

## De quoi est faite une page Web?

Une page Web telle qu'on la voit à l'écran contient du texte, des images, parfois des sons et des animations. Mais la page Web que vous voyez quand vous la consultez n'existe pas telle quelle sur le serveur d'où elle vient. Le serveur, lui, ne connaît que le fichier de code HTML qui sert à bâtir la page.

En effet, toute page Web n'est en réalité qu'une série de codes, appelé HTML (pour *Hyper Text Markup Language*). Le serveur contient un fichier de codes et c'est ce fichier qui est envoyé à votre fureteur. Le fureteur **interprète** ensuite ces codes pour **recréer** la page Web sur votre écran. C'est le HTML qui voyage.

Par exemple, le site du Collège Lionel-Groulx, à l'adresse [www.clg.qc.ca](http://www.clg.qc.ca), a l'air de ceci au moment d'écrire ces notes de cours:

Le fichier HTML qui a été envoyé à votre fureteur pour que vous puissiez voir la page a l'air de ceci (un court extrait du fichier total):

```

<div id="menu">
  <div id="menu-inner">
    <ul>
      <li><a href="/futur-etudiant/">Futur étudiant</a></li>
      <li><a href="/etudiant/">Étudiant</a></li>
      <li><a href="/programmes/">Programmes</a></li>
      <li><a href="/vie-etudiante/">Vie étudiante</a></li>
      <li><a href="/le-college/">Le Collège</a></li>
      <li><a href="http://www.formationcontinue.clg.qc.ca/"
target="/_blank/">Formation continue et services aux entreprises -
Plastigroulx</a></li>
      <li><a href="/international/">International</a></li>
      <li><a href="/bibliotheque/">Bibliothèque</a></li>
    </ul>
  </div><!-- /menu-inner -->
</div><!-- /menu -->

```

```

<div id="featured">
  <div id="featured-inner-env">
    <div id="featured-inner">

      <div id="featuredContent"><div id="c846" class="csc-default"
><div class="slide">
  <div class="featured-image">

    </div>

    <div class="featured-content">
      <h2>Félicitations au Big Band de Serge Arsenaault </h2>

      <p class="bodytext">Le Big Band de Lionel-Groulx a remporté l'or au
Jazzfest des jeunes du Québec qui avait lieu à l'École André Laurendeau à St-Hubert
le 2 avril dernier. Le Big Band de Serge Arsenaault, enseignant du cours Musique
d'ensemble, concourrait dans la catégorie des cégeps et des universités (A-
101).&nbsp; L'ensemble est composé d'étudiants de différentes cohortes et
d'étudiants musiciens bénévoles.&nbsp; FÉLICITATIONS À TOUS! </p>

      <div class="featured-bt">

        </div>
      </div>
    <div class="clear"></div>
  </div></div><div id="c803" class="csc-default" ><div class="slide">
  <div class="featured-image">

    </div>

    <div class="featured-content">
      <h2>Demandes de 2e tour acceptées jusqu'au 20 avril en musique classique
préuniversitaire</h2>

      <p class="bodytext">Possibilité de cinq Doubles DEC : Sciences de la
nature, Arts et lettres, profil Lettres ou Arts d'interprétation, Sciences humaines
profil Mathématiques ou Éducatons et apprentissages. </p>

      <div class="featured-bt">

        <a href="http://www.clg.qc.ca/programmes/programmes-
preuniversitaires/musique/"></a>

        </div>
      </div>
    </div>
  </div>

```

On peut voir le texte du site caché un peu partout parmi un fouillis de codes bizarres... Ces codes bizarres sont en fait des balises HTML (également appelées *tags* ou *étiquettes*) qui disent à votre fureteur comment positionner le texte et les images, quel format et taille de caractères utiliser, de quelles couleurs il doit se servir, etc. Ce sont ces codes qui permettent au fureteur de créer la page qu'il vous affiche.

***Doit-on absolument comprendre et écrire tous ces codes pour créer un site Web?***

Heureusement, non. Il y a fort longtemps, c'était à peu près la seule solution. De nos jours, il existe tout un tas d'outils qui peuvent générer du HTML pour nous – tout ce qu'on a à faire c'est de taper du texte et faire de la mise en page.

Toutefois, c'est un grand atout que de comprendre les bases du HTML. Pourquoi? Premièrement, pour ne pas être totalement dépendant d'un outil qui a ses forces, oui, mais aussi ses faiblesses (bugs, fonctionnalités mal ou non implémentées). Deuxièmement, une bonne façon d'apprendre à faire des pages Web est de regarder comment d'autres sites ont été conçus, ce qui est impossible si on ne comprend pas le HTML. Et troisièmement, parce que si on connaît le HTML, on comprend mieux les limites dans lesquelles on travaille avec l'outil et la structure que l'on doit utiliser.

## Petite histoire du HTML

---

Le HTML proprement dit a vu le jour quelque part au début des années 90, fruit de longues discussions de chercheurs qui voulaient mettre en place un protocole commun utilisé par tout le monde sur le Web. Le HTML est basé sur le principe de l'hypertexte : un document texte contenant des liens qui, lorsqu'ils sont cliqués, amènent le lecteur à d'autres documents. Le concept d'hypertexte ne date pas d'hier : il a été imaginé dans les années 1940 par un certain Vannevar Bush. Il avait imaginé une machine purement hypothétique appelée le "memex", qui pourrait améliorer la mémoire humaine en permettant à l'utilisateur de conserver et de retrouver des fichiers par association. Même si Bush est mort bien avant la naissance du Web, beaucoup le considèrent comme le père du Web puisque c'est son concept qui en a servi de base.



L'idée de l'hypertexte est bonne. Elle nous permet de lire un texte, puis d'aller consulter d'autres documents pertinents à la section qu'on est en train de lire si on le désire. Le principe de "sauter d'un texte à l'autre" pour approfondir un sujet est une approche beaucoup plus rapide pour l'apprentissage que de simplement lire tout un tas de documents un après l'autre et de faire des liens nous-mêmes après coup.

La toute première version du HTML a été mise en place en 1990 par le CERN (Centre Européen de Recherche Nucléaire). Elle n'était qu'une version préliminaire (baptisée après coup "version 0"). Le CERN a donc donné naissance au Web, en concevant le projet, en créant le langage qui y serait utilisé et en développant le tout premier navigateur.

En 1993, la version HTML 1.0 est publiée par l'IETF (*Internet Engineering Task Force*). La version 1.0 était très simple et permettait simplement d'afficher du texte à l'écran, avec un formatage limité, sans interactions. Elle tentait simplement d'officialiser et de standardiser le HTML version 0. Cette même année, le NCSA (*National Center for Supercomputing Applications*) lance son navigateur, le Mosaic, pour les PC, les MacIntosh et les serveurs X-Windows.

En 1994, Marc Andreessen de NCSA et le fondateur de Silicon Graphics James Clark forment Netscape. Ils lancent la version 1.0 du Netscape Navigator, version qui supporte une suite de balises HTML qui ne font pas partie du HTML standard.

Cette année-là, il y avait environ 3000 sites existant sur le Web... Un an plus tard, ce nombre va doubler, pour ensuite être décuplé l'année d'ensuite.

En 1995, le HTML 2.0 (également créé par le IETF) ajoute la possibilité d'insérer des images, des listes et des formulaires (permettant à un visiteur sur le site d'envoyer de l'information vers le serveur). Les sites Web sont plus attrayants et offrent plus de possibilités. Netscape 1.2 est distribué, incluant une version pour Windows 3.1 et Windows 95. Cette même année, Microsoft lance Internet Explorer et décide de l'incorporer à Windows 95, avec les conséquences juridiques que l'on connaît.

Le Web gagne vite en popularité : en 4 ans il a convaincu 50 millions d'utilisateurs dans le monde (à titre de comparaison, la radio a eu besoin de 38 ans et la télé 13 ans pour réussir le même exploit).

Comme de plus en plus de gens connaissent maintenant le Web, beaucoup de designers de sites demandent plus du HTML. C'est là que les problèmes ont commencé. À cette époque, Netscape était le navigateur le plus répandu. Ses concepteurs ont donc décidé de prendre les choses en mains et de créer des balises HTML spéciales pour Netscape, ce qui permettait à Netscape d'afficher des sites Web plus complexes et attrayants. Le problème est évident : si un site Web utilise ces balises, il doit être vu avec Netscape, sinon la mise en page peut être complètement différente (ou le site peut carrément ne pas fonctionner). Les ajouts de Netscape sont intéressants, mais bien des développeurs Web ne veulent pas limiter leur public à un type de navigateur...

De leur côté, les autres navigateurs tentent tant bien que mal d'implanter des versions qui supportent les balises étendues de Netscape, mais le font parfois mal ou en oublient quelques-unes... Le besoin d'un standard établi par une source externe se fait sentir.

L'IETF tente de standardiser HTML en créant un HTML 3.0. Cette nouvelle version standardisait plusieurs fonctionnalités des balises Netscape et donnait encore plus de possibilités aux créateurs de sites Web. Malheureusement, les navigateurs mettent du temps à prendre en charge ces nouvelles balises, ou en choisissent quelques-unes et laissent faire les autres... À cause de cela, le HTML 3.0 est abandonné. Des balises spécifiques à certains navigateurs continuent de faire surface, créant carrément une scission entre les utilisateurs (Netscape vs Internet Explorer).

Le W3C (*World Wide Web Consortium*) décide de corriger la situation en créant HTML 3.2 en 1996 (son premier mandat). Un ménage est fait dans la grande quantité de balises propres à Netscape ou à Microsoft et la majorité d'entre elles ne feront pas partie du standard. Le HTML 3.2 a été créé de façon modulaire, permettant aux navigateurs d'être modifiés une partie à la fois sans causer de gros problèmes (et en mettant tout le monde sur le même pied d'égalité). La version 3.2 permet de faire des liens sur une image et supporte le Java et le Javascript – elle devient vite populaire et bientôt tous les navigateurs la supportent.

En 1997, le W3C sort le HTML 4.0, qui supporte maintenant les cadres (*frames*), les feuilles de style et l'intégration d'éléments multimédia. C'est à peu de chose près le HTML qui est utilisé partout aujourd'hui. La version 4.01, lancée en fin de 1999, n'apporte que des corrections minimales à la version 4.0.

Le HTML est resté le même pendant très, très longtemps à partir de là. On disait même à la fin des années 90 qu'il n'y aurait jamais d'autres versions et que l'avenir serait dans le XHTML, un dérivé du langage.

En fin de compte, bien que le XHTML a effectivement existé et a évolué jusqu'en 2009, il n'a jamais été utilisé formellement. La toute dernière version du HTML, le HTML 5, a été lancée en 2014, après un long processus de création et d'évolution qui avait commencé en 2008.

## ***Le HTML est notre ami (il ne faut pas en avoir peur)***

Bien que l'exemple de code HTML montré plus haut est plus effrayant qu'attirant, il ne faut pas se fier à cette première impression et décider de ne jamais apprendre le langage.

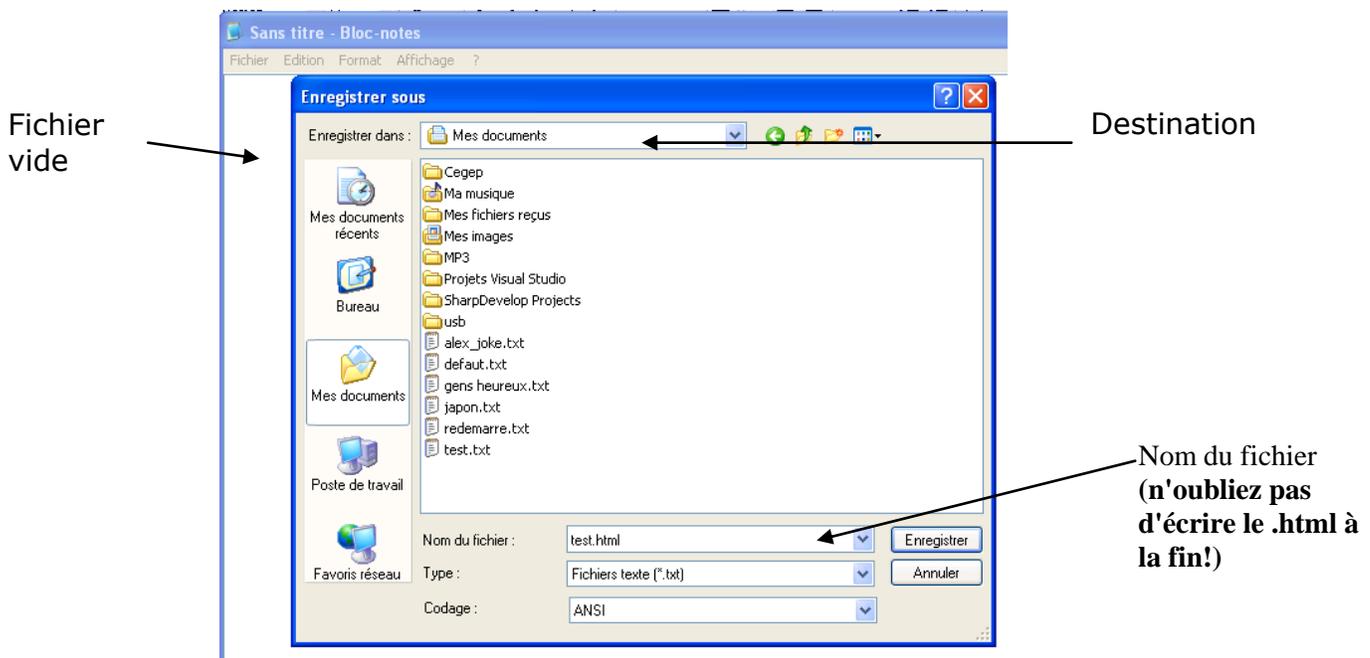
Le HTML est très structuré, avec peu d'exceptions, et une fois qu'on maîtrise les principes de base il devient beaucoup plus clair. De là, le reste n'est que mémorisation de balises et de leurs fonctions...

### ***Un premier test***

Nous allons créer un fichier HTML (donc une page Web) à l'aide du bloc-notes de Windows. Démarrez le bloc-notes (qui se trouve normalement dans le menu Démarrer, dans la section "Tous les programmes" puis dans "Accessoires").

Commençons par créer un fichier entièrement vide. N'écrivez rien dans le bloc-notes et sauvegardez ce fichier en faisant Fichiers->Enregistrer Sous.

Placez-le dans "Mes Documents". Appelez-le `test.html` puis cliquez sur le bouton Enregistrer.



Gardez le bloc-notes ouvert, nous allons y revenir. Allons maintenant dans Internet Explorer et ouvrons cette page Web pour qu'il nous l'affiche. Allez dans Fichier->Ouvrir, cliquez sur le bouton "Parcourir" et retrouvez `test.html`.

À ce stade-ci, pas de surprise : on se retrouve devant une page blanche.

Maintenant retournez dans le bloc-notes et écrivez la ligne suivante dans le texte:

```
<title>Bienvenue au cours de HTML</title>
```

**Enregistrez le fichier.** Retournez dans Internet Explorer et cliquez sur le bouton "Actualiser" (rappelez-vous, ce bouton force Internet Explorer à aller rechercher une copie de la page afin de voir les dernières mises à jour).

Pouvez-vous remarquer la différence?

Maintenant, ajoutons un peu de texte à notre page, sur le bloc-notes:

```
<title>Bienvenue au cours de HTML</title>
```

```
Ceci est ma première page Web.
```

Encore une fois, enregistrez le fichier du bloc-notes puis retournez actualiser la page sur Internet Explorer. Voilà! Vous venez d'écrire votre première page Web et du même coup votre première balise HTML!

### *Qu'est-ce qu'une balise HTML?*

Une balise est une expression placée entre crochets « plus petit/plus grand ». Par exemple:

```
<title>
```

Toute balise respecte toujours ce format. La grande majorité des balises viennent par paires : une "de début" et une "de fin". La seule différence entre les deux est que la balise de fin commence par une barre oblique:

```
<title>
</title>
```

Le texte qui est écrit entre ces deux balises est affecté d'une façon quelconque, selon la balise. Par exemple, la balise `center` centre le texte qui se trouve entre la balise de début et la balise de fin. Par exemple, essayez ceci:

```
<center> Ce texte sera centré. </center>
```

La balise de fin n'est pas obligée de se trouver sur la même ligne que la balise de début. À ce moment-là, tout le texte qui se trouve entre les deux balises sera affecté. Essayons ceci:

```
<center> Ce texte sera centré.
Ce texte aussi.
Et même ce texte.
</center>
```

Que constatez-vous de surprenant en essayant ce dernier exemple?

Eh oui, c'est un point important à ne pas oublier en HTML : comme c'est un langage de formatage de texte, le format original du texte est ignoré. Seules les balises ont un effet réel sur le format, même quand il s'agit de revenir à la ligne!

Modifions l'exemple précédant pour que chaque ligne de texte dans le fichier soit écrite sur une ligne différente dans la page Web. Pour ce faire, il suffit d'ajouter des balises `<br>`. Cette balise est spéciale car elle fait partie des quelques exceptions qui n'ont pas besoin d'être ouvertes puis refermées avec une balise de fin. On peut la mettre seule quelque part et à cet endroit un retour à la ligne sera fait. Par exemple:

```
<center> Ce texte sera centré.<br>
Ce texte aussi.<br>
Et même ce texte.<br>
</center>
```

On remarque en passant qu'on peut très bien imbriquer une balise dans une autre. Dans notre dernier exemple, les balises `<br>` sont placées entre les deux balises `<center>` et `</center>` et cela ne pose aucun problème. En fait c'est même très courant – on voudra souvent appliquer deux ou plusieurs caractéristiques de formatage à un texte.

Si on veut passer plusieurs lignes d'un seul coup, on peut placer plusieurs `<br>` de suite.

### ***La balise <p>***

La balise `<p>` est similaire à la balise `<br>`: elle sert à changer de paragraphe. Lorsqu'un navigateur rencontre une balise `<p>`, il fait un retour de chariot et saute une ligne supplémentaire. Le `<p>` est donc équivalent à `<br><br>`.

Toutefois, à partir du HTML 3.0 on a proposé que la balise `<p>` respecte les standards du HTML et doive être ouverte et refermée: `<p>comme ceci</p>`. Ce n'est pas obligatoire et ce n'est pas toujours respecté, malheureusement. On doit toutefois le faire si on veut modifier l'alignement du paragraphe, ce que l'on verra plus tard.

On va donc prendre la bonne habitude d'entourer ses paragraphes de `<p>` et `</p>`.

### ***Les sections d'une page Web***

Une page Web est normalement constituée de deux sections: la tête (*head*) et le corps (*body*).

Dans la tête, on retrouvera des informations qui ne sont pas affichées dans la page elle-même, des informations **sur** le site. On y mettra par exemple le titre de la page (qui sera affiché dans la barre de titre du navigateur, grâce à la balise `<title>`) et d'autres paramètres utiles pour le site Web (que nous ne couvrirons pas pour l'instant).

Le corps de son côté contiendra tout le reste, soit la page Web elle-même, **le contenu** du site.

Même si ce n'est pas nécessaire pour que la page soit fonctionnelle, c'est une **règle de bonne programmation** que d'utiliser les balises `<head>` et `<body>` pour indiquer les deux sections de votre page Web. N'oubliez pas que vous pouvez imbriquer les balises, donc vous pouvez faire ceci:

```
<head>

<title>Bienvenue au cours de HTML</title>

</head>
```

```

<body>

<center> Ce texte sera centré.<br>
<p>Ce texte aussi.</p>
Et même ce texte.<br>
</center>

</body>

```

Même si vous pouvez théoriquement entrer tout ce texte sur une seule et même ligne et obtenir le même résultat, voici d'autres bonnes règles de programmation:

- Mettez les balises <head>, </head>, <body> et </body> seules sur leur ligne.
- Laissez quelques lignes vides entre les deux sections.
- Lorsque vous utilisez une balise <br>, faites un retour à la ligne dans le texte.

Comparez le code ci-dessus avec le suivant, qui enfreint toutes ces règles:

```

<head><title>Bienvenue au cours de HTML</title></head>
<body><center>Ce texte sera centré.<p>Ce texte
aussi.</p>Et même ce texte.<br></center></body>

```

Ce dernier exemple de code est beaucoup plus difficile à comprendre, donc beaucoup plus difficile à modifier. Mais il produit le même résultat.

### *La balise générale <html>*

Toujours dans les règles de bonne programmation, vous devriez toujours inclure l'ensemble de votre page Web (les deux sections au grand complet) à l'intérieur d'une balise <html>. Ceci définit sans équivoque que votre texte est écrit en HTML. Notre exemple devient donc:

```

<html>
<head>

<title>Bienvenue au cours de HTML</title>

</head>

<body>

<center> Ce texte sera centré.<br>
<p>Ce texte aussi.</p>
Et même ce texte.<br>
</center>

</body>
</html>

```

## Les liens en HTML

---

À la base de l'hypertexte, il y a bien sûr les liens – voilà un autre concept simple et utile à connaître en HTML. Commençons par un exemple de lien:

```
Pour visiter le site de Jan Terri,
cliquez <a href="https://janterril.bandcamp.com/">ici</a>.
```

Ce petit bout de code HTML produira l'effet suivant sur un navigateur:

Pour visiter le site de Jan Terri, cliquez [ici](https://janterril.bandcamp.com/).

Quelques conclusions:

- Pour faire un lien, on utilisera la balise `<a> . . . </a>`. Le "a" est l'abréviation de "*anchor*", qui signifie "ancre". Le lien est une ancre accrochée à une autre page (quelle belle image)...
- Comme pour la majorité des autres balises, la balise `<a>` a besoin d'être ouverte puis fermée plus loin. Le texte qui se trouve entre les deux deviendra le lien lui-même. On peut faire un lien sur un mot, une phrase ou même un paragraphe complet. Dans notre exemple, le mot "ici" est entre le `<a>` et le `</a>`, c'est donc ce simple mot qui servira de lien.
- En règle générale, un lien est affiché en **bleu et souligné** lorsqu'il n'a pas encore été visité, puis il devient **mauve et souligné** lorsqu'il l'a été. Toutefois, **ce comportement dépend du navigateur** et des propriétés choisies par l'utilisateur. Il existe un moyen de modifier la couleur des liens, mais certains navigateurs utilisent leurs couleurs en tout temps et ignorent les vôtres. Quant au soulignement, il est entièrement dû au navigateur. N'oubliez jamais que **le HTML est interprété par le navigateur** et si quelqu'un veut voir des liens en orange et sans souligné, il peut le faire que vous le vouliez ou non.
- La balise `<a>` n'est pas un simple `<a>`, il y a autre chose à l'intérieur. On appelle cela un paramètre. Un paramètre sert à donner une information supplémentaire à la balise. Dans le cas du lien, l'information à donner est bien sûr la page vers laquelle le lien mènera.

### Les balises à paramètres

Certaines balises acceptent un ou plusieurs paramètres, qui définiront de façon plus précise leur comportement. Lorsqu'un paramètre est ajouté à une balise, il devra être placé à l'intérieur des crochets (`<` et `>`) et après le nom de la balise. Au moins un espace devra séparer le nom de la balise et le paramètre. Par exemple:

```
<a href="http://www.google.com">
```

Nom de la balise                      Paramètre

Ceci par contre ne fonctionnera pas, puisqu'il n'y a pas d'espace entre le nom de la balise et le paramètre:

```
<a href="http://www.google.com">
```

Le navigateur qui recevrait une telle balise ne la comprendrait pas et l'ignorerait.

Le paramètre lui-même est composé de trois parties:

- Un nom de paramètre (dans notre exemple `href`, qui signifie "*Hypertext Reference*")
- Un symbole d'égalité (=)
- Une valeur (dans notre exemple "`http://www.google.com`")

Ces trois parties peuvent être séparées par des espaces, mais ce n'est pas obligatoire.

Cette façon de faire permet à une balise de recevoir plusieurs paramètres, chacun d'entre eux identifié par son nom, et chacun d'entre eux recevant une valeur. Notre exemple n'en contient qu'un, mais la syntaxe reste la même.

Les guillemets autour de la valeur sont obligatoires si la valeur contient un espace (ce qui ne sera jamais le cas pour une adresse Web), sinon ils sont facultatifs. Toutefois c'est une bonne habitude de les utiliser systématiquement, ça évite de devoir se poser des questions et élimine les risques d'erreurs.

### *La référence hypertexte* (`href`)

Il est possible de faire pointer un lien vers un autre site Web (comme dans notre exemple), mais on peut également le faire pointer vers une autre page à nous, qui se trouve sur le même serveur.

C'est pourquoi, lorsque vous faites un lien vers un autre site Web, vous devez absolument débiter l'adresse par "`http://`" (ou "`https://`", pour utiliser la version sécurisée du protocole) de façon à indiquer au navigateur qu'il devra aller chercher une autre adresse, en utilisant le protocole HTTP (le protocole du Web). Sinon, il cherchera un autre fichier à vous sur votre serveur.

Toutefois, savoir faire un lien vers une autre page de notre propre site est également très utile... Créons une nouvelle page HTML que l'on appellera `destination.html`. Mettons-y une simple phrase:

```
Vous êtes arrivé à destination!
```

Créons ensuite une autre page que l'on appellera `source.html`. Celle-ci contiendra un lien vers la page de destination:

```
Pour vous rendre à destination,  
<a href="destination.html">cliquez ici!</a>
```

Ouvrons la page `source.html` dans Internet Explorer et voyons ce que ça donne!

Vous remarquerez au passage:

- Cette fois-ci, notre lien est fait sur les deux mots "cliquez ici!"

- Nous n'avons pas utilisé le préfixe "http://" dans la référence, dans le but intentionnel de le faire atteindre un autre fichier à nous. Ce fichier doit se trouver dans le même répertoire que la source, sinon le lien ne fonctionnera pas.

On pourrait essayer de déplacer le fichier `destination.html`, question de tester ce que ça donne. Dans votre répertoire "Mes Documents" (où devraient se trouver les deux fichiers HTML que l'on vient de créer), créez un nouveau répertoire appelé `Rep2`. Déplacez-y `destination.html`, puis réessayez votre page `source.html` dans Internet Explorer.

Pas de surprise, notre lien ne fonctionne plus. Selon vous, que pourrait-on faire pour "arranger" notre lien si on ne voulait pas déplacer les fichiers HTML? Vous pouvez expérimenter pour trouver la solution...

Ceci nous permet d'arriver à une conclusion importante : mieux vaut penser d'avance à notre organisation de fichiers sur un site Web, puisque de changer des fichiers de place risque de briser des liens que l'on devra réparer.

### *Lien vers une adresse de courriel*

On peut également utiliser la balise `<a>` pour faire un lien sur une adresse de courriel. Lorsqu'on cliquera sur le lien, notre logiciel de courriel par défaut sera démarré et un nouveau courriel sera créé pour nous, avec l'adresse du destinataire déjà entrée!

Pour ce faire, il suffit de demander à l'ancre d'utiliser le protocole "mailto" plutôt que le protocole "http". Par exemple:

Contactez le `<a href="mailto:etienne.forest@gmail.com">webmestre</a>`

Ici, on donne comme référence hypertexte (`href`) l'adresse courriel du destinataire, en spécifiant bien d'utiliser "mailto" comme protocole (remarquez qu'on ne doit pas mettre de "://" après les deux points comme on le faisait avec `http`). Si vous oubliez le "mailto", le navigateur cherchera un fichier appelé "etienne.forest@gmail.com" sur votre serveur et ne le trouvera pas.

### *Les erreurs courantes*

Voici quelques erreurs communément rencontrées par les débutants:

Erreur	Résultat
Oubli du "http://" ou du "mailto:" dans la référence.	Le navigateur cherchera un fichier local plutôt qu'un autre site ou une adresse de courriel. Comme il n'en trouvera pas, il affichera un message d'erreur du style "Impossible de trouver la page demandée".
Les guillemets ont été ouverts avant la référence mais ils n'ont jamais été fermés.	La balise <code>&lt;a&gt;</code> (et tout le texte qu'elle contient) sera carrément affichée sur la page Web et aucun lien ne sera créé.
Utilisation des "://" après le "mailto:" dans une référence de courriel.	Les "://" sont considérés comme faisant partie de l'adresse courriel, ce qui crée un nouveau

	courriel expédiable à (par exemple) <code>//etienne.forest@gmail.com.</code>
La balise de fin ( <code>&lt;/a&gt;</code> ) est oubliée.	Tout le texte de la page devient un énorme lien vers la page de destination.
La balise de fin ( <code>&lt;/a&gt;</code> ) est écrite immédiatement après la balise de début, sans texte entre les deux.	Le lien est invisible et "incliuable" puisqu'il n'y a pas de texte pour servir de lien.

## ***Les images en HTML***

---

L'insertion d'images est le premier moyen qui a été développé en HTML (dès la version 2.0) pour rendre les sites Web plus agréables à l'oeil et plus intéressants.

Insérer une image en HTML est très simple: il suffit d'utiliser la balise `<img>`. C'est une autre balise qui n'a pas besoin d'être fermée, elle peut exister seule, comme le `<br>`. L'image sera insérée dans le texte à l'endroit où se trouve la balise, tout simplement.

Évidemment, la balise `<img>` a besoin de paramètres pour fonctionner. Le plus important est bien sûr le nom (et l'emplacement) du fichier d'image.

Enregistrez l'image du sympathique cochon fourni pour le cours dans votre dossier "Mes Documents" (là où vous avez créé vos pages Web précédentes).

Dans une de vos pages Web, ajoutez le code suivant:

```

```

Vous pouvez ensuite admirer le résultat en ouvrant votre page.

Vous aurez bien sûr compris que le paramètre "src" est la source de l'image. La balise d'image est un excellent exemple de balise à paramètres multiples, puisqu'on pourra ajouter tout un groupe d'autres paramètres si on le désire.

Nous verrons pour l'instant deux autres paramètres importants: `width` (largeur) et `height` (hauteur) qui permettront de redimensionner l'image. Par exemple, modifions notre ligne précédente:

```

```

On constate que l'image prend maintenant moins de place à l'écran. Toutefois, quelques considérations importantes:

- Le fait d'utiliser `width` et `height` ne change que les dimensions d'affichage à l'écran. L'image transférée est aussi grande qu'avant, mais elle est rapetissée en temps réel à l'affichage par votre navigateur. Autrement dit, utiliser ces paramètres pour diminuer une image ne diminuera pas la durée du téléchargement.
- Les valeurs de largeur et de hauteur sont entrées en pixels. Si vous entrez des dimensions disproportionnées par rapport à l'original, vous vous retrouverez avec une image "écrasée" ou "étirée". C'est dans des cas comme ça que l'utilisation d'un logiciel de création de sites Web est fort appréciée.

- Il faut faire à attention à l'orthographe de `width` et `height` qui n'est pas toujours facile à retenir pour les francophones...
- Lorsque l'on utilise plusieurs paramètres dans une même balise, on les sépare simplement par des espaces.
- Typiquement, on n'utilisera pas de guillemets autour des valeurs des paramètres lorsque celles-ci sont numériques. Par contre ça fonctionne quand même si on en met alors si on le fait ce n'est pas dramatique.

### ***Une image comme un lien***

Utiliser une image comme un lien est très simple : il nous suffit d'utiliser une balise `<a>` comme on l'a déjà fait, mais en plaçant une balise d'image au milieu plutôt que du texte. Par exemple:

```
Pour vous rendre à destination, cliquez sur le cochon!<p>
<a href="destination.html"></a>
```

C'est un exemple classique de balise à l'intérieur d'une autre.

### ***Les couleurs : pas facile en HTML...***

---

On peut mettre une couleur de fond à notre page Web en utilisant le paramètre `bgcolor` ("*background color*" ou couleur de fond) de la balise `<body>` (voilà qui donne une raison d'être à cette balise, autre que pour la simple organisation du document). Toutefois, c'est une opération qui n'est pas aussi facile que le reste.

En effet, l'utilisation des codes de couleurs n'est pas très intuitive. Le paramètre `bgcolor` accepte comme valeur quelque chose comme ceci:

```
<body bgcolor="#FF00FF">
```

Cet exemple donne du violet... Comment comprendre le tout? Il faut connaître le principe du RGB. RGB signifie Red-Green-Blue (Rouge-Vert-Bleu). Le principe est le suivant : toute couleur peut être obtenue en mélangeant le rouge, le vert et le bleu en différentes proportions. Par exemple, pour obtenir du violet, on mélange beaucoup de rouge avec beaucoup de bleu (et pas de vert du tout). On peut faire du blanc si on mélange au maximum les trois couleurs et du noir si on ne met rien.

Chaque couleur est donc représentée par trois valeurs, une pour la quantité de rouge, l'autre pour la quantité de vert, la dernière pour la quantité de bleu. Chaque valeur peut aller de 0 à 255.

Pour compliquer encore plus les choses, ces trois nombres doivent être convertis en hexadécimal! L'hexadécimal est un système de numérotation en base 16 dans lequel on utilise les symboles 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E et F pour représenter les nombres de 0 à 15. À partir de 16, on écrira « un paquet de seize et zéro unité », ce qui donnera 10. À dix-sept, on ajoute une unité et on obtient 11, ainsi de suite jusqu'à trente et un (qui donne 1F), puis à trente-deux on a deux paquets de seize et zéro unité (20).

Ouf, pas mal compliqué tout ça... Ce qui veut dire que notre couleur se retrouve composée de trois valeurs (rouge, vert et bleu) allant chacune de 00 à FF (80 est la valeur moyenne). On colle ces trois valeurs ensemble et on obtient quelque chose comme le FF00FF de notre exemple (un maximum de rouge – 255 ou FF, pas de vert – 00, un maximum de bleu, donc du violet).

Et juste pour compliquer encore plus les choses, on commence par un symbole # (qui est facultatif mais couramment utilisé pour indiquer qu'on utilise le système hexadécimal).

Bref, c'est assez difficile d'obtenir une couleur précise par soi-même, d'un seul coup, sans devoir réfléchir un peu puis procéder par essai-erreur pour finir par tomber sur ce que l'on veut.

Une alternative que l'on peut utiliser pour se simplifier un peu la vie : utiliser un site de type « Color Picker », un sélecteur de couleurs. Par exemple : <http://www.colorpicker.com/>

### ***Et les caractères?***

On peut également changer la couleur des caractères à l'aide de la balise <font>. Cette balise doit être ouverte et fermée car elle affectera uniquement le texte qui se trouve entre les deux. Elle accepte entre autres le paramètre color, qui fonctionne exactement comme le bgcolor de <body>. Par exemple, pour écrire en bleu:

```
<font color="#0000FF">Bleu, bleu, l'amour est bleu!</font>
```

La balise <font> permet également de modifier la police de caractères, avec le paramètre face. Face accepte un nom de police comme valeur (Par exemple, Times New Roman, Courier New, Verdana, Arial...). On peut modifier la taille des caractères avec le paramètre size, qui accepte un nombre de 1 à 7, 1 étant petit et 7 étant grand (la taille par défaut correspond à 3).

```
<font face="Verdana" size=4 color="#00FFFF">Un exemple</font>
```

## ***Atelier formatif***

---

### ***Numéro 1***

Voici quelques balises simples et utiles:

<b> ... </b>	met le texte en caractères gras (bold)
<i> ... </i>	met le texte en italique
<center> ... </center>	centre le texte horizontalement
<u> ... </u>	souligne le texte
 	fait un retour à la ligne
<p>	fait un retour à la ligne et passe une ligne (équivalent à deux   de suite)

Créez une page Web qui aura comme titre "Ma première page Web" (ce titre devra apparaître dans la barre de titre d'Internet Explorer et non dans la page elle-même). Le contenu de la page Web devra imiter le mieux possible l'exemple suivant:

<p><b><u>Bienvenue à ma page Web</u></b></p> <p>Bonjour, je m'appelle <b>Roger</b> et voici ma première page Web!</p> <p>Elle a été réalisée avec le bloc-notes de <i>Windows</i>, quel outil merveilleux.</p> <p style="text-align: center;"><b>À bientôt!</b></p>
---

Vous devez utiliser toutes les règles de bonne programmation que nous avons vues jusqu'à maintenant.

**Conseils:**

- Si une balise ne semble pas fonctionner, c'est généralement parce qu'elle est mal écrite. Une balise que le navigateur ne comprend pas est simplement ignorée.
- À l'opposé, si une balise de fin est mal écrite, le format de la balise sera appliqué sur tout le texte, de la balise de début jusqu'à la fin de la page... Si ce problème se présente, cherchez la balise de fin – soit elle aura été oubliée ou soit elle aura été mal écrite.

**Numéro 2**

Construisez maintenant un site Web un peu plus élaboré ayant pour thème la présentation de vos restaurants préférés. Voici les spécifications requises:

- Votre site devra comprendre au moins trois pages différentes : une page principale et deux pages secondaires.
- La page principale devra présenter un lien vers chacune des pages secondaires
- Les pages secondaires devront présenter des liens pour retourner à la page principale.
- Utilisez au moins une image sur chaque page (vous pouvez utiliser des images trouvées sur le Web, puisque ce site ne sortira pas d'ici)
- Utilisez au moins un lien vers un autre site Web que le vôtre
- Utilisez un lien vers une adresse de courriel
- Jouez avec les couleurs de fond et de police, et expérimentez avec différentes polices et tailles de caractères.
- Utilisez quelques balises vues dans ce cours pour rendre la présentation agréable.
- N'oubliez pas les balises organisationnelles `html`, `head` et `body`!

Ne vous gênez pas pour essayer de nouvelles notions!