

Le HTML

Sommaire

Introduction.....	3
I. Les balises HTML	4
A. Définition.....	4
B. Rôle	4
C. Écriture des balises.....	4
D. Les balises d'entêtes.....	6
II. La balise <head>	10
A. La balise <title>.....	10
B. Les balises <meta>.....	11
C. Le code CSS.....	13
D. Le code Javascript.....	13
E. Première page.....	13

Crédits des illustrations :
© DR.

Les repères de lecture



Retour au chapitre



Définition



Objectif(s)



Espace Élèves



Vidéo /
Audio



Point important /
À retenir



Remarque(s)



Pour aller
plus loin



Normes et lois



Quiz

Introduction

Ce cours va vous permettre de créer vos premières pages HTML et de les visualiser dans un navigateur. Pour construire une page HTML, il vous faut un éditeur. Dans ce cours, nous utiliserons Notepad++ gratuit et téléchargeable sur le net.

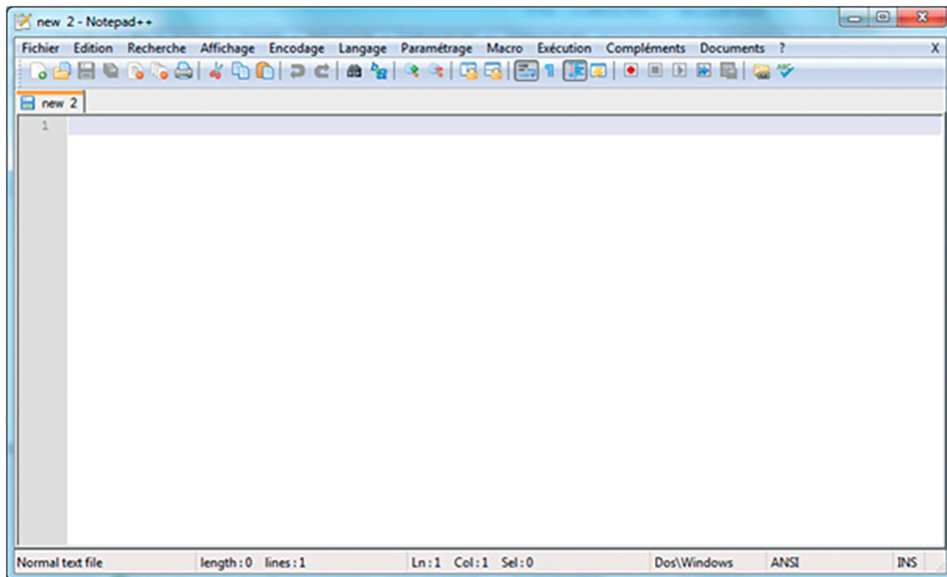


Fig. 1

I. Les balises HTML

A. Définition

Les balises sont des instructions qui délimitent un contenu (un paragraphe, une image, une vidéo, etc.). Elles permettent au navigateur d'afficher les éléments se trouvant à l'intérieur, selon des critères définis.

B. Rôle

Les balises servent à structurer le contenu de vos pages web, c'est-à-dire ordonner de façon cohérente les objets contenus dans votre page.

La notion d'objet est très importante en informatique et dans le Web en particulier.

Dans une page HTML, chaque élément HTML représente un objet qui peut contenir d'autres objets, et chaque objet a des propriétés qui lui sont propres.

***Exemple :** un paragraphe pourra avoir une largeur et une hauteur, ce qui ne sera pas le cas pour une chaîne de caractères. Pour cette dernière, une propriété pourrait être la dimension du corps de la police de caractères.*

C. Écriture des balises

Elles commencent par le signe « < » et se terminent par le signe « > ». Dès que le navigateur voit le signe « < », il se prépare à afficher quelque chose.

Aucun espace ne doit figurer dans le nom des balises. Toutes les balises doivent s'ouvrir et se fermer pour indiquer le début et la fin d'une commande.

1. Balises fermées

En général, la majorité des balises en HTML s'ouvrent et se ferment, toutefois nous verrons qu'il existe des exceptions.

Exemple : `Ceci est un texte important en gras`

Dans ce cas, `` est la balise ouvrante et `` est la balise fermante.

a. Balises auto-fermantes

Les exceptions dont je parlais concernent les balises qui ne contiennent rien à part elle-même.

Exemple 1 : une image

```

```

Exemple 2 : une ligne horizontale

```
<hr>
```

2. Les attributs

Les attributs sont divers et variés et se placent dans la balise ouvrante dans le cas des balises fermées et dans la balise elle-même pour une balise auto-fermante.

Exemple 1 : `<h1 id="titre_principal">Ceci est un texte de couleur rouge</h1>`

Exemple 2 : `<hr class="clear">`

Ils servent à modifier l'aspect initial du contenu d'une balise, nous verrons par la suite de quelle manière.

a. Notion de relation filiale ou relation parent-enfant

En HTML et nous verrons que cela est vrai aussi pour le CSS, les balises s'imbriquent les unes dans les autres en suivant une règle bien précise.

```
Ex :  
<p>  
<span class="bleu"><strong>texte en bleu et pertinent en gras</strong></span>  
</p>
```

Fig.2

Dans ce cas, on dira que la balise `<p>` est le parent des balises `` et `` et que `` est l'enfant de la balise `` qui est elle-même l'enfant de la balise `<p>`.

Cette notion qui sera développée plus loin dans ce cours est primordiale car elle est déterminante pour une bonne utilisation des règles CSS car un enfant peut hériter des propriétés de son parent.

Enfin, comme je l'ai déjà dit précédemment, il sera indispensable que vous connaissiez les balises principales car si l'exemple précédent avait été écrit comme ceci :

```
<span class="bleu">  
<strong>  
<p>  
texte en bleu et pertinent en gras  
</p>  
</strong>
```

Fig.3

Nous aurions quelque chose qui n'est pas valide même si votre texte s'affiche dans le navigateur car des éléments dit « inline » encadrent des éléments dit de « bloc » (nous en explorerons plus tard toute la portée).

3. La balise HTML

La première instruction que nous devons donner au navigateur, est qu'il va lire du HTML.

Nous allons utiliser la balise `<html></html>`.

`<html>` est l'ouverture de la balise : pour indiquer au navigateur qu'il commence à lire du HTML

`</html>` est la fermeture de la balise : pour indiquer au navigateur qu'il termine de lire du HTML

Toutes vos pages commenceront par `<html>` et se termineront par `</html>`. A l'intérieur de ces balises, d'autres viendront s'imbriquer pour créer le contenu de la page.

Il faut donc une grande rigueur dans l'écriture de votre code.

La page HTML est séparée en deux parties : l'en-tête et le corps de la page. L'entête est tout ce qui concerne le DOCTYPE et la balise `<head>`, le corps de la page est tout ce qui concerne la balise `<body>`.



Fig.4

D. Les balises d'entêtes

1. Code source

Pour regarder le code source d'une page HTML, on va se servir d'un inspecteur de code sur le modèle du défunt Firebug, célèbre module complémentaire de Firefox.

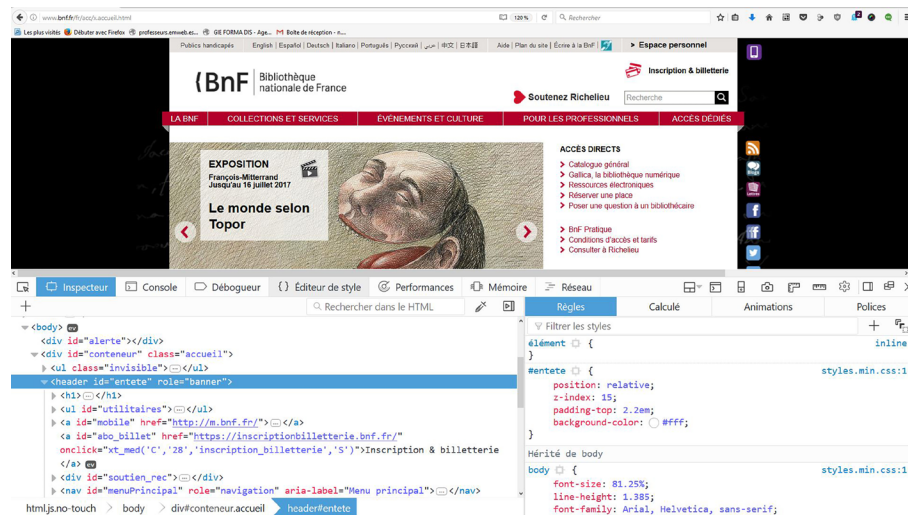


Fig.5

Il suffit de cliquer sur F12 ou aller dans Outils > Développement web > inspecteur pour Mozilla.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1">
<body id="home">
</body>
</html>
```

Fig.6

La première ligne est très importante car elle précise au navigateur de quelle façon il doit interpréter le reste des balises. On l'appelle le Doctype du document.

2. Les DOCTYPES

Pourquoi un DOCTYPE ?

Dans les normes du HTML et du XHTML, un **DOCTYPE** (contraction pour « Document Type Declaration ») informe le validateur de la version de (X)HTML que vous utilisez, et doit apparaître en première position dans chaque page web.

Les DOCTYPEs sont essentiels pour que les documents web s'affichent et fonctionnent correctement. En cas de DOCTYPE non conforme les navigateurs passent en mode « Quirks » (« habitudes bizarres ») qui va considérer votre balisage comme invalide et démodé. Les DOCTYPEs sont donc préconisés par le W3C.

Source: <http://www.w3.org/QA/2002/04/valid-dtd-list.html>

Depuis le déploiement du HTML5, on peut utiliser sans risque le nouveau DOCTYPE reconnu par les principaux navigateurs et qui lira correctement les versions de langages plus anciennes.

On utilisera donc le DOCTYPE suivant:

```
<!DOCTYPE html>
```

Auparavant les DOCTYPES avaient des formes différentes suivant que l'on parlait de XHTML, HTML4.01, etc..., à ceci s'ajoutait les trois versions existantes de ces langages (Strict, Transitionnel, Frameset). Voici ce que cela donnait par exemple pour le XHTML 1.0 Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```



Dans les premiers temps nous ne coderons pas spécifiquement en HTML5 afin de ne pas multiplier le nombre de balises et d'assimiler petit à petit les plus communes.

Le DOCTYPE HTML5 nous permet de simplifier l'écriture du code mais ne nous interdit nullement de coder comme avant. D'autre part, tous les sites ne sont pas encore passés au HTML5, loin s'en faut, il faut donc pouvoir maintenir un site écrit dans une version précédente du langage.

Vous remarquez comment le DOCTYPE du HTML5 va nous permettre d'éviter bien des soucis. En outre, toutes les anciennes balises HTML sont prises en compte par le HTML5 à l'exception de quelques balises qui sont désormais qualifiées d'obsolètes ou de dépréciées, telles ``, `<center>`, `<frame>`, etc.

3. Règles de syntaxe

a. Les 4 règles de syntaxe du HTML

Toute balise ouvrante doit obligatoirement être fermée.

Juste:

```
<span>xhtml</span>
```

Faux:

```
<span>xhtml<span>
```

Les valeurs des attributs doivent être entre guillemets ou apostrophes.

Juste:

```
<span id="monSpan" class='spanXhtml'>xhtml</span>
```

Faux:

```
<span id=monSpan class=spanXhtml>xhtml</span>
```

Chaque attribut doit avoir une valeur.

Juste:

```
<input type="checkbox" checked="checked" >
```

Faux:

```
<input type="checkbox" checked >
```

Chaque élément doit être imbriqué correctement.

Juste:

```
<p>j'aime le <strong>xhtml</strong></p>
```

Faux:

```
<p>j'aime le <strong>xhtml</p></strong>
```


Il faut cependant faire très attention à la différence entre contenu HTML et mise en forme par les CSS. Le XHTML strict exigeait déjà un tel cloisonnement. Voyons rapidement quelles sont les règles des DOCTYPES plus anciens.

b. XHTML 1.0 Strict

Même règles que pour le HTML mais les balises et attributs doivent être écrits en minuscules.

Juste :

```
<a href="http://www.esecad.fr" >j' aime l' ESECAD</a>
```

Faux :

```
<A HREF="http://www.esecad.fr" >j' aime l' ESECAD</A>.
```

- il n'accepte pas d'utiliser l'attribut « target » sur les balises <a>
- il n'accepte pas d'utiliser la balise <iframe>

c. XHTML 1.0 transitional ou Framset

Même règles que XHTML 1.0 Strict mais accepte les balises et les attributs agissant sur la présentation de la page web.

Exemple : , <center>...

4. HTML et sémantique

La sémantique, c'est quoi ?

La sémantique, c'est le sens donné à un élément de la page web.

Ici, je ne donne aucun sens :

```
<div>Chapitre 1 : Le xhtml et la sémantique</div>
```

```
<div>Le xhtml doit donner du sens !</div>
```

Maintenant, il y a un sens :

```
<h2>Chapitre 1 : Le <abbr>xhtml</abbr> et la  
sémantique</h2>
```

```
<p>Le <abbr>xhtml</abbr> doit donner du sens !</p>
```

Votre travail doit donc être non seulement de produire un code HTML-CSS permettant d'obtenir un affichage correct, mais également d'attribuer aux éléments de la page des balises appropriées en termes de sémantique.

La sémantique, pour quoi faire ?

D'abord, pour améliorer l'accessibilité : un appareil de synthèse vocal saura ainsi que tel élément est le titre de la page, celui-là, le sous-titre n° 1 de la page et ce texte, le premier paragraphe du sous-titre n° 1... Il saura aussi que ces 4 lettres sont une abréviation et que ce mot est un mot à définir suivi de sa définition.

Ensuite pour améliorer son référencement : un moteur de recherche n'est pas intelligent (enfin pas encore !) et ne sait pas ce que tel ou tel élément de la page représente. Entourer un élément par une balise spécifique indique au robot ce que représente cet élément. Le moteur ne pourra que mieux comprendre la structure et le sens de vos pages web.

5. HTML et accessibilité

Le **HTML** permet d'améliorer **l'accessibilité des sites internet**.

L'accessibilité, c'est quoi ?

L'accessibilité, c'est la capacité à accéder à un site Internet via n'importe quel logiciel capable de naviguer sur un site web : les navigateurs web traditionnels (Firefox, Internet explorer, Safari...) mais aussi les appareils de synthèse vocale utilisés par les handicapés, les navigateurs des téléphones mobiles, le navigateur de mon frigidaire (qui me proposera d'acheter des œufs en ligne quand je serai à cours), ou le navigateur intégré au tableau de bord de ma voiture (qui peut-être dans un futur proche me proposera une réservation en ligne d'une chambre de l'hôtel le plus proche)...

L'utilité du HTML, est de réaliser un site et un site seulement, de manière à ne pas décliner autant de sites qu'il existe de logiciels capables de naviguer sur un site internet.

II. La balise <head>

Rappel : La balise head contient tout ce que vous ne voyez pas sur une page, dont 3 éléments essentiels.



La balise <title> est le titre principal de la page web.

Il s'affiche en haut à gauche de la fenêtre de votre navigateur.

Exemple : <title>Le titre de mon site</title>

A. La balise <title>

Le contenu de la balise <title> est particulièrement important car lu en premier, par les robots des moteurs de recherche. Il doit donner le nom du site et refléter les concepts de la page.

Votre démarche doit être la suivante dans le choix des mots : quels sont les mots-clés les plus pertinents par rapport à mon contenu ?

C'est ce qui apparaît en premier dans les résultats de recherche des moteurs de recherche, c'est également ce qui est indiqué lorsque vous consultez votre historique de navigation, et c'est la première indication qui guide les aveugles sur le contenu de la page.

Pour appliquer ce que nous avons vu, enregistrez votre fichier sous « index.html » à l'aide de Notepad++ (ou TextWrangler ou Bluefish) et double cliquez dessus afin de l'ouvrir.



Fig.7

La page de base est prête à recevoir du contenu, voici à quoi elle devrait ressembler :

```
<!DOCTYPE html>
<head>
  <title>Le titre de ma page</title>
</head>
<body>
</body>
</html>
```

Fig.8



Commencez à vous organiser, créez un dossier **www** sur votre disque **C:** ; puis créez un sous dossier mon premier-site, puis à l'intérieur créez un sous-dossier img (dans lequel vous placerez les images qui seront utilisées dans vos pages) et un sous dossier __sources dans lequel vous placerez tous les fichiers de travail et ressources qui vous serviront à remplir votre site. (Ce dossier ne sert pas directement à la réalisation de votre site, c'est un dossier de travail, votre « fourre-tout » qui n'ira pas sur le serveur distant).

B. Les balises <meta>

Elles vont permettre d'indiquer certaines données et informations aux navigateurs web, moteurs de recherche ou autres outils d'indexation qui balayent vos pages.

La question de l'obsolescence des balises META est souvent posée. En matière de référencement, on conseille souvent de « renseigner les balises META » sans trop connaître le degré d'efficacité d'une telle pratique. Néanmoins, nous allons regarder de plus près les balises qui concernent l'encodage des caractères (indispensable), la description (conseillée) et les mots-clés.

Encodage des caractères :

Si vous avez un problème d'encodage, vous allez au-devant d'apparition de « hiéroglyphes » dans vos pages en remplacement des caractères accentués ou autres caractères rares. Afin, de ne pas aller au-devant d'une telle mésaventure, nous utiliserons la norme Unicode UTF-8 universelle, car elle peut prendre en compte tous les caractères, dont bien entendu les caractères français.

Attention l'écriture des balises <meta> diffère quelque peu entre le HTML5 et les versions précédentes, il faut surtout être cohérent.

Vous avez le choix entre une version courte :

```
<meta charset="utf-8">
```

Et une version longue (plus prudente):

```
<meta content="text/html; charset=utf-8"
http-equiv="Content-Type">
```

En Europe, on emploie également la norme ISO, ce qui donne:

```
<meta content="text/html; charset=ISO-8859-3"
http-equiv="contenttype">
```

Description

Cette META permet de fournir une description courte (deux à trois phrases courtes) de la page web. Cette description est notamment utilisée (dans certains cas) par les principaux moteurs de recherche, pour fournir le court texte d'aperçu des pages de résultats, ou bien par les réseaux sociaux (Facebook, Google+) pour définir le paragraphe de texte associé au lien partagé.

Pour rappel, voici un exemple de balise META Description:

```
<meta name="description" content="Tout sur les appa-
ritions muettes de la Banane Masquée dans le cinéma
colombien des années 20 et 30." >
```

On conseille généralement de ne pas utiliser la même description pour les différentes pages d'un site. Dans l'idéal, chaque page aura sa propre description habilement composée.

Voici quelques conseils pour la rédaction de cette description: elle doit être unique, et non dupliquée sur d'autres pages du site; elle doit refléter fidèlement le contenu de la page; elle doit être attractive pour l'utilisateur dont le regard « scanne » une page de résultats de moteur de recherche; elle doit reprendre les principaux mots-clés de la page et être plutôt brève (on recommande de ne pas dépasser 200 caractères, bien qu'il ne s'agisse pas d'une règle stricte).

On ne dispose pas toujours d'un contenu spécifique pour la META Description de chaque page. Dans ce cas, trois stratégies sont possibles:

- ne pas utiliser la balise META Description du tout;
- utiliser une META Description seulement pour certaines pages clés, ou uniquement pour la page d'accueil;
- extraire un contenu des contenus disponibles en base de données, par exemple le texte d'accroche pour une page d'article ou le début de la description du produit pour une page produit.

Notez qu'une page ne sera pas « pénalisée » par un moteur de recherche parce qu'elle n'utilise pas de META Description. Mais si vous n'en utilisez pas, faites attention à ne pas utiliser la balise du tout, et évitez les balises sans contenu:

```
<metaname="description" content="" ><!-- À éviter
-->
```

Dernière chose : fournir une META Description ne garantit pas qu'elle sera effectivement utilisée par les moteurs de recherche dans leurs pages de résultats!

C. Le code CSS

Soit directement dans la page entre les balises
`<style></style>`

Soit grâce à un lien vers la feuille de style externe

Exemple : `<link href="style.css" rel="stylesheet" media="screen">`

href = url de la feuille de style. Indispensable.

rel = type de lien (ici stylesheet signale qu'il s'agit d'un fichier de feuille de styles). Indispensable.

media = le type de media pour lequel le lien destiné (ici pour un écran). Facultatif mais fortement conseillé.

D. Le code Javascript

Soit directement dans la page entre les balises

`<script type="text/javascript"></script>`

Soit grâce à un lien vers une feuille Javascript externe

Exemple : `<script type="text/javascript" src="fancybox/jquery.mousewheel-3.0.4.pack.js"></script>`

E. Première page

Au final voici à quoi pourrait ressembler le code de notre première page :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
  <title>Une page html </title>
</head>
<body>
  Le code html du contenu de la page se trouve ici.
</body>
</html>
```

Fig.9

Désormais nous allons nous intéresser aux balises que l'on écrit entre les balises du corps de la page soit entre `<body>` et `</body>`.