

NSI – 1ere	COURS Séquence 5A : Modalités de l'interaction Homme Machine sur le WEB.	LFV
------------	---	-----

I. Quelques rappels sur le WEB.

1. Historique.

Le "World Wide Web", également appelé "**le Web**", a été développé par le Britannique Sir Timothy John Berners-Lee et le Belge Robert Cailliau au début des années 90. À cette époque les principaux centres de recherche mondiaux étaient déjà connectés les uns aux autres via un réseau de réseaux nommé "**Internet**", mais pour faciliter les échanges d'information Tim Berners-Lee utilise un système existant, le système hypertexte, qui permet, à partir d'un document, de consulter d'autres documents en cliquant sur des **mots clés**. Ces mots "cliquables" sont appelés hyperliens. Les hyperliens sont plutôt connus aujourd'hui sous le simple terme de "liens internet".

La première page web mise en ligne est toujours consultable à l'adresse <http://info.cern.ch/hypertext/WWW/TheProject.html> sur le site du CERN (Conseil Européen pour la Recherche Nucléaire), organisme où le WEB a été développé.

2. Les bases du Web.

La navigation sur le Web nécessite :

- Un serveur WEB : un logiciel sur lequel il est possible de se connecter afin d'accéder à des pages web stockées sur ce serveur.
- Un navigateur WEB : un logiciel qui se connecte à un serveur WEB et qui affiche les pages WEB reçues.

Techniquement le web se base sur trois éléments :

- Le protocole HTTP (HyperText Transfert Protocol) : ce protocole de la couche application permet la bonne communication entre le navigateur WEB et le serveur WEB (vous connaissez forcément son nom).
- Les URL (Uniform Resource Locator) : il s'agit d'une adresse **unique** permettant de se connecter au bon serveur (par exemple <https://fr.wikipedia.org/wiki/Chaton>).
- Les langages de description HTML (HyperText Markup Language) et CSS (Cascade StyleSheet) : deux langages permettant de construire des pages WEB.

II. Le langage de description HTML

HTML (ou Hypertext Markup Language) permet de créer la structure d'une page web. Grâce à un jeu de balises entourée par < et >, il permet de décomposer la page comme un traitement de texte : titre, sous-titre, section etc. On peut par exemple indiquer que l'on utilise un titre avec les balises ouvrante et fermante <h1></h1> ou indiquer l'utilisation d'une énumération avec et pour chaque item. Certaines balises peuvent avoir des attributs que l'on retrouve dans la balise ouvrante comme <html lang="fr"> où lang est un attribut de la balise html et qui a pour valeur "fr" qui indique que la page est en français.

a. Généralités.

Pour décrire le contenu d'un document HTML, on le structure avec ces balises qui vont donner des indications sur le sens du texte. Ces indications permettent :

- *Au navigateur d'appliquer une mise en forme* par défaut très succincte, par exemple les titres seront plus gros que le corps du texte (la mise en forme finale est dédiée à un autre langage, le CSS).
- *De donner du sens* aux éléments du document de façon à aider les moteurs de recherche ou autres robots à analyser le document.

Remarque : Chaque balise est décrite dans un document fourni par le W3C (World Wide Web Consortium), une communauté internationale qui édite les standards du web.

b. Structure minimale d'une page HTML

Une page HTML comporte toujours le code minimal représenté ci-contre : lorsque vous écrirez le code d'une page html vous copierez et collerez donc ce code dans votre nouvelle page et vous modifierez cette trame de départ :

- La balise `title` contient le nom de l'onglet du navigateur
- La balise `body` contient le contenu de votre page

```
<!DOCTYPE html>
<html lang="fr">

  <head>
    <title>Vous êtes en NSI</title>
    <meta charset="UTF-8"/>
  </head>

  <body>
    Ma première page en HTML
  </body>

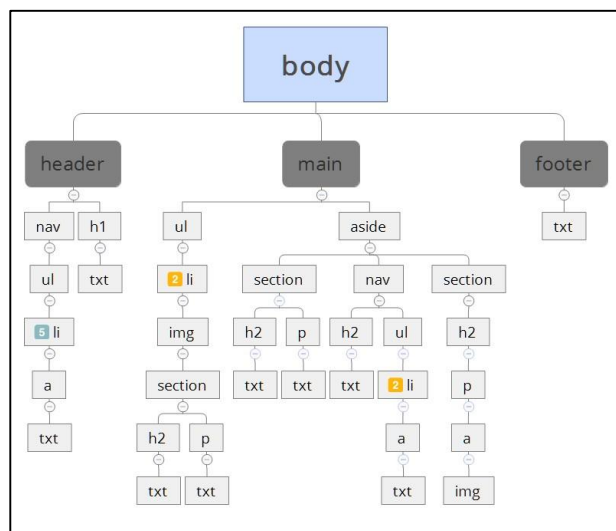
</html>
```

c. Hiérarchie des balises

Les balises ouvrantes et fermantes doivent respecter entre elles les mêmes règles que pour un parenthésage. Par exemple voici des exemples de code. Parmi eux un seul a des balises correctement imbriquées :

<pre><h1> ... </h1> <p> ... </p> <p> ... </p></pre>	<pre><h1> ... </h1> <p> ... </p> <p> ... </p></pre>	<pre><h1> ... </h1> <p> </p> <p> ... </p></pre>	<pre><h1> ... </h1> <p> </p> <p> ... </p></pre>
--	--	--	--

Le fait que les balises sont imbriquées les unes dans les autres traduit naturellement le fait que les éléments du document sont imbriqués les uns dans les autres. On peut visualiser ces imbrications à l'aide d'un arbre qui va résumer la structure d'un document HTML. Sur l'exemple ci-contre, le corps du document est composé d'un header, d'un main et d'un footer. Le header contient une balise `nav` et une balise `h1`. La balise `nav` du header contient etc.



d. Les différents types d'éléments en HTML.

En HTML, tout élément est soit de type BLOCK ou INLINE.

Un élément de type block va toujours commencer sur une nouvelle ligne et prendre toute la largeur disponible dans la page ; il peut contenir d'autres éléments de type block ainsi que des éléments de type inline. Les éléments HTML de type block les plus communs sont les suivants : `p`, `h1`, `h2`, `h3`, `ol`, `ul`, `form`, `div`

Les éléments de type inline ne vont pas commencer sur une nouvelle ligne mais s'insérer dans la ligne actuelle. Ils ne prennent que la largeur qui leur est nécessaire (c'est-à-dire la largeur de leur contenu). Les éléments HTML de type inline les plus communs sont les suivants : `strong`, `em`, `a`, `img`, `span`.

Remarque : la balise `<a>` `` est une des plus importantes puisque c'est elle qui permet de créer des liens hypertextes et est donc à l'origine du succès du web.

e. Réaliser un fichier HTML

Quel logiciel dois-je utiliser pour écrire de l'HTML ?

Tout simplement un éditeur de texte sans mise en forme du texte (donc pas un traitement de texte comme Word ou LibreOffice : sublime text ou notepad++). Pensez à encoder votre fichier en UTF8 (ou en tout cas avec le même encodage de caractères que votre charset de la balise meta).

Au moment de l'enregistrement le fichier doit porter l'extension .htm ou .html (les deux extensions sont acceptées. Le .htm date du temps du DOS et de Windows 3.1 où les noms de fichiers étaient limités à 8 caractères pour le nom et 3 caractères pour l'extension...).

UN VALIDATEUR POUR TESTER VOTRE CODE HTML : <https://validator.w3.org/nu/#textarea>

III. Le langage de description CSS

CSS (Cascading Style Sheet) a pour rôle de gérer l'apparence d'une page web : il permet de définir le positionnement, la décoration, les couleurs, la police de telle ou telle partie de la page. Ce langage évolue au cours des années, nous en sommes aujourd'hui à CSS3.

Dans tout ce qui suit un «élément HTML» désigne un type de balise HTML (h1, h2, form, p, div etc.)

a. Appliquer un style CSS à une page HTML.

Il existe plusieurs façons d'appliquer un style CSS à une page web en HTML. Nous allons utiliser celle qui consiste à écrire notre code CSS dans un fichier séparé nommé style.css et qui sera enregistré dans le même dossier que nos fichiers en .html ; nous appellerons notre fichier style.css à partir de cette ligne insérée entre les balises <head> et </head>.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Vous êtes en NSI</title>
    <meta charset="UTF-8"/>
    <link rel="stylesheet" href="style.css"/>
  </head>
  <body>
    ...
  </body>
</html>
```

b. Les sélecteurs simples CSS

CSS permet d'appliquer un style d'affichage à un élément HTML. Il suffit donc dans le CSS de sélectionner l'élément afin de lui appliquer un style qui sera défini par ses propriétés. Prenons pour exemple l'élément paragraphe, repéré en HTML par les balises <p></p>. Nous allons lui appliquer la propriété color (c'est la couleur d'écriture) avec une valeur qui sera blue et la propriété font-size (taille de police de caractère) qui sera de 16 pixels :

```
p{
    color: blue ;
    font-size: 16px ;
}
```

ATTENTION pas d'espace entre color et :

Il est possible d'appliquer des propriétés à différents éléments en même temps :

```
p, h1{
    color: blue ;
    font-size: 16px ;
}
```

c. Les id et les class en CSS

Dans le code HTML, il est possible de donner un identifiant (comme un nom) à certaines balises grâce à l'attribut `id`. La valeur d'un `id` doit être unique dans une page car elle peut également servir à créer des liens vers un point précis d'une page web. Vous ne pouvez donc pas donner deux fois la valeur "p_licornes" aux `id` de deux paragraphes `p` qui évoqueraient des licornes. Il faudrait par exemple utiliser deux `id` "p_licornes_1" et "p_licornes_2". Voici un exemple :

Côté HTML	Côté CSS
<pre>... <p id = "p_licornes_1"> ... arc-en-ciel ... </p> <p id = "p_licornes_2"> ... magique ... </p> ...</pre>	<pre>P{ font-size: 16px ; } #p_licornes_1{ color: purple ; } #p_licornes_2{ color: pink ; }</pre>

Avec ce code CSS :

- les deux paragraphes auront une police de taille 16 pixels grâce au sélecteur `p`,
- seul le paragraphe d'`id` "p_licornes_1" aura son texte écrit en violet grâce au sélecteur `#p_licornes_1`,
- seul le paragraphe d'`id` "p_licornes_2" aura son texte écrit en rose grâce au sélecteur `#p_licornes_2`.

Lorsqu'on veut rajouter un sens sémantique à plusieurs balises on peut utiliser l'attribut `class` dans le code HTML. Cela permet d'indiquer au navigateur que plusieurs balises ont un sens sémantique en commun et cela permet aussi de leur appliquer à toutes les mêmes propriétés en CSS.

Côté HTML	Côté CSS
<pre>... <h1 class = "licornes"> ... </h1> <p class = "licornes"> ... arc-en-ciel ... </p> <p class = "licornes"> ... magique ... </p> <h1 class = "chatons"> ... </h1> <p class = "chatons"> ... mignon ... </p> ...</pre>	<pre>P{ font-size: 16px ; } .licornes{ color: purple ; } .chatons{ color: grey ; }</pre>

Avec ce code CSS :

- tous les paragraphes auront une taille de police de 16 pixels grâce au sélecteur `p`,
- le premier titre et les deux premiers paragraphes auront une couleur de texte violette grâce au sélecteur `.licornes`,
- le second titre et le troisième paragraphe auront une couleur de texte grise grâce au sélecteur `.chatons`.

Remarque : dans le cas d'un conflit, c'est le sélecteur le plus précis qui remporte.

a. Commentaires, référence CSS sur le web et validateur

Il suffit de placer votre commentaire entre `/*` et `*/` : `/* Ceci est un commentaire */`

La référence du langage CSS avec des exemples simples c'est ici : <https://www.w3schools.com/css/default.asp>

UN VALIDATEUR POUR TESTER VOTRE CODE CSS : https://jigsaw.w3.org/css-validator/#validate_by_input