

Sommaire

Introduction - Un tutoriel HTML.....	2
Leçon 1 : Commençons depuis le début.....	3
Leçon 2 : Qu'est-ce que HTML?.....	5
Leçon 3 : Les éléments et les balises	7
Leçon 4 : Créer son premier site Web.....	9
Leçon 5 : Ce que vous avez appris jusqu'ici.....	13
Leçon 6 : Quelques éléments de plus	15
Leçon 7 : Les attributs	18
Leçon 8 : Les liens	21
Leçon 9 : Les images	24
Leçon 10 : Les tables	28
Leçon 11 : Plus sur les tables.....	32
Leçon 12 : La présentation (CSS).....	34
Leçon 13 : La mise en place des pages	38
Leçon 14 : Les standards du Web et la validation.....	40
Leçon 15 : Les dernières astuces.....	43

Introduction - Un tutoriel HTML

Souvent les personnes pensent qu'il est très difficile de faire un site Web. Ce n'est pas vrai ! **Tout le monde peut apprendre à faire un site Web.** Si vous lisez la suite, vous en aurez fabriqué un en juste une heure.

D'autres croient, à tort également, qu'il faut des logiciels chers et compliqués pour fabriquer des sites Web. C'est vrai que beaucoup de programmes prétendent pouvoir créer un site à votre place. Certains y parviennent mieux que d'autres. Mais si vous voulez qu'il soit bien fait, vous devez le construire vous-même. Heureusement, **c'est simple et gratuit, et vous avez déjà tous les programmes nécessaires.**

Le but de ce tutoriel est de vous donner une introduction facile néanmoins complète et exacte à la fabrication de sites Web. **Le tutoriel commence depuis le début et ne nécessite absolument aucune connaissance préalable en programmation.**

Le tutoriel ne peut pas tout montrer. Et il faudra donc de l'engagement et une volonté d'expérimenter. Mais ne vous inquiétez pas, apprendre à fabriquer des sites Web est très amusant et est extrêmement gratifiant quand on le réussit.

Utilisez le tutoriel de la façon qu'il vous plaît. Toutefois nous suggérons de lire deux ou trois leçons par jour et de prendre son temps pour expérimenter les choses nouvelles apprises dans chaque leçon.

Bon, assez parlé. Commençons depuis le début

Leçon 1 : Commençons depuis le début

Dans cette première leçon, vous trouverez une présentation succincte des outils dont vous aurez besoin pour fabriquer un site Web.

De quoi a-t-on besoin ?

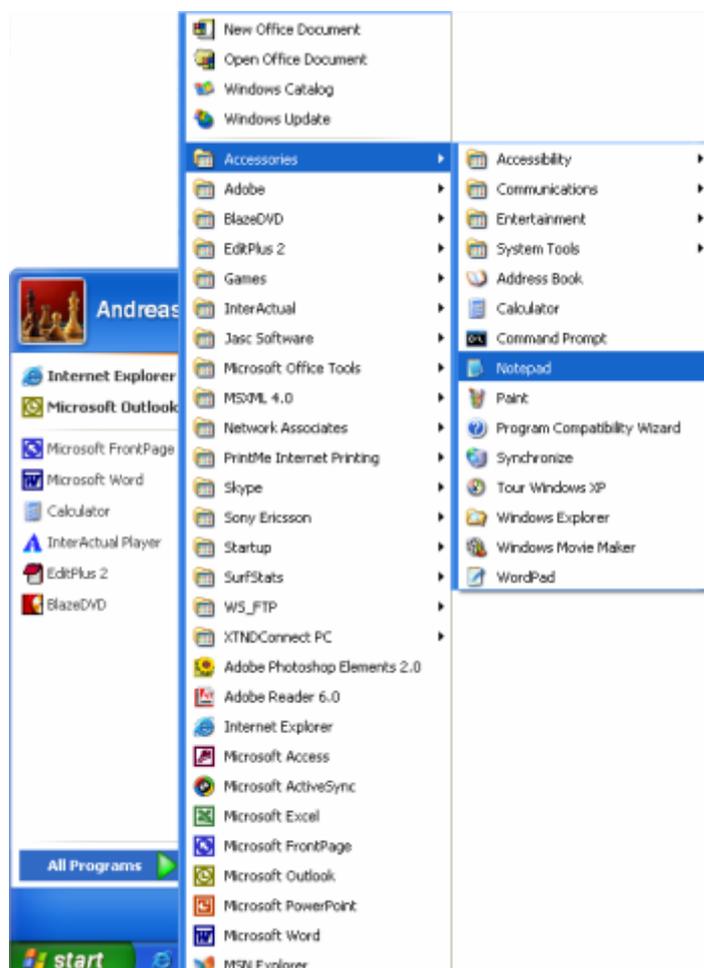
Vous avez sûrement déjà tout ce qui est nécessaire.

Vous avez un « navigateur ». Un navigateur est le programme qui vous permet de parcourir et d'ouvrir les sites Web. Vous êtes même en train de lire cette page dans votre navigateur.

Quel navigateur utiliser n'a aucune importance. Le plus courant est Internet Explorer de Microsoft. Mais il y en a d'autres comme Opera et Firefox de Mozilla, et on peut les utiliser tous.

Vous avez peut-être entendu parler, ou même pratiqué, des programmes tels que FrontPage de Microsoft, Dreamweaver de Macromedia ou encore Word de Microsoft, qui peuvent (ou prétendent) créer des sites Web pour vous. Oubliez ces programmes pour l'instant ! Ils ne seront d'aucune aide pour vous apprendre à coder votre site Web.

À la place, vous n'avez besoin que d'un simple éditeur de texte. Si vous êtes sur Windows, vous pouvez utiliser « Bloc-notes », trouvé habituellement dans le menu « démarrer », dans « Tous les programmes » à la rubrique « Accessoires » :



Si vous n'utilisez pas Windows, vous pouvez prendre un éditeur de texte simple similaire. Par exemple, Pico (Linux) ou TextEdit (Mac).

« Bloc-notes » est un programme d'édition de texte très élémentaire qui est excellent pour coder car il n'interfère pas avec ce que vous tapez. Il vous laisse le contrôle total. Le problème avec beaucoup de programmes prétendant créer des sites Web est qu'ils offrent un grand choix de fonctions standards. Par contre, tout doit rentrer dans ces fonctions standards. Ce type de programme ne peut donc pas créer un site Web exactement comme vous le souhaitez. Ou, plus gênant, ils changent le code que vous avez écrit. Avec « Bloc-notes » ou d'autres éditeurs de textes simples, vous êtes responsables de vos réussites ou de vos erreurs.

Un navigateur et « Bloc-notes » (ou un éditeur de texte simple similaire) sont tout ce dont vous avez besoin pour suivre ce tutoriel et fabriquer vos propres sites Web.

Est-ce que je dois être en ligne ?

Vous n'avez pas besoin d'être connecté à Internet, ni pour lire ce tutoriel ni pour la fabrication de vos sites Web.

Si vous ne voulez pas être connecté pendant la lecture de ce tutoriel, vous pouvez soit l'imprimer, soit simplement vous déconnecter pendant la lecture à l'écran. Vous pouvez élaborer le site Web sur le disque dur de votre ordinateur puis le placer sur Internet lorsqu'il est fini.

Quoi d'autre ?

Passez à la leçon suivante pour y lire ce qu'il en est de HTML avant d'entrer dans le vif du sujet dans la leçon 3.

Leçon 2 : Qu'est-ce que HTML?

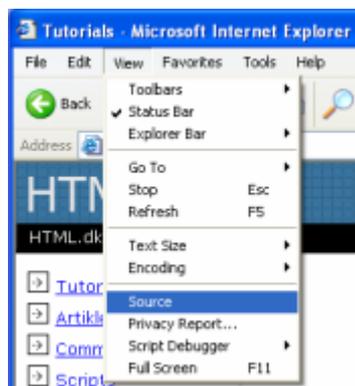
Cette leçon vous présente brièvement votre nouvel ami : HTML.

Qu'est-ce que HTML?

HTML est la « langue maternelle » de votre navigateur.

Pour faire court, HTML a été inventé en 1990 par un scientifique nommé Tim Berners-Lee. L'objectif était de faciliter l'accès par des scientifiques d'universités différentes aux documents de recherche de chacun. Le projet eut un succès inespéré, plus grand que tout ce que Tim Berners-Lee avait pu imaginer. En inventant HTML, il posa les fondations du Web tel que nous le connaissons aujourd'hui.

HTML est un langage qui permet de présenter des informations (par exemple, des recherches scientifiques) sur Internet. Ce que vous voyez quand vous regardez une page sur Internet est l'interprétation par votre navigateur du code HTML. Pour voir le code HTML d'une page sur Internet, cliquez simplement sur la rubrique « Vue » dans le menu supérieur de votre navigateur et sélectionnez « Source ».



Pour un œil non entraîné, le code HTML semble compliqué mais ce tutoriel vous aidera à en déterminer tout le sens.

À quoi puis-je utiliser HTML ?

Si vous voulez faire des sites Web, il n'y a pas d'autre voie que HTML. Même si vous vous servez d'un programme pour créer des sites Web, tel que Dreamweaver, une connaissance élémentaire de HTML peut vous rendre les choses plus faciles et beaucoup améliorer votre site Web. La bonne nouvelle c'est que HTML est facile à apprendre et à utiliser. D'ici deux leçons tout juste, vous aurez appris comment faire votre premier site Web.

HTML sert à faire des sites Web. C'est aussi simple que ça !

Bon, mais à quoi correspondent les lettres « H T M L » ?

HTML est l'abréviation de « HyperText Mark-up Language », et c'est tout ce qu'il faut savoir pour le moment. Toutefois, pour bien situer les choses, voici une explication plus précise.

- « **Hyper** » s'oppose à linéaire. Au bon vieux temps, quand les souris n'étaient encore que ce que les chats attrapaient, les programmes informatiques

s'exécutaient de façon linéaire : lorsque le programme avait terminé une action, il allait à la ligne suivante, puis encore à la suivante, et ainsi de suite. Mais HTML est différent : vous pouvez aller n'importe où et quand vous le voulez. Par exemple, il n'est pas nécessaire de visiter MSN.com avant de visiter HTML.net.

- « **Text** » s'explique tout seul.
- « **Mark-up** » (N.d.T. balisage) est ce que vous faites avec le texte. Vous marquez le texte de la même façon que vous le feriez dans un logiciel de traitement de texte avec des titres, des puces et des caractères gras, etc.
- « **Language** » (N.d.T. langage) est ce qu'est HTML. Il utilise plusieurs mots anglais.

Dans ce tutoriel, vous apprendrez le dénommé XHTML (Extensible HyperText Markup Language), qui est en bref une nouvelle façon mieux structurée d'écrire du HTML.

Maintenant que vous savez à quoi correspond HTML (et XHTML), allons-y pour ce qui nous concerne : fabriquer des sites Web.

Leçon 3 : Les éléments et les balises

Vous êtes maintenant prêt à apprendre l'essence de HTML : les éléments.

Les éléments donnent la structure d'un document HTML et indiquent comment vous voulez que le navigateur présente votre site Web. En général, les éléments se composent d'une balise ouvrante, d'un contenu et d'une balise fermante.

Les balises ?

Les balises sont les étiquettes que vous utilisez pour marquer le début et la fin d'un élément.

Toutes les balises ont le même format : elles commencent par un signe inférieur à « < » et finissent par un signe supérieur à « > ».

Généralement parlant, il y a deux types de balises : les balises ouvrantes (<html>) et les balises fermantes (</html>). La seule différence entre les deux est la barre oblique « / ». Vous étiquetez le contenu en le plaçant entre une balise ouvrante et une balise fermante.

HTML n'est qu'une question d'éléments. Apprendre HTML c'est connaître et utiliser des balises différentes.

Montrez-moi des exemples ?

D'accord, l'élément `b` sert pour marquer des lettres en caractères gras. Tout le texte entre la balise ouvrante `` et la balise fermante `` s'inscrit en caractères gras dans le navigateur. La lettre « `b` » est l'abréviation de « `bold` » (N.d.T. gras).

Exemple 1 :

```
<b>Ce texte doit être en caractères gras.</b>
```

Voici ce que ça donne dans le navigateur :

Ce texte doit être en caractères gras.

Les éléments `h1`, `h2`, `h3`, `h4`, `h5` et `h6` servent à marquer des titres, « `h` » tient pour « `heading` » (N.d.T. titre), où `h1` est le premier niveau et avec normalement le corps le plus grand, `h2` est le deuxième niveau et normalement un texte légèrement plus petit, et `h6` est le sixième et dernier niveau dans la hiérarchie des titres et normalement le plus petit texte.

Exemple 2 :

```
<h1>Voici un titre</h1>
<h2>Voici un sous-titre</h2>
```

Dans le navigateur, cela donnera :

Voici un titre

Voici un sous-titre

Donc, j'ai toujours besoin d'une balise ouvrante et d'une balise fermante ?

À ce qu'on dit, il y a une exception à chaque règle, et en HTML, quelques éléments font exception, avec une ouverture et une fermeture concentrée dans une seule balise. Ces éléments appelés éléments vides ne sont pas associés à un passage particulier dans le texte mais sont plutôt des étiquettes isolées, par exemple, un saut de ligne qui se présente comme ceci : `
`.

Les balises s'écrivent-elles en majuscules ou en minuscules ?

La plupart des navigateurs ne se soucient guère que vous tapiez vos balises en majuscules, en minuscules, ou dans un mélange des deux. `<HTML>`, `<html>` ou `<HtMl>` donneront normalement le même résultat. Quoi qu'il en soit, la méthode **correcte** est de taper les balises en minuscules. Donc, prenez l'**habitude d'écrire vos balises en minuscules**.

Où dois-je placer toutes ces balises ?

Vous tapez les balises dans un document HTML. Un site Web contient un ou plusieurs documents HTML. Lorsque vous surfez sur le Web, vous ouvrez simplement des documents HTML différents.

Si vous poursuivez jusqu'à la prochaine leçon, vous aurez fait votre premier site Web dans 10 minutes.

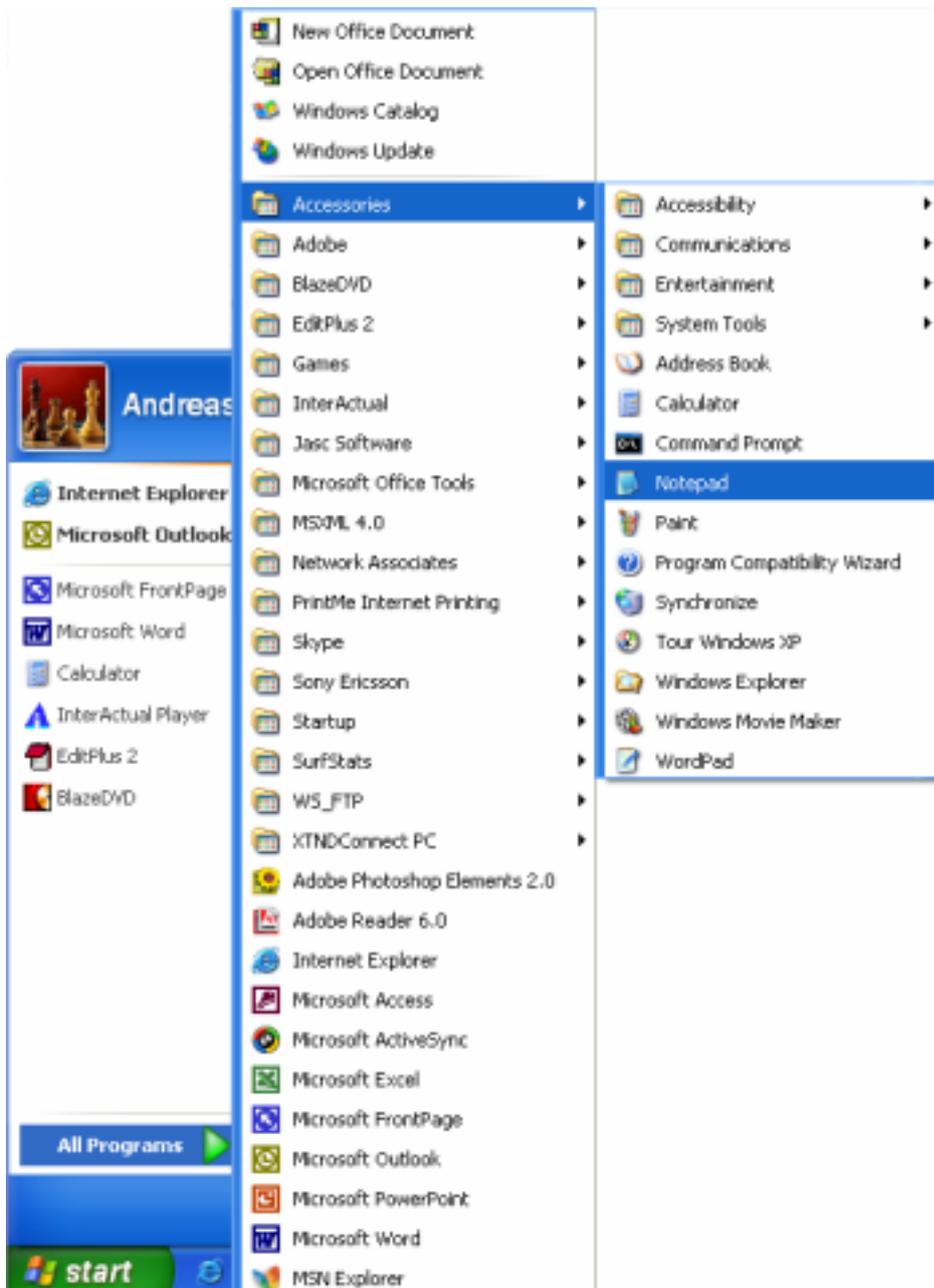
Leçon 4 : Créer son premier site Web

Avec ce que vous avez appris dans les leçons précédentes, vous êtes maintenant à quelques minutes de réaliser votre premier site Web.

Comment ?

Dans la leçon 1, nous avons vu ce qui était nécessaire pour faire un site Web : un navigateur et « Bloc-notes » (ou un éditeur de texte similaire). Puisque vous lisez ces lignes, votre navigateur est déjà ouvert. Vous devez seulement ouvrir une autre fenêtre de navigateur (ouvrez le navigateur encore une fois) afin que vous puissiez lire ce tutoriel et en même temps voir votre nouveau site Web.

Lancez également « Bloc-notes » (dans Accessoires sous Programmes dans le menu Start) :



Nous sommes maintenant prêts !

Que dois-je faire ?

Commençons par quelque chose de simple. Que dites-vous d'une page où s'inscrirait : « Hourra ! Mon premier site Web. » Lisez la suite et vous verrez combien c'est facile.

HTML est simple et logique. **Le navigateur lit le code HTML comme vous, de haut en bas et de gauche à droite.** Un document HTML commence donc par ce qui vient en premier dans la page et finit par ce qui vient en dernier.

La première chose à faire est de dire au navigateur que vous lui « parlerez » dans le langage HTML. On le fait avec la balise `<html>` (ce n'est pas surprenant ici). Donc, avant de faire quoi que ce soit d'autre, tapez « `<html>` » à la première ligne de votre document dans « Bloc-notes ».

Comme vous l'avez peut-être retenu des leçons précédentes, `<html>` est une balise ouvrante qui doit être close par une balise fermante à la fin de la saisie du code HTML. Pour vous assurer de ne pas oublier la balise fermante du code HTML, tapez donc maintenant la balise fermante `</html>` quelques lignes en-dessous, vous écrirez le reste du document entre `<html>` et `</html>`.

Le document a ensuite besoin d'une « en-tête » (N.d.T. head), laquelle fournit des informations à propos du document, et d'un « corps » (N.d.T. body), lequel constitue le contenu du document. Puisque HTML n'est rien sans logique, l'en-tête (`<head>` et `</head>`) se trouve avant le corps (`<body>` et `</body>`).

Votre document devrait ressembler à ceci :

```
<html>

  <head>
  </head>

  <body>
  </body>

</html>
```

Notez comment nous avons structuré les balises avec des nouvelles lignes (avec la touche Entrée) ainsi que des indentations (avec la touche Tabulation). En principe, la façon dont on structure le document HTML n'a aucune importance. Mais pour vous aider (et ceux qui liront votre code) à garder une vue d'ensemble, **il est fortement recommandé de structurer son code HTML proprement avec des sauts de ligne et des indentations**, comme dans l'exemple ci-dessus.

Si votre document est comme l'exemple précédent, vous avez fait votre premier site Web, un site particulièrement terne et probablement pas celui dont vous avez rêvé en entamant ce tutoriel mais un site Web quand même. Ce que vous avez fait constituera la charpente de tous vos documents HTML futurs.

C'est très bien, mais comment ajouter du contenu à mon site Web ?

Comme vu auparavant, votre document HTML se compose de deux parties : une en-tête et un corps. **Dans la section d'en-tête, vous inscrivez des informations concernant la page, tandis que le corps contient les informations qui constituent la page.**

Par exemple, si vous souhaitez donner un titre à la page, qui apparaîtra dans la barre supérieure du navigateur, il faudra le faire dans la section « head ». L'élément servant pour un titre est `title`. C'est-à-dire que vous devez écrire le titre de la page entre la balise ouvrante `<title>` et la balise fermante `</title>` :

```
<title>Mon premier site Web</title>
```

Remarquez que ce titre n'apparaîtra pas dans la page même. Tout ce que vous voulez faire apparaître dans la page est du contenu, et doit donc être ajouté entre les balises « body ».

Comme promis, nous voulons que la page affiche « Hourra ! Mon premier site Web ». C'est le texte que nous voulons communiquer et il appartient ainsi à la section « body ». Donc dans la section de corps, tapez ce qui suit :

```
<p>Hourra ! Mon premier site Web.</p>
```

Le « p » dans `<p>` est l'abréviation de « paragraphe », et c'est exactement ça, un paragraphe de texte.

Votre document HTML devrait ressembler à ça :

```
<html>

  <head>
    <title>Mon premier site Web </title>
  </head>

  <body>
    <p>Hourra ! Mon premier site Web.</p>
  </body>

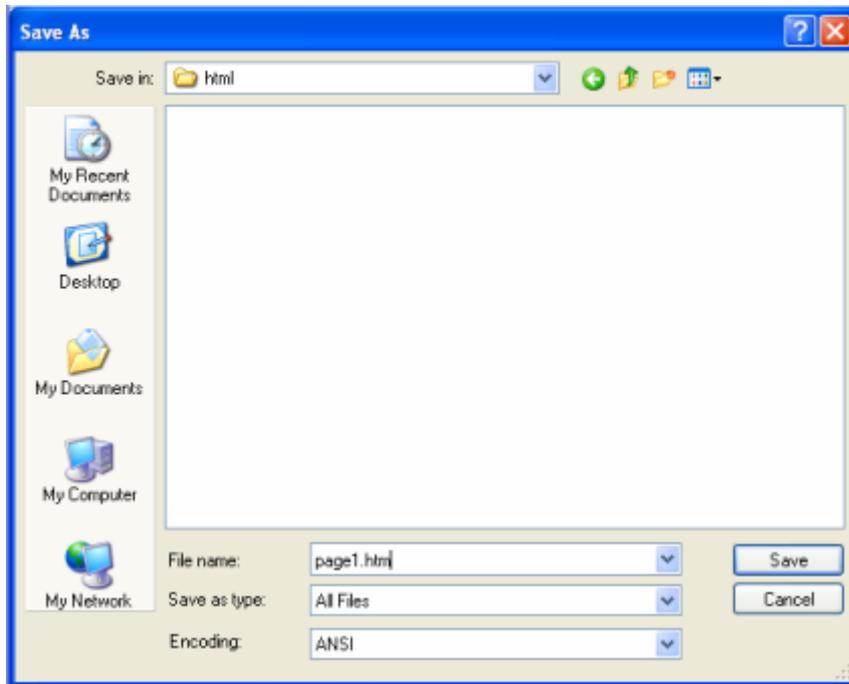
</html>
```

Ça y est ! Vous avez maintenant fabriqué votre premier véritable site Web !

Tout ce que vous avez à faire ensuite est de l'enregistrer sur votre disque dur et de l'ouvrir dans votre navigateur :

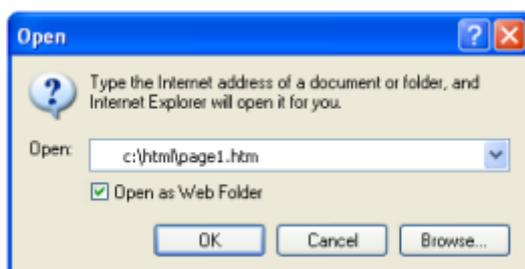
- Dans « Bloc-notes », sélectionnez « Enregistrer sous... » dans « Fichier », dans le menu supérieur.

- Sélectionnez « Tous fichiers » dans la boîte « Enregistrer avec le type ». C'est très important, car sinon vous l'enregistrerez comme un document de type texte et non comme document HTML.
- Maintenant sauvegardez votre document avec le nom « page1.htm » (le suffixe « .htm » indique qu'il s'agit d'un document HTML, « .html » donne le même résultat. J'utilise toujours « .htm », mais vous pouvez choisir l'extension que vous préférez). L'endroit où vous enregistrez le document sur le disque dur importe peu, tant que vous savez où le retrouver.



Maintenant allez au navigateur :

- Dans le menu supérieur, sélectionnez « Ouvrir » dans « Fichier ».
- Cliquez « Parcourir » dans le dialogue qui apparaît.
- Cherchez votre document HTML et cliquez « Ouvrir ».



Il devrait alors s'afficher « Hourra ! Mon premier site Web.» dans le navigateur. Félicitations !

Si vous tenez absolument à ce que le monde entier voit votre chef d'œuvre tout de suite, vous pouvez sauter à la leçon 14 et apprendre à placer la page sur Internet. Sinon soyez patient et continuez à lire. La fête a juste commencé.

Leçon 5 : Ce que vous avez appris jusqu'ici

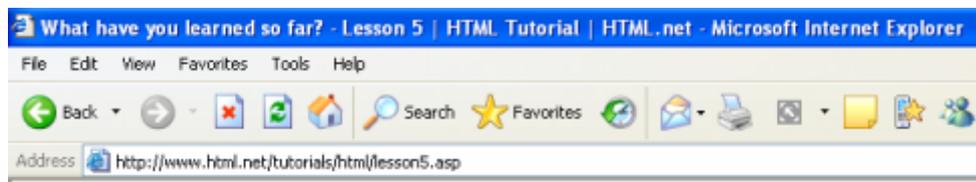
Toujours commencer depuis le gabarit minimal établi dans la leçon précédente :

```
<html>
  <head>
    <title></title>
  </head>

  <body>
  </body>

</html>
```

Dans la section « head », toujours écrire un titre : `<title>Le titre de la page</title>`. Notez que le titre apparaîtra dans le coin supérieur gauche de votre navigateur :



Le titre est particulièrement important car il sert aux moteurs de recherche (tel que Google) pour indexer votre site Web, et il apparaît dans les résultats de recherche.



Web

Tip: Try removing quotes from your search to get more results.

[What have you learned so far? - Lesson 5 | HTML Tutorial | HTML.net](#)

Lesson 5: What have you learned so far? Always start with the basic template we made in the previous lesson: `<html> <head> <title></title> </head> <body> ...`

www.html.net/tutorials/html/lesson5.asp - 5k - [Cached](#) - [Similar pages](#)

Dans la section « body », vous inscrivez le contenu réel de la page. Vous connaissez déjà quelques-uns des éléments les plus importants :

```
<p>Utilisé pour les paragraphes.</p>
<b>Affiche le texte en caractères gras.</b>
<h1>Titre</h1>
<h2>Sous-titre</h2>
<h3>Sous-sous-titre</h3>
```

Retenez que la seule façon d'apprendre HTML se fait par tentatives et erreurs. Mais ne vous inquiétez pas, vous ne détruisez pas votre ordinateur ni Internet. Continuez à expérimenter, c'est le meilleur moyen d'acquérir de l'expérience.

Qu'est-ce que c'est supposé vouloir dire ?

Personne ne peut devenir un bon créateur de site Web en apprenant les exemples de ce tutoriel. Ce que vous y trouverez est juste une compréhension élémentaire des pièces de construction, et pour devenir bon vous devez les utiliser dans des situations nouvelles et créatives.

Donc, lancez-vous à l'eau et tenez-vous sur vos deux pieds... bon, peut-être pas. Mais faites une tentative et expérimentez avec ce que vous avez appris.

Et ensuite ?

Essayez de créer quelques pages seul(e). Par exemple, montez une page avec un « title », un titre, du texte, un sous-titre et encore du texte. Vous pouvez très bien regarder dans le tutoriel pour créer vos premières pages. Mais plus tard, voyez si vous pouvez le faire seul(e) sans regarder

Leçon 6 : Quelques éléments de plus

Avez-vous pu créer quelques pages par vous-même ? Sinon voici un exemple :

```
<html>

  <head>
  <title>Mon site Web</title>
  </head>

  <body>
  <h1>Un titre</h1>
  <p>du texte, du texte du texte, du texte</p>
  <h2>Sous-titre</h2>
  <p>du texte, du texte du texte, du texte</p>
  </body>

</html>
```

Et maintenant quoi ?

Il est temps d'apprendre sept éléments nouveaux.

De la même façon qu'on met du texte en caractères gras en le plaçant entre une balise ouvrante `` et une balise fermante ``, on peut afficher le texte en italique avec la balise ouvrante `<i>` et la balise fermante `</i>`. Bien sûr, le « i » est l'abréviation de « italique ».

Exemple 1 :

```
<i>Voici un texte en italique.</i>
```

Ce qui donne dans le navigateur :

Voici un texte en italique.

De même, on peut diminuer le texte avec la balise `small` :

Exemple 2 :

```
<small>Voici un text diminué.</small>
```

Ce qui donne dans le navigateur :

Voici un texte diminué.

Puis-je utiliser plusieurs éléments en même temps ?

Vous pouvez aisément utiliser plusieurs éléments en même temps tant que vous évitez le **chevauchement des éléments**. Cet exemple le montre mieux :

Exemple 3 :

Si vous voulez écrire un texte en caractères gras et en italique, il faut le faire comme ceci :

```
<b><i>Un texte en caractères gras et en italique</i></b>
```

Et NON comme cela :

```
<b><i>Un texte en caractères gras et en italique</b></i>
```

La différence tient au fait que, dans le premier exemple, nous avons clos la balise ouverte la première en dernier. Nous évitons ainsi de nous tromper ainsi que le navigateur.

Plus d'éléments !

Comme indiqué dans la leçon 4, **il existe des éléments à la fois ouverts et fermés dans la même balise**. Ces éléments dits vides ne sont pas associés à un passage particulier dans le texte mais sont des étiquettes isolées. Un exemple d'un tel élément est celui de `
`, qui force un saut de ligne :

Exemple 4 :

```
Du texte<br /> et encore dans une nouvelle ligne
```

Dans le navigateur, cela donne :

Du texte

et encore dans une nouvelle ligne

Remarquez que la balise s'écrit comme la contraction d'une balise ouvrante et d'une balise fermante, avec un espace et une barre oblique à la fin : `
`. En principe, on pourrait aussi l'écrire `
</br>`, mais pourquoi rendre les choses plus compliquées que nécessaire ?

Un autre élément ouvert et fermé dans la même balise est `<hr />`, lequel sert à tracer une ligne horizontale ; « hr » tient pour « horizontal rule » (N.d.T. ligne horizontale) :

Exemple 5 :

```
<hr />
```

Dans le navigateur :

Comme exemples d'éléments nécessitant une balise ouvrante et une balise fermante, comme c'est le cas pour la plupart, les éléments `ul`, `ol` et `li`. Ces éléments servent à fabriquer des listes.

« ul » est l'abréviation de « unordered list » (N.d.T. liste non ordonnée), et insère des puces pour chaque élément de liste. « ol » est l'abréviation de « ordered list » (N.d.T. liste ordonnée), et numérote chaque élément de liste. Pour créer les éléments de la liste, on utilise la balise `li`, abréviation de « list item » (N.d.T. élément de liste). Confus ? Voyez les exemples :

Exemple 7 :

```
<ul>
  <li>Un élément de liste</li>
  <li>Un autre élément de liste</li>
</ul>
```

Ça donnera dans le navigateur :

- Un élément de liste
- Un autre élément de liste

Exemple 8 :

```
<ol>
  <li>Premier élément de liste</li>
  <li>Second élément de liste</li>
</ol>
```

Et dans le navigateur :

1. Premier élément de liste
2. Second élément de liste

Ouf ! C'est tout ?

C'est tout pour l'instant. Encore une fois, expérimentez et fabriquez vos propres pages avec les sept éléments appris dans cette leçon :

```
<i>Italique</i>
<small>Texte diminué</small>
<br /> Saut de ligne
<hr /> Ligne horizontale
<ul>Liste</ul>
<ol>Liste ordonnée</ol>
<li>Élément de liste</li>
```

Leçon 7 : Les attributs

Vous pouvez ajouter des attributs à beaucoup d'éléments.

Qu'est-ce qu'un attribut ?

Comme vous le savez, les éléments structurent le document HTML et indiquent au navigateur comment présenter le site Web (par exemple, `
` dit au navigateur de faire un saut de ligne). Certains éléments peuvent recevoir plus d'information. Cette information supplémentaire est appelée un attribut.

Exemple 1 :

```
<h2 style="background-color:#ff0000;">Mon affinité avec  
HTML</h2>
```

Les attributs s'inscrivent toujours au sein de la balise ouvrante, et ils sont suivis par un signe égal et la valeur de l'attribut entre des guillemets. Ici le point-virgule sert à séparer plusieurs instructions de style. Nous y reviendrons plus tard.

Qu'est-ce que c'est ?

Il existe beaucoup d'attributs différents. Le premier que vous allez voir est « style ». **Avec l'attribut `style`, vous pouvez ajouter une mise en page à votre site.** Par exemple, une couleur d'arrière-plan :

Exemple 2 :

```
<html>  
  
  <head>  
  </head>  
  
  <body style="background-color:#ff0000;">  
  </body>  
  
</html>
```

Ici la page sera complètement rouge dans le navigateur, faites-le et voyez vous-même. Nous expliquerons bientôt en détails comment le système de couleurs fonctionne.

Notez que quelques balises et attributs ont une orthographe américaine, par exemple, « color » et non « colour ». Il faut faire attention à utiliser la même orthographe que la nôtre dans les exemples de ce tutoriel, sinon les navigateurs ne pourront pas comprendre votre code. Également, n'oubliez jamais le guillemet final de l'attribut.

Comment la page est-elle devenue rouge ?

Dans l'exemple précédent, nous avons demandé la couleur d'arrière-plan "#ff0000". C'est un code de couleur pour rouge en nombre hexadécimal (HEX). **Chaque couleur est associée à un nombre hexadécimal.** Voici quelques exemples :

Blanc : #ffffff
Noir : #000000 (des zéros)
Rouge : #ff0000
Bleu : #0000ff
Vert : #00ff00
Jaune : #ffff00

Un code de couleur hexadécimal se compose d'un caractère « # » et de six chiffres ou lettres. Il existe plus de 1000 codes HEX, et il n'est pas facile de déterminer quel code HEX est associé à telle couleur particulière. Pour faciliter les choses, nous avons réalisé un tableau des 216 couleurs les plus courantes : [Le tableau des 216 couleurs sûres pour le Web.](#)

Vous pouvez également utiliser les noms anglais des couleurs les plus courantes, à savoir "white", "black", "red", "blue", "green" et "yellow" (N.d.T. blanc, noir, rouge, bleu, vert et jaune).

Exemple 3 :

```
<body style="background-color: red;">
```

Assez pour les couleurs. Revenons aux attributs.

Quels éléments ont des attributs ?

On peut appliquer plusieurs attributs à la plupart des éléments.

Vous utiliserez souvent des attributs dans des balises comme « body » et rarement, par exemple, dans une balise « br » puisqu'un saut de ligne ne s'ajuste normalement pas.

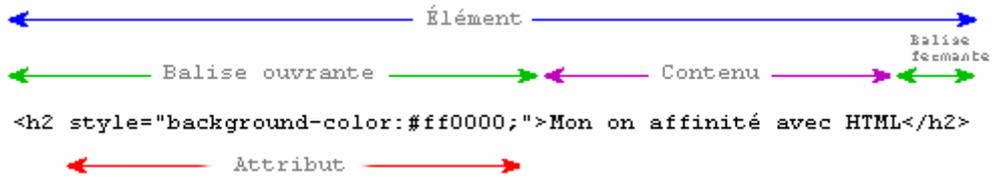
Tout comme pour les éléments, il y a beaucoup d'attributs différents. Certains attributs sont taillés sur mesure pour un seul élément particulier tandis que d'autres s'utilisent avec plusieurs éléments différents. Et vice versa : certains éléments peuvent uniquement contenir un seul type d'attribut tandis que d'autres peuvent en contenir plusieurs.

Cela peut sembler embrouillé mais une fois familiarisé avec les différents attributs, c'est en réalité très logique, et vous verrez rapidement qu'ils sont très faciles à utiliser et qu'ils offrent beaucoup de possibilités.

Ce tutoriel introduira les attributs les plus importants.

De quelles parties un élément est-t-il constitué exactement ?

Généralement, un élément se compose d'une balise ouvrante avec zéro à plusieurs attributs, d'un contenu et d'une balise fermante. Aussi simple que ça. Voir l'illustration ci-dessous.



Leçon 8 : Les liens

Dans cette leçon, vous apprendrez à faire des liens entre les pages.

De quoi ai-je besoin pour faire un lien ?

Pour faire un lien, on se sert de ce qu'on utilise toujours pour coder du HTML : un élément. **Un simple élément avec un seul attribut et vous pourrez relier tout et n'importe quoi.** Voici un exemple montrant à quoi un lien vers HTML.net aurait l'air :

Exemple 1 :

```
<a href="http://www.html.net/">Voici un lien vers HTML.net</a>
```

Ce qui donnerait dans le navigateur :

[Voici un lien vers HTML.net](http://www.html.net)

L'élément « a » tient pour « anchor » (N.d.T. ancre). Et l'attribut `href` est l'abréviation de « hypertext reference » (N.d.T. appel hypertexte), qui indique où conduit le lien, habituellement à une adresse Internet ou à un nom de fichier.

Dans l'exemple précédent, l'attribut `href` vaut "http://www.html.net", ce qui correspond à l'adresse complète de HTML.net et qu'on appelle une adresse URL (Uniform Resource Locator). Notez qu'il faut toujours inclure « http:// » dans les adresses URL. L'expression « Voici un lien vers HTML.net » est le texte affiché dans le navigateur pour le lien. Rappelez-vous de clore l'élément par ``.

Et pour les liens entre mes propres pages ?

Si vous voulez faire un lien entre des pages sur le même site, il n'est pas nécessaire d'écrire l'adresse (URL) entière du document. Par exemple, si vous avez deux pages (appelons-les `page1.htm` et `page2.htm`) enregistrées dans le même dossier, vous pouvez faire un lien d'une page à l'autre simplement en inscrivant le nom du fichier dans le lien. Dans ces conditions, un lien de `page1.htm` vers `page2.htm` apparaîtrait comme ceci :

Exemple 2 :

```
<a href="page2.htm">Cliquez ici pour aller à la page 2</a>
```

Si la page 2 se trouvait dans un sous-dossier (appelé "sous-dossier"), le lien serait alors :

Exemple 3 :

```
<a href="sous-dossier/page2.htm">Cliquez ici pour aller à la page 2</a>
```

En sens inverse, un lien de la page 2 (dans le sous-dossier) à la page 1 serait le suivant :

Exemple 4 :

```
<a href="../../../page1.htm">Un lien vers la page 1</a>
```

Le chemin « ../ » pointe sur le dossier à un niveau supérieur par rapport à la position du fichier où le lien se trouve. En suivant le même raisonnement, vous pouvez aussi pointer sur une position deux dossiers (ou plus) en amont avec « ../../ ».

Vous avez compris le système ? Autrement, vous pouvez toujours écrire l'adresse (URL) complète du fichier.

Et pour les liens dans la même page ?

On peut également créer des liens internes à une page, par exemple, une table des matières en haut de la page avec des liens vers chaque chapitre en-dessous. Vous avez seulement besoin d'un attribut très utile, appelé `id` (pour identification), et du symbole « # ».

Utilisez l'attribut `id` pour marquer l'élément que vous voulez relier. Par exemple :

```
<h1 id="heading1">titre 1</h1>
```

On crée ensuite un lien vers cet élément en utilisant le symbole « # » dans l'attribut de liaison. Le « # » indique au navigateur de rester sur la même page. Ce caractère doit être suivi de l'`id` de la balise à relier. Par exemple :

```
<a href="#heading1">Lien vers le titre 1</a>
```

Tout sera plus clair avec un exemple :

Exemple 5 :

```
<html>

  <head>
  </head>

  <body>

    <p><a href="#heading1">Lien vers le titre 1</a></p>
    <p><a href="#heading2">Lien vers le titre 2</a></p>

    <h1 id="heading1">Titre 1</h1>
    <p>Texte texte texte texte</p>

    <h1 id="heading2">Titre 2</h1>
    <p>Texte texte texte texte</p>

  </body>

</html>
```

Dans le navigateur, cela apparaîtra comme ceci (cliquez sur les deux liens) :

[Lien vers le titre 1](#)

[Lien vers le titre 2](#)

Titre 1

Texte texte texte texte

Titre 2

Texte texte texte texte

(Remarque : L'attribut `id` doit commencer par une lettre).

Puis-je relier autre chose ?

On peut également faire un lien vers une adresse e-mail. Cela se fait presque de la même façon que pour relier un document :

Exemple 6 :

```
<a href="mailto:nobody@html.net">Envoyer un e-mail à  
« nobody » chez HTML.net</a>
```

Dans le navigateur, cela donne :

[Envoyer un e-mail à « nobody » chez HTML.net](mailto:nobody@html.net)

La seule différence entre un lien d'e-mail et un lien de fichier, c'est que l'on tape « `mailto:` » suivi d'une adresse e-mail au lieu de taper l'adresse d'un document. Au clic du lien, le programme de messagerie par défaut se lance avec un message vide pour l'adresse e-mail indiquée. Veuillez noter que cela fonctionnera seulement si un programme de messagerie est installé sur votre ordinateur. Essayez donc !

Y a-t-il d'autres attributs à connaître ?

Pour créer un lien, vous devez toujours utiliser l'attribut `href`. Vous pouvez en outre placer un attribut `title` sur le lien :

Exemple 7 :

```
<a href="http://www.html.net/" title="Visitez HTML.net pour  
apprendre HTML">HTML.net</a>
```

Cela apparaîtra ainsi dans le navigateur :

[HTML.net](http://www.html.net/)

L'attribut `title` sert à fournir une brève description du lien. Si vous positionnez le curseur sur le lien sans le cliquer, vous verrez apparaître le message « Visitez HTML.net pour apprendre HTML ».

Leçon 9 : Les images

Est-ce qu'il ne serait pas formidable d'avoir une photo de l'acteur et légende musicale David Hasselhoff en plein milieu de la page ?

Ça sonne comme un défi ...

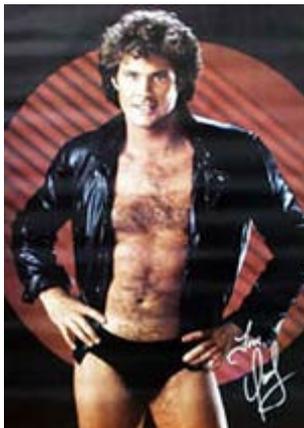
Peut-être, mais en réalité c'est très facile à faire. Il vous faut seulement un élément :

Exemple 1 :

```

```

Ce qui donne dans le navigateur :



Tout ce dont vous avez besoin est d'abord de dire au navigateur que vous voulez insérer une image (`img`) puis de lui dire où elle se trouve (`src`, abréviation de « source »). Vous voyez ?

Remarquez que l'élément `img` est ouvert et fermé avec la même balise. Comme la balise `
`, il n'est pas associé à un morceau de texte.

Le nom « david.jpg » est celui du fichier d'image à insérer dans la page, et « .jpg » est le type de fichier de l'image. Tout comme l'extension « .htm » indique qu'il s'agit d'un document HTML, « .jpg » indique au navigateur que ce fichier est une image. Il existe trois sortes de types de fichier d'image que vous pouvez insérer dans vos pages :

- GIF (Graphics Interchange Format)
- JPG / JPEG (Joint Photographic Experts Group)
- PNG (Portable Network Graphics)

Les images GIF conviennent habituellement mieux pour les graphiques et les dessins, et les images JPEG pour les photographies. Deux raisons à cela : premièrement les images GIF se composent seulement de 256 couleurs, tandis que les images JPEG en contiennent des millions, et deuxièmement, le format GIF est mieux adapté pour compresser des images simples que le format JPEG optimisé pour des images plus complexes. Meilleure la compression, plus petit le poids du fichier d'image, et plus rapide le chargement de la page. Comme l'expérience vous l'a sans doute montré, les pages inutilement « lourdes » peuvent être très gênantes pour l'utilisateur.

Traditionnellement, les formats GIF et JPEG ont été les deux types d'images dominants, mais récemment, le format PNG est devenu de plus en plus populaire (principalement aux dépens du format GIF). **Le format PNG représente de beaucoup de points de vue le meilleur des deux formats JPEG et GIF : des millions de couleurs et une compression efficace.**

Où trouver des images ?

Pour fabriquer vos propres images, il vous faut un programme d'édition d'images. **Un programme de création d'image est l'un des outils essentiels pour créer de jolis sites Web.**

Malheureusement, aucun bon programme d'édition d'images n'est fourni avec Windows ou d'autres systèmes d'exploitation. Il faudrait donc envisager d'investir dans l'un des trois meilleurs logiciels d'édition d'images actuellement sur le marché, à savoir « Paint Shop Pro », « PhotoShop » ou « Macromedia Fireworks ».

Quoi qu'il en soit, comme annoncé précédemment, il ne sera pas nécessaire d'acheter des logiciels coûteux pour finir ce tutoriel. Pour l'instant, vous pouvez télécharger l'excellent logiciel d'édition d'images « [Irfan View](#) », un gratuitiel (N.d.T. freeware) qui ne coûte donc rien.

Ou vous pouvez juste emprunter des images à d'autres sites en les téléchargeant. Mais faites attention à ce que le téléchargement des images n'enfreigne pas de droits d'auteur. Par contre, il est toujours utile de savoir télécharger des images, et voici comment :

1. Cliquez sur le bouton de droite après avoir placé le pointeur sur une image, quelle que soit l'image sur Internet ;
2. Sélectionnez « Sauvegarder l'image sous ... » dans le menu qui apparaît ;
3. Sélectionnez un emplacement sur votre ordinateur pour l'image et appuyez « Sauvegarder ».

Faites-le avec l'image suivante et sauvegardez-la sur votre ordinateur au même emplacement que vos documents HTML. (Remarquez que le logo est sauvegardé comme fichier PNG : logo.png) :



Vous pouvez maintenant insérer l'image dans l'une de vos pages. Essayez vous-même.

C'est tout ce que je dois savoir pour les images ?

Il y a d'autres choses à connaître concernant les images.

D'abord vous pouvez aisément insérer des images situées dans d'autres dossiers, ou même situées sur d'autres sites Web :

Exemple 2 :

```

```

Exemple 3 :

```

```

Deuxièmement, les images peuvent être des liens :

Exemple 4 :

```
<a href="http://www.html.net">  
</a>
```

Ce qui donnera dans le navigateur (cliquez sur l'image) :



Y a t-il d'autres attributs à connaître ?

Vous aurez toujours besoin d'utiliser l'attribut `src`, qui indique au navigateur où l'image se trouve. Hormis cela, beaucoup d'autres attributs peuvent être utiles pour insérer des images.

L'attribut `alt` sert à fournir une description de remplacement de l'image si, pour une raison ou pour une autre, l'image n'était pas affichée à l'utilisateur. C'est particulièrement important pour les utilisateurs aveugles, ou si la page se charge très lentement. Donc, toujours utiliser l'attribut `alt` :

Exemple 5 :

```

```

Certains navigateurs affichent le texte de l'attribut `alt` sous forme d'un petit dialogue contextuel lorsque l'utilisateur place le curseur sur l'image. Veuillez noter que l'utilisation de l'attribut `alt` vise à offrir une description de remplacement pour l'image. L'attribut `alt` ne devrait pas servir à créer des messages contextuels spéciaux pour les utilisateurs car les utilisateurs aveugles entendront le message sans savoir ce que l'image représente.

L'attribut `title` peut servir pour décrire brièvement l'image :

Exemple 6 :

```

```

Ce qui donne dans le navigateur :



Si vous placez le curseur sur l'image sans cliquer, vous verrez apparaître le texte « Apprenez HTML avec HTML.net » comme message contextuel.

Deux autres attributs importants sont `width` et `height` :

Exemple 7 :

```

```

Ce qui donne dans le navigateur :



On peut utiliser les attributs `width` et `height` pour fixer la largeur et la hauteur d'une image. Les valeurs utilisées pour fixer ces dimensions sont en pixels. Le pixel est l'unité de mesure de la résolution des écrans. (Les résolutions d'écrans les plus courantes sont de 800 x 600 et 1024 x 768 pixels). Au contraire des centimètres, les pixels sont des unités de mesure relatives qui dépendent de la résolution de l'écran. Pour une résolution d'écran élevée, 25 pixels peuvent correspondre à l'écran à 1 centimètre, tandis que sur un écran de basse résolution à 1,5 cm.

Si on ne fixe pas sa largeur et sa hauteur, l'image sera insérée dans sa dimension réelle. Mais on peut manipuler ses dimensions avec `width` et `height` :

Exemple 8 :

```

```

Ce qui donne dans le navigateur :



Par contre, il vaut mieux garder à l'esprit que le poids réel en kilo-octets du fichier d'image reste le même, et qu'il faudra autant de temps pour charger l'image, quoiqu'elle apparaisse plus petite à l'écran. Ainsi **vous ne devriez jamais réduire la dimension de l'image avec les attributs `width` et `height`**. Il faudrait plutôt toujours redimensionner les images avec un logiciel d'édition d'images afin de rendre ses pages plus légères et plus rapides.

Ceci dit, c'est toujours une bonne idée d'utiliser les attributs `width` et `height` car le navigateur pourra déterminer l'occupation de l'image dans la page avant son chargement complet. Cela permet au navigateur une mise en page harmonieuse et plus rapide.

Il suffit ici avec David Hasselhoff et les images.

Leçon 10 : Les tables

Les tables servent pour l'affichage de « données tabulaires », c'est-à-dire des informations présentées logiquement en rangées et en colonnes.

Est-ce difficile ?

Dresser des tables en HTML peut sembler compliqué au premier abord mais en restant calme et en regardant où on met les pieds, c'est en réalité strictement une affaire de logique, comme toute chose en HTML.

Exemple 1 :

```
<table>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

Ce qui donne dans le navigateur :

Cellule 1 Cellule 2

Cellule 3 Cellule 4

Quelle est la différence entre <tr> et <td> ?

Comme vous pouvez le voir dans l'exemple précédent, il s'agit probablement de l'exemple HTML le plus compliqué fourni jusqu'ici. Décomposons-le et expliquons les différentes balises :

On utilise 3 éléments différents pour insérer des tables :

- La balise ouvrante `<table>` et la balise fermante `</table>` marquent le début et la fin de la table. Logique.
- La balise `<tr>` tient pour « **t**able **r**ow » (N.d.T. rangée de table), et commence et termine les rangées dans le sens horizontal. Encore logique.
- La balise `<td>` est l'abréviation de « **t**able **d**ata » (N.d.T. données de table). Elle marque le début et la fin de chaque cellule dans les rangées de la table. Tout simple et logique.

Voici l'explication de l'exemple 1 : la table commence avec la balise `<table>`, suivie d'une balise `<tr>` indiquant le début d'une nouvelle rangée. Dans cette rangée, deux cellules sont insérées : `<td>Cellule 1</td>` et `<td>Cellule 2</td>`. La rangée se termine par une balise `</tr>`, et une nouvelle rangée `<tr>` commence juste après. Cette nouvelle rangée contient également deux cellules. La table se termine par une balise `</table>`.

Pour que les choses soient claires, **les rangées sont des lignes horizontales de cellules et les colonnes des lignes verticales de cellules** :

Cellule 1	Cellule 2
Cellule 3	Cellule 4

Cellule 1 et Cellule 2 forment une rangée, Cellule 1 et Cellule 3 une colonne.

Dans l'exemple précédent, la table a deux rangées et deux colonnes. Toutefois, une table peut avoir un nombre illimité de rangées et colonnes.

Exemple 2 :

```
<table>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
  <tr>
    <td>Cellule 5</td>
    <td>Cellule 6</td>
    <td>Cellule 7</td>
    <td>Cellule 8</td>
  </tr>
  <tr>
    <td>Cellule 9</td>
    <td>Cellule 10</td>
    <td>Cellule 11</td>
    <td>Cellule 12</td>
  </tr>
</table>
```

Dans le navigateur, cela donne :

Cellule 1 Cellule 2 Cellule 3 Cellule 4

Cellule 5 Cellule 6 Cellule 7 Cellule 8

Cellule 9 Cellule 10 Cellule 11 Cellule 12

Y a-t-il des attributs ?

Bien sûr qu'il y en a. Par exemple, l'attribut `border` sert à indiquer l'épaisseur de la bordure autour de la table :

Exemple 3 :

```
<table border="1">
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

Ce qui donne dans le navigateur :

Cellule 1	Cellule 2
Cellule 3	Cellule 4

L'épaisseur de la bordure s'exprime en pixels ([cf. leçon 9](#)).

Comme pour les images, on peut également fixer la largeur d'une table en pixels, ou autrement en pourcentage de l'écran :

Exemple 4 :

```
<table border="1" width="30%">
```

Cet exemple affichera une table dont la largeur est 30 % celle de l'écran dans le navigateur. Essayez par vous-même.

Plus d'attributs ?

Il y a des tas d'attributs pour les tables. En voici deux :

- align : indique l'alignement horizontal du contenu dans toute la table, dans une rangée ou dans une seule cellule. Par exemple, les valeurs "left" (à gauche), "center" (au centre) ou "right" (à droite).
- valign : indique l'alignement vertical du contenu dans une cellule. Par exemple, les valeurs "top" (en haut), "middle" (au milieu) ou "bottom" (en bas).

Exemple 5 :

```
<td align="right" valign="top">Cellule 1</td>
```

Que puis-je insérer dans mes tables ?

En théorie, on peut insérer n'importe quoi dans les tables : du texte, des liens et des images ... **TOUJOURS EST-IL que les tables sont faites pour présenter des données tabulaires** (c'est-à-dire, des données dont la présentation en rangées et

en colonnes revêt un sens), donc abstenez-vous d'y placer des choses simplement au motif que vous voulez les mettre l'une à côté de l'autre.

Aux premiers temps d'Internet, c'est-à-dire quelques années en arrière, les tables servaient souvent pour la mise en page. Mais si vous voulez contrôler la présentation du texte et des images, il existe une solution beaucoup plus fûtée pour y parvenir (indice : CSS). Plus à ce sujet tout à l'heure.

Maintenant, mettez en pratique ce que vous venez d'apprendre et créez plusieurs tables de dimensions différentes, avec des attributs et du contenu différents. Ça devrait vous occuper pour quelques heures.

Leçon 11 : Plus sur les tables

Ce titre « Plus sur les tables » sonnera peut-être ennuyeux. Mais voyez le bon côté, si vous maîtrisez les tables, plus rien concernant HTML ne sera inaccessible.

Que reste-t-il alors ?

Les deux attributs `colspan` et `rowspan` sont mis à profit pour créer des tables plaisantes.

Le terme « colspan » est l'abréviation de « column span » (N.d.T. recouvrement de colonne). **L'attribut `colspan` est utilisé dans la balise `<td>` pour indiquer combien de colonnes la cellule recouvre :**

Exemple 1 :

```
<table border="1">
  <tr>
    <td colspan="3">Cellule 1</td>
  </tr>
  <tr>
    <td>Cellule 2</td>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

Dans le navigateur, cela donne :

Cellule 1		
Cellule 2	Cellule 3	Cellule 4

En fixant la valeur de l'attribut `colspan` à "3", la cellule dans la première rangée recouvre trois colonnes. Si nous avions fixée sa valeur à "2", la cellule ne recouvrirait que deux colonnes, et il aurait fallu insérer une cellule supplémentaire dans la première rangée afin que le nombre de colonnes corresponde dans les deux rangées.

Exemple 2 :

```
<table border="1">
  <tr>
    <td colspan="2">Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
    <td>Cellule 5</td>
  </tr>
```

```
</table>
```

Dans le navigateur, cela donne :

Cellule 1	Cellule 2	
Cellule 3	Cellule 4	Cellule 5

Et pour rowspan ?

Comme vous l'avez peut-être deviné, **l'attribut rowspan indique combien de rangées la cellule doit recouvrir :**

Exemple 3 :

```
<table border="1">
  <tr>
    <td rowspan="3">Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
  </tr>
  <tr>
    <td>Cellule 4</td>
  </tr>
</table>
```

Dans le navigateur, cela donne :

Cellule 1	Cellule 2
	Cellule 3
	Cellule 4

Dans l'exemple précédent, la valeur de `rowspan` est fixée à "3" dans la Cellule 1. Elle indique que la cellule doit recouvrir trois rangées (la sienne propre plus deux autres). Cellule 1 et Cellule 2 sont donc dans la même rangée, tandis que Cellule 3 et Cellule 4 forment deux rangées indépendantes.

Embrouillé ? Bon, ce n'est pas si simple et on perd facilement le fil. De fait, on serait bien inspiré de dessiner la table sur un bout de papier avant de commencer le HTML.

C'est clair ? Alors allez-y et créez de votre chef quelques tables comportant à la fois des attributs `colspan` et `rowspan`.

Leçon 12 : La présentation (CSS)

Ne serait-ce pas super si vous pouviez donner à vos pages la présentation qu'elles méritent ?

C'est sûr, mais comment ?

Pour la présentation de votre site Web, utilisez les feuilles de style en cascade (CSS). Dans cette leçon, vous trouverez une brève introduction à CSS. Par la suite, vous pouvez apprendre CSS en entier depuis le début dans [notre tutoriel CSS](#). Donc considérez cette leçon seulement comme une mise en appétit.

CSS est la meilleure moitié de HTML. Et pour le codage, leur statut n'est pas égal : **HTML s'occupe du gros-œuvre (la structure), tandis que CSS fait les finitions (la présentation).**

Comme indiqué dans la [leçon 7](#), **on peut ajouter du code CSS avec l'attribut `style`**. Par exemple, on peut fixer le type et le corps de la police d'un paragraphe :

Exemple 1 :

```
<p style="font-size:20px;">Ce texte a un corps de 20</p>
<p style="font-family:courier;">Ce texte est dans la police
Courier</p>
<p style="font-size:20px; font-family:courier;">Ce texte est
dans la police Courier avec un corps de 20</p>
```

Dans le navigateur, cela donne :

Ce texte a un corps de 20

Ce texte est dans la police Courier

Ce texte est dans la police Courier avec un corps de 20

Dans cet exemple, nous nous servons de l'attribut `style` pour définir le type de police à utiliser (avec la commande `font-family`) et le corps de la police (avec la commande `font-size`). Remarquez dans le dernier paragraphe comment nous avons fixé à la fois le type et le corps de la police avec un point-virgule de séparation.

C'est beaucoup de travail ?

Une caractéristique astucieuse de CSS est la possibilité de gérer la présentation de manière centralisée. Au lieu d'utiliser l'attribut `style` dans chaque balise, on peut indiquer au navigateur en une seule fois comment disposer le texte sur la page :

Exemple 2 :

```
<html>
```

```
<head>
<title>Ma première page avec CSS</title>

<style type="text/css">
  h1 {font-size: 30px; font-family: arial;}
  h2 {font-size: 15px; font-family: courier;}
  p {font-size: 8px; font-family: "times new roman";}
</style>

</head>

<body>
<h1>Ma première page avec CSS</h1>
<h2>Bienvenue sur ma première page avec CSS</h2>
<p>Ici vous pouvez voir CSS à l'œuvre </p>
</body>

</html>
```

- [Afficher l'exemple](#)

Dans l'exemple précédent, **le code CSS a été placé dans la section « head » et il s'applique donc à la totalité de la page.** Pour ce faire, utilisez juste la balise `<style type="text/css">` indiquant au navigateur qu'il s'agit de CSS.

Dans l'exemple, tous les titres de la page seront dans la police Arial avec un corps de 30px, tous les sous-titres seront dans la police Courier avec un corps de 15px, et tout le texte des paragraphes ordinaires sera dans la police "times new roman" avec un corps de 8px.

Une autre option consiste à écrire le code CSS dans un document séparé. **Avec un document CSS séparé, vous pouvez gérer la présentation de plusieurs pages en même temps.** Très pratique si on veut changer le type ou le corps de la police d'un grand site Web contenant des centaines ou des milliers de pages. Nous n'aborderons pas ce sujet ici mais vous pouvez le voir plus tard dans notre tutoriel CSS.

Quoi faire d'autre avec CSS ?

On peut se servir de CSS pour bien plus que définir les types et les corps des polices. Par exemple, on peut ajouter des couleurs et des arrière-plans. Voici quelques exemples à expérimenter :

```
<p style="color:green;">Un texte en vert</p>

<h1 style="background-color: blue;">Un titre sur fond
bleu</h1>

<body style="background-image:
url('http://www.html.net/logo.png') ;">
```

Essayez de placer ces exemples dans quelques pages, à écrire comme ci-dessus et à insérer dans la section « head ».

Est-ce que CSS ne concerne que les couleurs et les polices ?

Hormis ajouter une présentation tels que des couleurs, des types de polices, etc., **CSS peut aussi servir à contrôler la mise en page et la présentation** (les marges, le flottement, l'alignement, la largeur, la hauteur, etc.). La régulation des différents éléments par CSS permet de présenter ses pages avec élégance et précision.

Exemple 3 :

```
<p style="padding:25px;border:1px solid red;">J'aime CSS</p>
```

Dans le navigateur, cela donne :

J'aime CSS

La propriété `float` permet de faire flotter un élément à droite ou bien à gauche. L'exemple suivant en illustre le principe :

Exemple 4 :

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit,
    sed diam nonummy nibh euismod tincidunt ut laoreet
dolore
    magna aliquam erat volutpat. Ut wisi enim ad minim
veniam,
    quis nostrud exerci tation ullamcorper suscipit
lobortis nisl ut aliquip ex ea commodo consequat...</p>
```

Dans le navigateur, cela donne :



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat...

Dans cet exemple, un élément (l'image) flotte à gauche, et l'autre (le texte) remplit l'espace ainsi libéré.

La propriété `position` permet de placer un élément exactement là où on veut qu'il soit dans la page :

Exemple 5 :

```

```

- [Afficher l'exemple](#)

Dans l'exemple, l'image se place à 50 pixels du bas et à 10 pixels de la droite dans le navigateur. Mais vous pouvez la placer exactement où vous voulez. Très facile et très cool, n'est-ce pas ?

Super oui ! Mais facile ?

On n'apprend pas CSS en 10 minutes. Et comme annoncé auparavant, cette leçon ne constitue qu'un bref aperçu. Vous apprendrez beaucoup plus après dans notre [tutoriel CSS](#).

Pour l'instant, concentrons-nous sur HTML et passons à la prochaine leçon où l'on apprendra à mettre en place le site Web sur Internet sous le regard du monde entier !

Leçon 13 : La mise en place des pages

Jusqu'à présent, vous n'avez eu la satisfaction que de visionner les pages. Il est temps maintenant que le reste du monde puisse admirer votre chef d'œuvre..

Le monde est-il prêt pour ça ?

Le monde est prêt, et bientôt vous le serez aussi. **Pour installer votre site Web sur Internet, vous avez juste besoin d'espace sur un serveur et d'un programme FTP libre.**

Si vous avez un accès Internet, il se peut que vous disposiez déjà d'un espace serveur gratuit pour votre site. L'adresse de votre espace serveur est alors probablement quelque chose comme « <http://home.fournisseur.com/~identifiant> ». Il faudra peut-être l'activer d'abord. Renseignez-vous à ce sujet dans les documents fournis par votre fournisseur d'accès Internet, ou dans leurs pages d'assistance.

Une autre option consiste à prendre un espace serveur gratuit sur Internet. Comme pour la mise en place d'un compte e-mail (par exemple, chez Hotmail), **vous pouvez vous inscrire pour de l'espace serveur libre sur Internet.** Plusieurs sociétés proposent un tel service, entre autres [Angelfire](#) (cliquez sur « Sign Up » et sélectionnez l'adhésion gratuite), cela prendra seulement quelques minutes pour s'inscrire.

Pour accéder au serveur, vous aurez besoin de connaître son nom d'hôte (N.d.T. Host Name), par exemple <ftp.angelfire.com>, et d'avoir votre nom d'utilisateur (N.d.T. username) et votre mot de passe (N.d.T. password) sous la main.

C'est tout ce qu'il faut ?

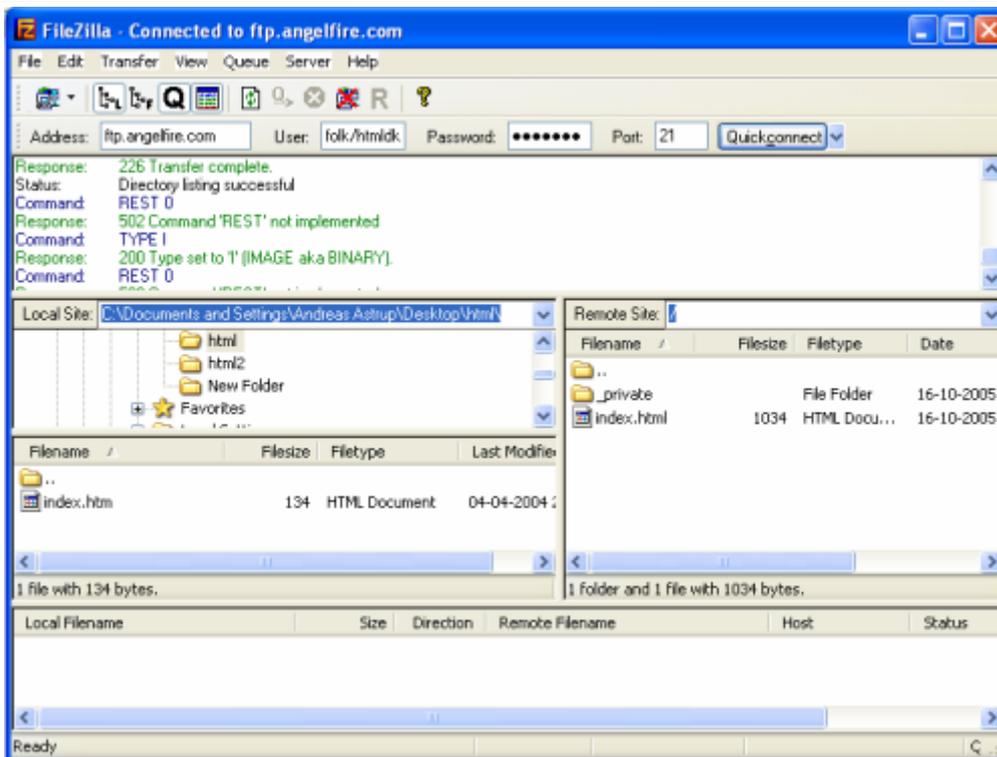
Pour accéder au serveur et mettre en place vos pages, vous aurez également besoin d'un logiciel FTP. Il se peut que vous n'ayiez pas encore de logiciel de ce type, mais heureusement on peut en télécharger des gratuits.

Il existe beaucoup de logiciels FTP différents. L'un des meilleurs est FileZilla, qui est entièrement gratuit. Vous pouvez ainsi télécharger FileZilla à filezilla.sourceforge.net.

Et comment est-ce que j'installe les pages ?

On décrit ci-dessous comment charger les pages sur un compte gratuit chez Angelfire avec FileZilla. Toutefois, la procédure est plus ou moins la même pour tous les fournisseurs d'accès et tous les logiciels FTP.

Lancez le logiciel FTP une fois connecté sur Internet. Inscrivez le nom d'hôte ("ftp.angelfire.com" à « Address »), le nom d'utilisateur (à « User ») et le mot de passe (à « Password ») puis cliquez « Connect ». Vous devriez maintenant avoir accès au serveur. D'un côté de la fenêtre du logiciel, vous pouvez voir le contenu de votre ordinateur (« Local Site »), et de l'autre côté, le contenu du serveur (« Remote Site ») :



Repérez les documents HTML et les images sur votre ordinateur (côté « Local Site ») et transférez-les sur le serveur (côté « Remote Site ») en double-cliquant sur eux. Maintenant le monde entier peut les voir ! (Par exemple, à l'adresse <http://www.angelfire.com/folk/htmlnet/page1.htm>).

Nommez une des pages "index.htm" (ou "index.html") et elle deviendra automatiquement la page d'entrée, c'est-à-dire que si vous tapez <http://www.angelfire.com/folk/htmlnet> (sans nom de fichier), vous ouvrirez en réalité <http://www.angelfire.com/folk/htmlnet/index.htm>.

À plus long terme, il sera peut-être préférable d'acquérir un nom de domaine propre (tel que www.votre-nom-de-domaine.com) pour éviter les adresses longues et compliquées attribuées par votre fournisseur d'accès, ou des fournisseurs d'espace serveur gratuit. On peut trouver et acheter des noms de domaines, par exemple, chez [Speednames](http://www.Speednames.com) ou [NetworkSolutions](http://www.NetworkSolutions.com).

Leçon 14 : Les standards du Web et la validation

Dans cette leçon, vous renforcerez un peu plus votre connaissance théorique de HTML.

Qu'y a-t-il d'autre à connaître à propos de HTML ?

On peut coder en HTML de plusieurs façons. Et les navigateurs peuvent interpréter HTML d'autant de façons. On pourrait dire que **HTML comporte beaucoup de dialectes**. C'est la raison pour laquelle les sites Web semblent différents sur des navigateurs différents.

Presque dès le début d'Internet, il y a eu des tentatives pour élaborer un standard commun de HTML par le biais du [World Wide Web Consortium \(W3C\)](#) fondé par Tim Berners-Lee (oui ! le type formidable qui a inventé HTML). Mais le chemin fut long et difficile.

À l'époque, quand les navigateurs étaient payants, le navigateur Netscape dominait. Les standards HTML s'appelaient alors 2.0 et 3.2. Mais avec une part de marché de plus de 90 %, Netscape n'était pas obligé de tant se soucier (et ne se soucia pas) de standards communs. Au contraire, Netscape inventait ses propres éléments étranges, qui ne fonctionnaient pas sur d'autres navigateurs.

Longtemps, Microsoft ignora presque complètement Internet. Après un temps, la société entra en compétition avec Netscape et introduisit un navigateur. Les premières versions du navigateur de Microsoft, Internet Explorer, n'étaient pas meilleures que Netscape dans le respect des standards HTML. Puis Microsoft choisit de distribuer son navigateur gratuitement (une option toujours très acclamée) et Internet Explorer devint bientôt le navigateur le plus populaire.

À partir des versions 4 et 5, Microsoft chercha à soutenir de plus en plus les standards HTML du W3C. Netscape n'entreprit pas de développer de nouvelle version de son navigateur et continua à en distribuer la version 4 dépassée.

Le reste appartient à l'histoire. Aujourd'hui, les standards HTML se nomment 4.01 et XHTML. Et c'est au tour d'Internet Explorer d'avoir une part de marché de 90 %. Internet Explorer a toujours ses propres éléments étranges mais utilise aussi les standards HTML du W3C. Tout comme d'autres navigateurs tels que Mozilla, Opera et Netscape.

Donc si votre code HTML respecte les standards du W3C, vous rendez vos sites Web lisibles par tous les navigateurs, que ce soit aujourd'hui et à l'avenir. Et par chance, ce que vous avez appris dans ce tutoriel est une version de HTML nouvelle plus stricte et plus propre appelée XHTML.

Super ! Puis-je l'annoncer à chacun ?

Avec tous les types différents de HTML, il est nécessaire de dire au navigateur quel « dialecte » HTML est utilisé, dans ce cas XHTML. Pour ce faire, on utilise une déclaration de type de document. La déclaration de type de document se place toujours en haut du document :

Exemple 1 :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="fr">

<head>
<title>Titre</title>
</head>

<body>
<p>texte texte</p>
</body>

</html>
```

Hormis la déclaration de type de document (la première ligne dans l'exemple précédent), qui indique au navigateur votre choix de XHTML, il est nécessaire de fournir des informations spéciales dans la balise `html` au moyen des deux attributs `xmlns` et `lang`.

Le terme « `xmlns` » est l'abréviation de « XML-Name-Space » (N.d.T. espace de nommage XML), et il devrait toujours avoir la valeur <http://www.w3.org/1999/xhtml>. C'est tout ce qu'il faut savoir. Si vous êtes assoiffé de connaissances compliquées, vous pouvez lire plus à propos des espaces de nommage sur le [site Web du W3C](#).

Dans l'attribut `lang`, vous indiquez la langue dans laquelle le document est écrit. À cet effet, on utilise le [standard ISO 639](#), qui liste les codes de toutes les langues du monde. Dans l'exemple ci-dessus, la langue indiquée est le français ("fr").

Avec la définition de type de document (DTD) de la déclaration, le navigateur sait exactement comment interpréter et afficher votre code HTML. De fait, utilisez l'exemple précédent comme gabarit pour tous vos futurs documents HTML.

La définition de type de document revêt également de l'importance pour la validation des pages.

Valider !? Pourquoi le faire et comment ?

Insérez une déclaration de type de document dans vos pages afin de toujours pouvoir corriger les erreurs de votre code HTML à l'aide du [validateur gratuit du W3C](#).

Pour un test, fabriquez une page et mettez-la en place sur Internet. Ensuite allez à validator.w3.org et tapez l'adresse (URL) de votre page et validez-la. Si votre HTML est correct, vous obtiendrez un message de félicitation. Sinon vous aurez un rapport d'erreurs indiquant exactement en quoi et où vous vous êtes trompé. Faites exprès des erreurs pour voir ce qui arrive.

Le validateur n'est pas seulement utile pour trouver les erreurs. Certains navigateurs compensent les défaillances des développeurs Web en essayant de réparer les erreurs du code HTML et en affichant les pages comme ils estiment qu'elles devraient

apparaître. Avec eux, il se peut que vous ne voyiez jamais d'erreurs dans le navigateur. Par contre, les autres navigateurs feront peut-être des estimations différentes et n'afficheront pas du tout la page. Le validateur peut vous aider à trouver des erreurs dont vous ignoriez même l'existence.

Validez toujours vos pages pour vous assurer qu'elles s'afficheront toujours correctement.

Leçon 15 : Les dernières astuces

Félicitations, vous êtes à la dernière leçon.

Donc maintenant je sais tout ?

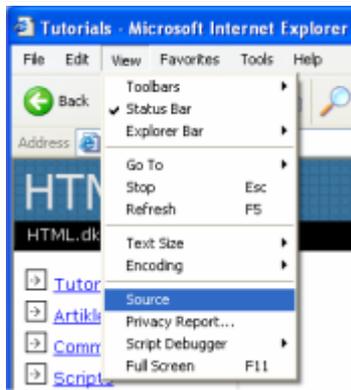
Vous avez beaucoup appris et vous êtes capable de fabriquer vos propres sites Web ! Néanmoins, vous ne connaissez que les bases et il y a encore beaucoup à maîtriser. Mais les fondations sont solides.

Dans cette dernière leçon, vous trouverez quelques astuces finales :

- Premièrement, c'est une bonne idée que de conserver l'ordre et la structure dans vos documents HTML. En chargeant des documents bien arrangés, vous montrerez non seulement votre maîtrise de HTML mais aussi faciliterez considérablement votre vision d'ensemble.
- Collez aux standards et validez vos pages. On ne le dira jamais assez : toujours écrire du XHTML propre, utiliser une déclaration de type de document et valider vos pages à validator.w3c.org.
- Donnez du contenu à vos pages. Rappelez-vous que HTML est un outil qui vous permet de présenter des informations sur Internet, donc assurez-vous qu'il y ait de l'information à présenter. Les belles pages peuvent être jolies mais la plupart des personnes utilise Internet pour y trouver de l'information.
- Évitez de surcharger vos pages avec des images lourdes et autres fantaisies trouvées sur Internet. Cela ralentit le chargement des pages et pourrait embrouiller les visiteurs. Les pages qui mettent plus de 20 secondes à charger sont susceptibles de perdre jusqu'à 50 % des visiteurs.
- Rappelez-vous d'ajouter votre site Web dans les moteurs de recherche/annuaires pour que les personnes n'appartenant pas à votre famille proche puissent les trouver et les apprécier. Sur la page d'accueil de tous les moteurs de recherche, vous trouverez un lien pour ajouter de nouvelles pages (le plus important est Google, mais il y en a d'autres tels que DMOZ, Yahoo, AltaVista, AlltheWeb et Lycos).
- Dans ce tutoriel, vous avez employé « Bloc-notes », qui est un éditeur simple d'utilisation très facile, mais il sera peut-être pratique d'utiliser un éditeur plus évolué offrant une meilleure vue d'ensemble et plus de possibilités. Vous trouverez un résumé et des critiques de plusieurs éditeurs sur Download.com.

Comment s'améliorer ?

Avant tout, il importe de continuer à travailler et expérimenter avec les choses apprises dans ce tutoriel. Étudiez les sites d'autres personnes et, si besoin, voyez comment ils sont faits avec « Voir source » (cliquez « Affichage » dans le menu de votre navigateur et sélectionnez « Source »).



Cherchez sur Internet des exemples et articles sur HTML. Il y a beaucoup de sites avec un bon contenu sur HTML.

Lisez les nombreux forums de discussion sur Internet et posez-y des questions. Deux des meilleurs endroits où aller sont [Newsgroups](#) et [Experts Exchange](#). Vous y cotoierez les experts véritables de qui vous apprendrez énormément.