

# Chapitre :

# Pages Internet



## I. Observation de pages

---

Nous utiliserons le logiciel Chrome ou firefox pour observer et comparer quelques pages Internet.

### Exercice 1

Sur chacune des pages suivantes :

- <http://www.ac-nancy-metz.fr/>
- [http://fr.wikipedia.org/wiki/Wikipédia:Accueil\\_principal](http://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal)  
(plus simplement : <http://fr.wikipedia.org/>)
- <http://www.google.fr/>
- <http://www.les-mathematiques.net/>
- <http://kastler.isn.free.fr/>

utiliser l'outil « Code source de la page » disponible en cliquant droit dans la page afin de :

- distinguer ce qui est commun à **toutes** les pages ;
- rechercher des éléments de code, des mots-clés qui semblent fréquents.

On pourra s'aider des couleurs différentes du code source.

## II. Le HTML

---

### 1. Qu'est-ce que c'est ?

L'**Hypertext Markup Language**, abrégé en **HTML**, est un langage utilisé pour écrire des pages Internet. Il utilise des balises (Markups) et permet de faire de l'hypertexte (le fait de pouvoir lier des mots d'un texte à une nouvelle page), d'où son nom.

Le contenu d'une page HTML peut être très varié :

- Du simple texte
- Des images (fixes ou animées)
- De la vidéo
- Des formulaires (permettant d'écrire un mot, de cocher des cases, etc.)
- Et surtout des liens vers d'autres pages

Le type de contenu qui peut être intégré dans une page a évolué avec le langage HTML, qui en est actuellement à la version 5 (la première à permettre entre autres d'inclure directement des vidéos, sans passer par flash). Il existe également d'autres langages, comme le XHTML (le X étant pour *Extensible*) qui lui est très similaire.

### 2. Structure d'une page HTML

Le code source d'une page HTML a en principe au minimum une structure telle que la suivante :

```

<!DOCTYPE html>
<html>
  <head>
    <title>
      Exemple de HTML
    </title>
  </head>
  <body>
    Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
  <p>
    Ceci est un paragraphe sans hyperlien.
  </p>
</body>
</html>

```

La toute première ligne indique au navigateur la version du langage utilisée (ici HTML).

Ensuite commence le code proprement dit de la page. Tous les mots entre "<" et ">" sont les balises. Pour chaque balise qui ouvre une structure, il y a une balise qui la referme. Cette dernière porte le même nom précédé d'un slash : <head> ... </head>, <body> ... </body>, <a> ... </a>, etc.

On remarque que pour certaines balises comme <a>, il est possible d'ajouter des éléments (<a href="cible.html">). Certaines balises ne nécessitent pas d'être fermées par une autre balise; on écrit alors un slash à la fin de la balise, comme par exemple <link ... /> ou <img .../> que l'on verra plus bas.

On observe généralement une indentation du texte qui permet une lecture claire de la structure.

En particulier, tout le code est entre <html> et </html>, et on observe deux parties principales à l'intérieur : l'entête (<head>) et le corps (<body>).

## a. Entête

L'entête, délimitée par les balises <head> et </head>, contient les informations générales sur la page. C'est ici que l'on donne le titre de la page avec <title>. On peut ajouter d'autres éléments comme :

```

<meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
<link rel="icon" href="Docs/icone.gif"/>
<link rel="stylesheet" type="text/css" href="style.css"/>
<script type="text/javascript" src="java.scripts" ></script>

```

<meta> permet d'indiquer plusieurs choses comme l'auteur, le type de codage des caractères, etc.

<link> permet de donner des adresses de fichiers qui peuvent être l'icône de la page, contenir le style de la page (que l'on verra plus tard), etc.

<script> permet de donner l'adresse d'un fichier (ou directement le texte), contenant un code (ou script) de nature différente du html qui peut être utilisé dans la page, en particulier du code javascript.

Il en existe d'autres.

## b. Corps

Le corps, délimité par les balises <body> et </body>, est la partie qui décrit le contenu de la page. C'est donc en principe la partie la plus importante dans le code.

Ici les balises indiquent la structure de la page.

```

<h1>Titre 1</h1>
  <h2>Sous-titre 1</h2>
  <ul>
    <li>Premier item d'une liste</li>
    <li>Second item d'une liste</li>
  </ul>
  <p>Un paragraphe</p>

```

Il existe beaucoup de balises. Principalement :

<h1>, <h2>, ... permettent de mettre des titres hiérarchisés (<h1> étant le plus grand, <h6> le plus petit).

<ul> et <li> permettent de faire des listes. Il existe différentes sortes de listes (numérotées, à puce, etc) et donc de balises.

Et d'autres encore (tableaux, formulaires, vidéos...)!

Il faut tout de même citer deux autres balises parmi les plus communes. Le code permettant de faire un lien vers une autre page est le suivant :

```
<a href="adresse de la page">Texte sur lequel on clique</a>
```

L'adresse de la page (ou du document) peut être **absolue** (`http://...`) ou **relative** (`Documents/...`).  
On ajoute une image ainsi :

```

```

Et l'on peut tout à fait rendre une image « cliquable » en utilisant href et en insérant une image plutôt qu'en écrivant du texte entre les balises `<a>` et `</a>`.

### 3. Construction d'une page

Le code d'une page Internet, écrit en HTML, s'écrit dans un simple éditeur de texte (plus simple qu'un logiciel de traitement de texte) tel que notepad++ qui permet la coloration syntaxique. Le fichier créé a en général pour extension .html (ou parfois .htm).

Il est toujours possible d'enregistrer une page d'une site Internet sur son ordinateur à partir du navigateur (avec « Enregistrer sous... »). On peut alors en général voir le contenu (mais pas toujours complet) de la page enregistrée, même sans être connecté.

On peut s'aider de la documentation trouvée sur de nombreux sites pour créer une page Internet et connaître entre autres toutes les balises. Il y a beaucoup de sites en anglais, en particulier les sites officiels des langages (HTML, XHTML), comme par exemple <http://www.w3.org/TR/html4/> pour le HTML4. Le W3C (World Wide Web Consortium) est une communauté de personnes qui travaillent ensemble au développement de standards du Web, c'est lui qui définit les standards des langages pour les pages Internet.

On trouve évidemment des pages en français, comme <http://www.aliasdmc.fr/courshtml/>

On peut toujours faire des recherches sur Internet et trouver d'autres sites qui traitent des points précis ou expliquent autrement certaines choses.

#### **Exercice 2**

Créer un fichier HTML avec une extension .html qui pourra être ouvert par le navigateur.

Le contenu, qui sera à poursuivre tout au long de l'année, contiendra des informations générales à votre propos et parler de votre travail en ISN.

Il sera en particulier bien de prévoir le téléchargement de documents créés durant l'année (documents de recherches, diaporamas, programmes).

Penser à l'organisation des fichiers. On pourra par exemple créer un dossier nommé « Documents » qui contiendra les divers fichiers proposés au téléchargement et les images utilisées sur la page.

Les adresses vers les fichiers personnels données dans href devront en principe être relatives.

Pour que la page soit accessible sur Internet, il faut qu'elle soit disponible sur un réseau, c'est à dire que le fichier .html et les documents qu'il utilise soient copiés sur un serveur qui communique avec l'extérieur.

Il existe un cas particulier de nom de fichier html : "index.html". Si un dossier contient un fichier à ce nom, il est possible dans le navigateur de mettre l'adresse du dossier sans préciser le nom du fichier. Autrement dit une adresse "http://www.mapage.com/" correspond à l'adresse "http://www.mapage.com/index.html".

#### **Exercice 3**

Nous allons utiliser un compte créé spécialement pour la page Internet ISN du lycée.

Aller sur le site <http://ftpperso.free.fr/> et entrer les informations de connexion fournies lors du cours.

Ensuite il est possible de créer des dossiers et d'envoyer des fichiers.

Copier les fichiers .html créés précédemment vers le serveur, et aller sur le site Internet pour vérifier.

Faire en sorte que d'éventuels dossiers de documents contiennent nécessairement un fichier index.html (ou éventuellement une redirection vers une autre page).

#### **Exercice 4**

Créer, par groupe, une page qui présente le groupe et contient des liens vers les pages personnelles.

Vous pouvez mettre des images, mais utilisez uniquement des créations personnelles.

Placer ensuite les nouveaux fichiers sur le serveur.

On pourra vérifier la bonne forme du code des pages créées avec le site de validation <http://validator.w3.org/>.

# III. Apparence (CSS)

---

## 1. Généralités

Si l'on reste sur le type de code HTML indiqué jusqu'ici dans ce cours, on obtient des pages très simples et peu élégantes. Il est possible d'avoir la main sur la présentation de chaque élément affiché sur la page, autrement dit de donner un style à la page. Et le style fait beaucoup !

**Exemple** Aller sur une page Internet et choisir « Affichage > Style de la page > Aucun style ».

Le style se définit grâce au CSS (*Cascading Style Sheets* ou « Feuilles de style en cascade »). C'est aussi un standard défini par le W3C qui date des années 1990, et qui a été pris en charge par les navigateurs vers les années 2000. Il existe là aussi différentes versions, la dernière en construction étant la 4.

Le but de la spécialité ISN n'étant pas de traiter ce problème, nous n'entrerons pas dans le détail. Nous nous contenterons éventuellement de prendre des fichiers existant et de les adapter, sans chercher à compliquer la page.

## 2. Lier le fichier HTML et le style

Tout d'abord, il faut indiquer, en entête du fichier HTML, où se situe la description du style. Par exemple :

```
<link rel="stylesheet" type="text/css" href="style.css"/>
```

**Remarque** Il est possible de récupérer sur presque toutes les pages Internet actuelles des fichiers de style CSS. Il suffit de regarder le code source et de cliquer sur les noms des fichiers.

Ensuite, dans le code HTML on peut entourer les diverses zones de la page d'une « boîte » (ou « div ») auxquels ont associé un nom (« id ») ou une classe (« class »). Une identité ne peut être donnée qu'à un seul élément, mais une classe peut être partagée par plusieurs éléments. Cela permet d'avoir des styles différents selon l'identité ou la classe. Par exemple :

```
<div id="page">
  <div id="gauche">
    <div id="photo">
      <a href="Docs/Image.jpg" title="Agrandir">
        </a>
    </div>
    <div id="menu">
      <h2 class="c1">Menu normal</h2>
      <h2 class="c3">Menu spécial</h2>
    </div>
  </div>
  <div id="droite">
    <h1>Titre 1</h1>
    <h2>Sous-titre 1</h2>
    <ul>
      <li>Liste de description :
        <dl>
          <dd>item 1</dd>
          <dd>autre item</dd>
          <dd>dernier item</dd>
        </dl>
      </li>
    </ul>
  </div>
</div>
```

### 3. Le langage CSS

Globalement on peut retenir qu'il est possible de définir des attributs (taille, couleur, etc) à tout élément (titre, texte simple), en partie pour le corps (body) :

```
body{
background-color:#f8f8d6;
color:#1c1c1c;
}
```

mais aussi pour tous les autres éléments :

```
h1{
text-align:center;
color:#101e2e;
font-size:x-large;
text-transform:uppercase;
border-top:2px dotted #101e2e;
padding-top:0.5em;
}

p{
font-size:medium;
margin:0 0 0 1em;
}

a{
color:#0003ac;
text-decoration:none;
background-color:transparent;
}

a:hover{color:#8da0ff;}
a:visited{color:#b74cbb;}
```

Les dernières lignes permettent de donner des couleurs différentes selon que la souris est sur le texte ou que le lien a déjà été visité.

Si l'on veut définir des attributs communs à plusieurs éléments, on liste les différents éléments en séparant par des virgules :

```
h2, h3{
color:#103e2e;
}
```

On peut définir des styles différents selon l'identité (id) ou la classe (class) d'un élément (voir l'exemple de corps de page html donné plus haut). Sur une page il ne peut y avoir qu'un élément ayant une identité donnée, alors que les classes peuvent être partagées par plusieurs éléments.

Ce qui concerne les objets ayant l'identité page est introduit dans le fichier css par un #page (le dièse indique une identité) :

```
#page{
margin:1% 1%;
color:#000;
background-color:#ffffbe;
border:3px solid #000;
height:700px;
}
```

Ce qui concerne les objets ayant la classe c1 est introduit par un .c1 (la classe est indiquée par un point) :

```
.c1{
color:#655f4b;
background-color:transparent;
}

a.c1{
color:#b00606;
display:block;
font-weight:bold;
margin-left:-3em;
padding-left:4px;
text-decoration:none;
}
```

Un élément ayant cette classe c1 aura une certaine couleur (#655f4b). Un lien (<a>) dans un élément de classe c1 aura une couleur particulière (#b00606), mais le fond (background-color) restera le même que celui défini pour la classe c1 (c'est le phénomène de cascade : la définition la plus générale reste la même pour les cas particuliers tant qu'elle n'est pas modifiée).

On peut également définir des attributs différents pour des éléments de même type mais ayant une identité (ou une classe différente) :

```
#menu .c1{
color:#444;
}

#droite .c1{
color:#bbb;
font-size:medium;
}
```

L'apparence d'un élément de classe c1 n'aura pas la même apparence dans menu que dans droite.

Des éléments d'explications (en français) se trouvent ici : <http://www.zonecss.fr/courscss/>  
Pour les couleurs (par exemple) : [http://www.createafreewebsite.net/color\\_chart.html](http://www.createafreewebsite.net/color_chart.html)

**Exercice 5** Modifier la page HTML précédemment créée pour permettre l'utilisation d'un style, en particulier ne pas oublier d'ajouter dans l'entête du fichier html l'adresse du fichier css.

Créer ensuite le fichier css. Pour cela on pourra utiliser des fichiers css trouvés sur Internet (par exemple dans des tutoriels, ou des pages connues), n'en garder que le strict nécessaire à la page créée.

## IV. Pour aller plus loin

---

### 1. Certification

Les langages HTML, XHTML, CSS étant définis précisément, ils doivent respecter une syntaxe précise. Les navigateurs ne sont pas toujours très difficiles et affichent des pages même mal écrites.

Il existe des outils sur Internet pour vérifier que le code que l'on a créé est parfaitement bien formé.

pour le html et le xhtml : <http://validator.w3.org/>

Pour le css : <http://jigsaw.w3.org/css-validator/>

Cela donne une garantie que des erreurs d'affichage ne se feront pas à cause d'un mauvais code.

**Exercice 6** Tester des pages web, vérifier que l'on en trouve qui sont mal écrites (ou en tout cas ne respectent pas le standard), même parmi les plus célèbres (google, wikipedia). Testez votre page!

Malheureusement, tous les navigateurs ne réagissent pas de la même manière à un code donné. Il existe des outils permettant de savoir comment l'ensemble des navigateurs réagissent sans avoir à les installer.

Par exemple : <https://browserling.com/>

**Exercice 7** Faire afficher quelques pages dans ce site en choisissant différents navigateurs.

## 2. Pages dynamiques

En dehors de la vidéo et des applications en flash qui donnent une impression de dynamisme à la page, il y a des outils très puissants qui permettent d'améliorer la qualité de navigation sur Internet.

En fait, la plupart des pages sont ce que l'on appelle « dynamiques » : elles s'adaptent en fonction de paramètres (nom ou pays de l'utilisateur, mots clés introduits dans des formulaires, etc.) Ce qui fait que les pages ne sont pas écrites directement en html, qui ne permet pas ce dynamisme : un fichier html est figé et affiche toujours la même chose.

Enregistrer la page google qui s'affiche sur votre ordinateur ne vous donnera pas le code d'un moteur de recherche personnel !

### a. PHP

Un langage fréquemment utilisé pour faire des pages dynamiques est le PHP (PHP : Hypertext Preprocessor, c'est un acronyme récursif).

C'est en fait un langage (algorithmique) qui est interprété par le serveur de la page web. Cela signifie que le serveur exécute l'algorithme, la sortie de l'algorithme étant un code html qui est envoyé au navigateur de la personne qui regarde le site.

Le code du fichier php n'est jamais visible par l'utilisateur. Seul l'administrateur du site peut le voir et le modifier.

Sans php, il serait par exemple très laborieux (mais pas infaisable ?) de faire un site de vente par Internet. Il faudrait créer une page par produit vendu (si raisonnablement on ne souhaite en afficher qu'un), alors qu'il suffit d'une seule page php bien faite pour parcourir l'ensemble des produits (on ajoute un moyen de chercher et d'afficher un produit donné). C'est un gain de temps et de mémoire. La gestion du nom du client est raisonnablement impossible en html seul.

PHP n'est pas utilisé seul, il est en général associé aussi à des bases de données qui contiennent les listes de produits et de clients avec leurs mots de passe, etc.

### b. Javascript

D'autres éléments permettent une dynamique à la page affichée. Il s'agit par exemple de javascript. Cela permet entre autres de modifier l'apparence ou même des éléments de la page en fonction d'événements (clic, heure). Un des effets d'un script javascript peut être par exemple de charger une autre page Internet au bout d'un certain temps.

Afficher les résultats d'une recherche avant même d'avoir fini de taper tous les mots clés est une autre possibilité.

Le monde des pages Internet s'est considérablement enrichi depuis ses débuts !