

Aide exercice 1

Théorie et technique

1. HTML, un langage de balises

Le langage HTML est un langage dit de « structuration ou de balisage ». Il permet de formaliser la présentation d'un document afin qu'il puisse être interprété par un navigateur Web.

Ainsi, pour créer une ligne de code en langage HTML on va :

Ouvrir une balise avec un nom qui définit sa fonction, **taper notre texte** si besoin, **refermer la balise**.

```
<balise>texte</balise>
```

Les balises HTML fonctionnent par paires afin d'agir sur les éléments qu'elles encadrent, avec quelques exceptions : les balises « autofermantes » que nous étudierons prochainement.

NB : Attention, les balises ne sont pas sensibles à la casse (majuscule et minuscule). Par convention nous écrivons, nous coderons, tout en minuscule.

2. Structure d'une page HTML

La structure du document HTML peut être comparée au squelette du corps humain. Elle se décompose en plusieurs parties que nous pourrions comparer aux différentes parties qui composent notre corps. Cette structure doit être connue par cœur puisque toutes les pages HTML que nous consultons comportent systématiquement cette structure.

a) Les DTD ou doctype

Les DTD « déclaration du type de document » permettent d'indiquer aux navigateurs qui interprètent le document HTML, la version du langage HTML utilisé. Cela est directement corrélé avec les balises HTML que nous allons utiliser dans notre document, les fonctionnalités que nous allons employer, et la représentation graphique produite par le navigateur. La version 5 du langage HTML offre un nombre de balises supplémentaires qui n'existent pas dans la version 4.

Cette ligne de code est le premier élément qui introduit notre document HTML.

Exemple :

```
<!doctype html>
```

Le doctype ci-dessus est un doctype HTML5, il est plus simple à écrire et donne accès au plus grand nombre de fonctionnalités disponible dans les navigateurs à l'heure actuelle .

b) <html>

La balise « <html> » marque le début et la fin des écritures du document HTML. Cela signifie que notre document HTML commencera, après le doctype, par la balise « <html> » et finira par la balise

« </html> ». Il est possible, à l'intérieure de cette balise de spécifier la langue utilisé dans la page, anglais, français ou autre.

Exemple :

```
<html> <!-- la balise html nue -->
```

OU

```
<html lang="fr">
<!-- la même balise html qui définit la langue du document sur
français. D'autres valeurs sont disponibles : en, es, it... -->
```

NB : Au détour d'une de vos consultations sur le web, Google vous a probablement proposé de traduire certaines pages en langues étrangères... Comment Google est-il au courant ? Cela provient, en partie, de la définition de la langue à l'intérieure de la balise <html>

c) < head>

Le code écrit entre les balises « <head> » n'est pas directement visible à l'écran, c'est-à-dire dans la mise en page rendue dans le navigateur. Mais alors a qui sert cette balise et que place-t-on à l'intérieur ?

1. Des balises <meta> :

- Référencer la page

Ces balises « <meta> » définissent des informations sur le référencement de la page à destination des moteurs de recherches (balises keywords et description).

```
<head>
<meta name="description" content="Ce site parle de..." />
<meta name="keywords" content="ski alpin, montagne, neige" />
</head>
```

Ici, l'attribut « name » définit le type de meta employé, ici « keywords » et « description ». « content », définit le contenu associé à cette balise : « ski alpin », « ce site parle de »...

La balise « description » correspond à la description du site présentée par Google après une recherche. Généralement on définit 250 caractères maximum dans le contenu.

Google

www.google.fr/ ▾

Version française du moteur de recherche. Propose des outils et des services pour les internautes.

La balise « keywords » permet de référencer les mots clés qui peuvent être potentiellement tapés par les internautes, dans le moteur de recherche, afin d'accéder à votre page web. Pour définir ces mots clés nous pouvons employer 500 caractères séparés par une virgule, et l'équivalent de 8 à 10 mots clés

NB : Il existe une 15^{ème} d'autres balises <meta> qui ont peu de crédibilité auprès des robots qui indexent le contenu des pages mais que nous employons tout de même à la manière d'une 4^{ème} de couverture d'un livre.

<http://www.alsacreations.com/article/lire/628-balises-meta.html>

- Indiquer l'encodage des caractères

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

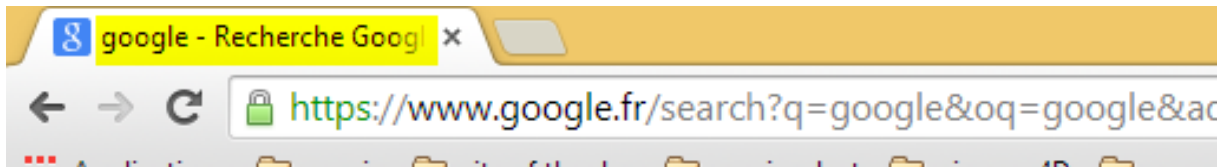
Le « charset » définit l'encodage des caractères employé dans le document HTML. En effet, le langage HTML a été à l'origine développé par les américains qui emploient un alphabet non-accentué. Comment étendre son domaine d'application aux alphabets occidentaux ou cyrilliques par exemple ?

Il était longtemps nécessaire de définir un encodage de caractère spécifique dans le « charset », propre à chaque langue. Désormais l'encodage « utf-8 » permet d'employer la quasi totalité des caractères universels de chaque langue.

- Le titre du document

La balise <title> définit le titre de la page web, visible dans l'intitulé de la fenêtre du navigateur.

```
<title>Le titre de ma page</title>
```



d) <body>

Le code placé entre ces 2 balises sera directement visible à l'écran. C'est donc ici que nous allons coder le contenu visible du site : textes / images au travers d'une mise en page.

e) Structure complète du document HTML

Voici la structure complète, standard, d'un document HTML. Cette structure sera toujours la même.

```
<!doctype html>
<html lang="fr">
<head>
  <meta name="description" content="description du site" />
  <meta name="keywords" content="mot clé 1, mot clé 2" />
  <meta charset="utf-8" />
  <title>Le titre de ma page</title>
</head>
<body>
</body>
</html>
```

3. Insérer du contenu image

Quelques images sur un site Web peuvent le rendre plus attractif et plus convivial, cependant il est important de ne pas sombrer dans l'excès car les images peuvent impliquer un temps de chargement assez long et peuvent dans certains cas nuire à la lisibilité.

La balise ``, qui est une balise autofermante, permet d'insérer des images dans une page HTML

Exemple :

```
<img />
```

L'image peut être située sur le même serveur que la page en cours de lecture mais peut également provenir d'un autre serveur en spécifiant son URL complète (on parle d'URL absolue et relative).

```
<!--Image située sur le même serveur ou dans le même dossier -->  
  
<!--Image située sur un serveur externe -->  

```

Trois formats d'image peuvent être représentés dans un navigateur :

- le format **GIF (déprécié)**, il est utilisé pour des images contenant peu de couleur. En effet le nombre de couleur dans le .gif est de 256 dont une couleur transparente. Il est déprécié aujourd'hui, ou uniquement utilisé pour produire certains effets d'animations avec sa version .gif animé...
- le format **JPG**, est utilisé pour des images contenant beaucoup de couleur, une photo par exemple. Le .jpg doit être au format colorimétrique RVB.
- le format **PNG**, concurrent du GIF contient plus de couleurs. Cependant, il reste encore peu utilisé.

Des options et des fonctionnalités (facultatives) peuvent être ajoutés en utilisant les attributs HTML de la manière suivante :

```
<img attribut="valeur" attribut2="valeur" />
```

La liste des attributs :

alt : Permet d'afficher un texte alternatif lorsque l'image ne s'affiche pas.

title: Permet d'afficher une infobulle lors du survol de l'image par le curseur.

width: Permet de spécifier la largeur de l'image.

height: Permet de spécifier la hauteur de l'image.

4. Le viewport d'un appareil mobile

Le « viewport » désigne schématiquement la surface de la fenêtre du navigateur. Cependant, la notion de « viewport » sur un appareil mobile est différente de celle sur un écran de bureau : sur

mobile, le navigateur ne dispose pas de "fenêtre" réelle, ni de barres de défilement car tout est prévu pour naviguer de manière tactile.

Comprendre la notion de viewport est absolument indispensable dans un projet d'intégration de site web pour tablettes et smartphones, ou dans un esprit d'adaptation "Responsive Web Design".

Lorsque nous développons un site web pour les appareils mobiles et les tablettes, il convient de définir le « viewport » comme étant la totalité de l'écran du dispositif de consultation. Pour se faire nous allons rajouter une balise <meta>

```
<head>
  ...
  <meta name="viewport" content="width=device-width,
initial-scale=1.0"><meta name="viewport"
content="width=device-width, initial-scale=1.0">
  ...
</head>
```

5. Les sections sémantiques

Les sections sémantiques permettent de découper un document en plusieurs grandes parties. Cette découpe n'a pas nécessairement une vocation de mise en page graphique, mais plutôt une vocation sémantique.

Plus simplement il s'agit de définir dans le corps de page <body>, les différentes parties de notre mise en page. L'en tête, la navigation, le pied de page, une grande zone thématique, un article....

Nom	Détails
<section>	Permet de définir une zone de contenu ayant une même thématique. Il s'agit généralement du corps de texte.
<article>	Il s'agit d'un article texte
<nav>	Le menu, le liens de navigation principaux
<header>	Section d'introduction de la page (en-tête de page). Cette partie peut avoir une vocation graphique uniquement
<footer>	Conclusion ou pied de page.

```
<body>
  <header> ... </header>
  <section> ... </section>
  <footer> ... </footer>
</body>
```

6. La hiérarchisation des contenus : niveaux de titres, paragraphes et retours chariots

Les niveaux de titres sont représentés à l'aide des balises <h1> à <h7>. Une page peut avoir jusqu'à 7 niveaux de titres différents. Le niveau 1, <h1> étant associé au niveau le plus fort sémantiquement. Pour définir les différents niveaux, nous utilisons donc <h1>, <h2>, <h3>...

Par défaut les navigateurs accordent graphiquement des tailles différentes à ces niveaux de titres en fonction de leurs importances. <h1> est donc le plus grand, <h7> le plus petit.

A noter que ces balises sont très importantes pour le référencement de la page par les moteurs de recherche.

```
<h1>Titre 1</h1>
```

Par défaut le reste des contenus textes, le corps de texte est organisé en paragraphe, exactement comme nous pourrions y faire référence dans support papier traditionnel. Pour se faire nous utilisons la balise <p>

```
<p>paragraphe 1</p>
<p>paragraphe 2</p>
```

Pour forcer un retour à la ligne nous pouvons utiliser la balise
 de la manière suivante.

```
<p>Un texte un peu long avec un changement d'idée<br />
La fin de mon texte ou une autre idée</p>
```

7. Les types de balises « block » et de type « inline »

En HTML, on distingue 2 types de balises.

- **Type « block »** : Ces balises prennent l'intégralité de la largeur écran disponible. Cela signifie que le contenu, les balises qui suivront passeront automatiquement à la ligne. Cela peut parfois prêter à confusion car les balises employées pour structurer du texte (<h1>, <p>, <div>), sont de types blocks, alors qu'elles accueillent généralement du texte qui dans notre esprit se place en ligne.

- **Type « inline »** : signifie que la progression du contenu après ces balises se fait en ligne. C'est le cas des balises qui traitent principalement la typographie <i>...

Exemple :

Mon titre de niveau 1

Mon texte **en gras** balise et en *italique* balise <i> progresse en ligne

- Le niveau de titre 1 `<h1>` est de type « block », il prend l'intégralité de la largeur disponible. Le reste des balises de types « inline » se placent les unes à la suite des autres.

8. La nature des balises : sémantiques ou génériques

Dans le langage HTML, il existe 2 natures de balises, les balises sémantiques et les balises génériques. Nécessairement ces balises sont de type « block » ou de type « inline » en ce qui concerne leur représentation graphique. Ce qui est invisible mais pourtant bien présent dans votre document (et qui impactera considérablement le référencement), c'est la sémantique associée à cette balise et au contenu que vous encadrez avec.

La plupart des balises qui permettent de hiérarchiser votre document (`<h1>`, `<h2>`, `<p>`, `<header>`, `<footer>`, `<section>`), sont des balises sémantiques.

Toutefois il existe d'autres balises qui sont simplement employées pour construire la présentation graphique du document, le design, la mise en page. De ce fait il n'est pas nécessaire qu'elles aient une vocation sémantique. On parle de balises « génériques ».

Il s'agit entre autre de 2 balises : `<div>` et ``

La balise `<div>` est de type « block » et la balise `` est de « inline ».

En définitive il serait possible de construire l'intégralité d'un site web en utilisant ces 2 balises. C'est malheureusement ce qui arrive bien souvent. Dans ce cas de figure, bien que la représentation graphique puisse être correct, le document n'a aucun sens, et sera mal référencé.

9. Embarquer du CSS dans une balise HTML

Pour embarquer du CSS nous utilisons toujours un attribut HTML qui s'intitule « style ». Cet attribut se place systématiquement dans la balise ouvrante.

Ce attribut se place à l'intérieur de n'importe quelle balise, un paragraphe, un span, un niveau de titre... La valeur de cet attribut sera composée du « nom de la propriété CSS », suivie de « sa valeur ».

Ex : ajout de CSS dans une balise de niveau de titre 1

```
<h1 style="les propriétés CSS"> contenu </h1>
```

Ex : mis en pratique

```
<p style="color:red;"> contenu </p>
```

La logique veut que l'on utilise les propriétés CSS en corrélation avec le type de balise. Par exemple, des propriétés de typographies avec des balises de niveaux de titres, de paragraphes...

Ainsi, il est possible de coupler plusieurs propriétés CSS de la manière suivante :

```
<p style="color:red;font-size:1em;"> contenu </p>
```

Une liste des propriétés CSS pour le traitement graphique des typographies est accessible à cette adresse : m2.ecoledunet.com/cours/css_typo.pdf

10. Définir une fonte pour l'ensemble de notre document HTML

Pour définir une fonte, nous devons employer la propriété CSS « font-family », sa valeur est simplement le nom de la font.

Ex :

```
<p style="font-family:Verdana;" > Mon contenu </p>
```

Dans cet exemple la typographie « Verdana » sera utilisée pour un paragraphe.

Si l'on souhaite associer une typographie à l'ensemble du document, nous placerons notre déclaration CSS dans <body>. Ainsi toutes les balises à l'intérieur de <body> hériteront de cette définition CSS.

Ex :

```
<body style="font-family:Verdana;" >
  tout le contenu de la page
</body>
```

Attention, il ne s'agit pas de rajouter une balise <body>, car celle-ci est unique dans le document, mais d'intégrer des propriétés CSS à bord de cette balise.

Par mesure de sécurité, au cas où la fonte ne serait pas disponible, il est conseillé de prévoir des typographies de secours. Nous pouvons procéder de la manière suivante :

Ex :

```
<body style="font-family:Verdana, Arial, Tahoma;" >
  tout le contenu de la page
</body>
```

Une liste des typographies « web safe », c'est-à-dire présente sur tous les systèmes d'exploitation est disponible à cette adresse :

<http://romy.tetue.net/list-of-web-safe-fonts>

11. Redéfinir le type d'une balise

Pour redéfinir le type d'une balise, nous utiliserons la propriété CSS « display », avec pour valeur son type.

Ex :

```
<img style="display:block" />
<!--Ou -->
<h1 style="display :inline">...</h1>
```


12. Proportions d'une image

Il existe principalement 2 unités de mesures qui permettent de déterminer la taille d'une image :

Le pixel : L'unité de mesure de l'écran. **Ex** : 500px

Le pourcentage : La taille de l'élément est définie en fonction de la taille écran de l'utilisateur. **Ex** : 50%

Les 2 propriétés CSS correspondantes s'intitulent « **width** » et « **height** ».

Ex :

```
<img style="width:50%;" />
<!--Ou-->
<img style="width:500px;" />
```

NB : Lorsque nous définissons une seule de ces caractéristiques (largeur ou hauteur), la définition manquante s'agrandira proportionnellement de façon à respecter le principe d'homothétie de l'image... (le ratio)

13. Centrer une image

Pour centrer une balise image en CSS, il convient d'associer 2 propriétés CSS : « display », qui permet de redéfinir le type d'élément en « block », et « margin » qui permet de centrer de la manière suivante :

Ex :

```

```

14. Les marges extérieures (margin)

Les marges permettent de définir un écart entre les blocks et d'autres éléments adjacents dans la mise en page.

Pour chaque block il est possible de définir indépendamment une marge haute, droite, basse ou gauche via les propriétés CSS « margin-top », « margin-right », « margin-bottom » et « margin-left ».

```
<div style="margin-top:5px;"> </div> <!-- Marge haute -->
<div style="margin-right:10px;"> </div> <!-- Marge droite -->
<div style="margin-bottom:15px;"> </div> <!-- Marge bas -->
<div style="margin-left:20px;"> </div> <!-- Marge gauche -->
```

Il est possible de coupler ces propriétés.

```
<div style="margin-top:5px; margin-bottom:15px;"> </div>
<!-- Marge haute et basse-->
```

Il existe une autre modalité d'écriture que l'on nomme « **concaténée** » où nous allons définir successivement la valeur des 4 marges que l'on va avoir, **en suivant le sens des aiguilles d'une montre** : Haut, droite, bas, gauche

```
<div style="margin: 5px 10px 15px 20px;"> </div>
```

Une autre syntaxe concaténée permet de définir les marges d'un block avec 2 valeurs, haut et bas, gauche et droite.

```
<div style="margin: 20px 5px;"> </div>
```

Dans cet exemple, la boîte a comme propriété :

Une marge haute et basse de 20px / une marge à droite et à gauche de 5px.

NB : Il est possible d'utiliser des marges négatives en introduisant la valeur de la marge par un signe moins, «-».

Ex : margin-top:-50px ;

15. Marges intérieures (padding)

Il s'agit de la marge présente entre la zone de contenu et la bordure du « block », on parle de marge intérieure. Elle est généralement utilisée pour décoller le texte du contour du block.

Cette propriété CSS s'intitule « padding », et obéit aux mêmes règles de compositions que « margin ».

```
<div style="padding-left:15px;">...</div><!--un coté-->  
<div style="padding:20px;">...</div><!--tous les cotés-->  
<div style="padding:20px 5px 15px 10px;">...</div>  
<!--tous les cotés séparément -->
```