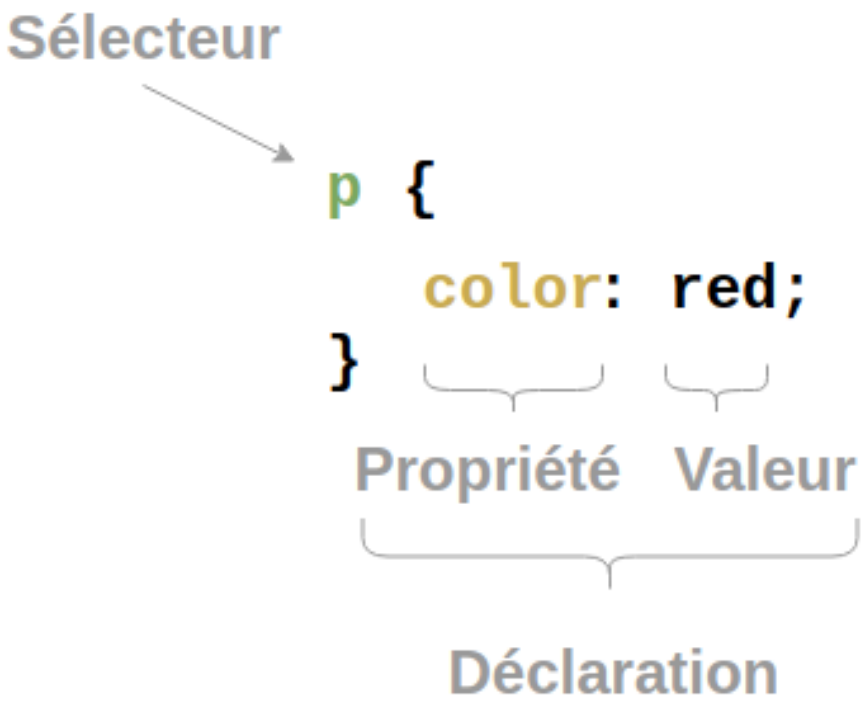

Figure 1: règle css

Sélecteur

C'est le nom de l'élément HTML situé au début de l'ensemble de règles. Il permet de sélectionner les éléments sur lesquels appliquer le style souhaité (en l'occurence, les éléments `p`). Pour mettre en forme un élément différent, il suffit de changer le sélecteur.

Déclaration

C'est une règle simple comme `color: red;` qui détermine les propriétés de l'élément que l'on veut mettre en forme.

Propriétés

Les différentes façons dont on peut mettre en forme un élément HTML (dans ce cas, color est une propriété des éléments 'p'). En CSS, vous choisissez les différentes propriétés que vous voulez utiliser dans une règle CSS.

Valeur de la propriété

À droite de la propriété, après les deux points, on a la valeur de la propriété. Celle-ci permet de choisir une mise en forme parmi d'autres pour une propriété donnée (par exemple, il y a d'autres couleurs que red pour la propriété color).

Les autres éléments importants de la syntaxe sont :

- chaque ensemble de règles, à l'exception du sélecteur, doit être entre accolades (`{}`).
- pour chaque déclaration, il faut utiliser deux points (`:`) pour séparer la propriété de ses valeurs.
- pour chaque ensemble de règles, il faut utiliser un point-virgule (`;`) pour séparer les déclarations entre elles.

Ainsi, si on veut modifier plusieurs propriétés d'un coup, on peut utiliser plusieurs déclarations dans une seule règle en les séparant par des points-virgules :

```
p {
  color: red;
  width: 500px;
  border: 1px solid black;
}
```

Sélectionner plusieurs éléments

Il est aussi possible de sélectionner plusieurs types d'éléments pour appliquer à tous une même règle. Il suffit de placer plusieurs sélecteurs, séparés par des virgules. Par exemple :

```
p,li,h1 {
  color: red;
}
```

Les différents types de sélecteurs

Il y a différents types de sélecteurs. Dans les exemples précédents, nous n'avons vu que les sélecteurs d'élément qui permettent de sélectionner les éléments HTML d'un type donné dans un document HTML. Mais ce n'est pas tout, il est possible de faire des sélections plus spécifiques. Voici quelques-uns des types de sélecteur les plus fréquents :

- Les sélecteurs de balise.

  Ils sélectionnent toutes les balises portant ce nom.

  ```
  p {
      color: red;
  }
  ```

- Les sélecteurs d'ID

  Ils sélectionnent LA balise portant cette id

  ```
  #mon-id {
      background-color: black;
  }
  ```

- Les sélecteurs de classes

  Ils sélectionnent TOUTES les balises portant cette classe

  ```
  .ma-classe {
      font-style: verdana;
  }
  ```

- Citons aussi les sélecteurs de “pseudo-classe”

  Ils sélectionnent les éléments donnés mais uniquement dans un certain état

  ```
  a:hover {
      color: blue;
  }
  ```

  Sélectionne les liens *uniquement quand la souris passe dessus*

Les grands types de mise en forme

On peut séparer en quelques grandes catégories les propriétés appliquées

Les polices de caractères

Ce sont les propriétés qu’on applique au texte lui même : couleur, taille, épaisseur, fonte etc.

Les boites

Ces propriétés décrivent les marges intérieures (`padding`) et extérieures (`margin`) ainsi que la bordure (`border`) ainsi que la taille en largeur, hauteur, la couleur de fond etc.

L’alignement

Souhaite-t-on aligner horizontalement, verticalement, avoir une image positionnée précisement ou à une position dépendant des dimensions de la page ?

Figure 2: html et css

La fenêtre de développement

C'est devenu depuis une dizaine d'année l'outil de base du développeur web. Tous les navigateurs modernes permettent de consulter le code d'une page et de l'éditer localement.

On peut ainsi appliquer le style, débugguer les erreurs et mesurer les performances dans un outil complet et intégré au navigateur.

Cela ne dispense pas d'écrire le code de sa page dans un éditeur de texte mais cela simplifie considérablement les efforts.