

# AP1 : prise en main d'un site WEB existant, ajout de contenu et développement personnel

**B1.3 Développer la présence en ligne**

**B1.6 Organiser son développement personnel**

**Objet du document :** Apprentissage des éléments principaux de HTML 5 et CSS 3 afin de pouvoir réaliser un site web

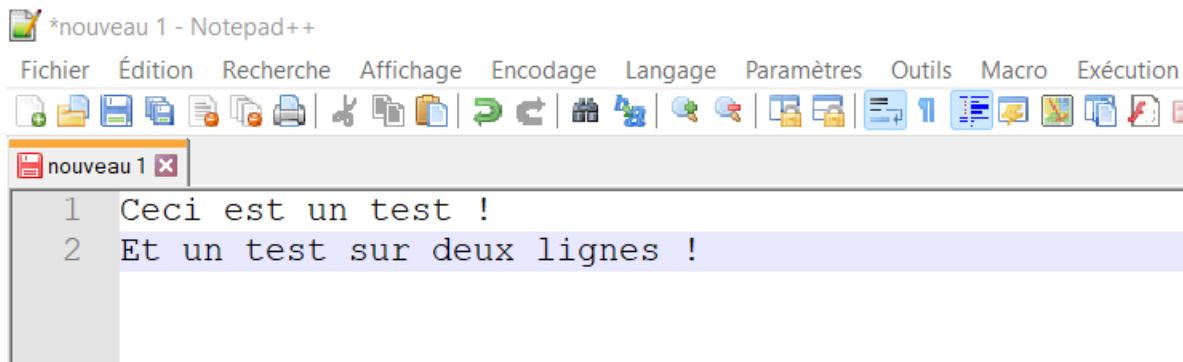
**Prérequis :** Editeur de texte Notepad++ ou VS code Navigateur Firefox et Chrome

## Chapitre 2 : Créez votre première page web en HTML

Tout d'abord, dans placez-vous dans votre dossier partagé.

Créer un dossier <AP1.1> dans lequel vous créez un dossier <Exercice>

Ouvrez Notepad++ et saisissez un texte ( le texte en lui-même n'a pas d'importance pour le moment)



Utilisez <Fichier> <Enregistrer sous>

Une boîte de dialogue vous demande où enregistrer le fichier et sous quel nom.

Enregistrez-le où vous voulez. Donnez au fichier le nom que vous voulez, en terminant par .html ; par exemple test.html , comme indiqué à la figure suivante.

**Attention :** un nom de fichier doit être écrit en minuscule, sans espace ni caractères spéciaux !

Enregistrer le document dans le répertoire exercice préalablement créé !

L'icône qui représente le fichier dépend de votre navigateur web par défaut, en général. Vous pouvez y voir une icône de Firefox, de Chrome... ou un aperçu, comme ici. N'y prêtez pas attention. Faites simplement un double-clic sur ce fichier et... votre navigateur s'ouvre et affiche le texte que vous avez écrit.

*Cela ne marche pas bien, on dirait ! Tout le texte s'affiche sur la même ligne alors qu'on avait écrit deux lignes de texte différentes !?*

En effet, bien vu !

Le texte s'affiche sur la même ligne alors qu'on avait demandé à l'écrire sur deux lignes différentes. Que se passe-t-il ?

En fait, pour créer une page web, il ne suffit pas de taper simplement du texte comme on vient de le faire. En plus de ce texte, il faut aussi écrire ce qu'on appelle des **balises**, qui vont donner des instructions à l'ordinateur comme « aller à la ligne », « afficher une image », etc.

## Les balises et leurs attributs

Bon, tout cela était trop facile. Évidemment, il a fallu que ces satanés informaticiens s'en mêlent et compliquent les choses. Il ne suffit pas d'écrire « simplement » du texte dans l'éditeur, il faut aussi donner des instructions à l'ordinateur. En HTML, on utilise pour cela des **balises**.

## Les balises

Les pages HTML sont remplies de ce qu'on appelle des **balises**. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher.

Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles < et > , comme ceci : <balise> .

À quoi est-ce qu'elles servent ? Elles indiquent la nature du texte qu'elles encadrent. Elles veulent dire par exemple : « Ceci est le titre de la page », « Ceci est une image », « Ceci est un paragraphe de texte », etc.

On distingue deux types de balises : les balises en paires et les balises orphelines.

## Les balises en paires

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :

```
<titre>Ceci est un titre</titre>
```

On distingue une balise ouvrante ( <titre> ) et une balise fermante ( </titre> ) qui indique que le titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est *pas* entre ces deux balises... n'est pas un titre.

```
Ceci n'est pas un titre <titre>Ceci est un titre</titre> Ceci n'est pas un titre
```

## Les balises orphelines

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

Une balise orpheline s'écrit comme ceci : `<image />`

Notez que le / de fin n'est pas obligatoire. On pourrait écrire seulement `<image>` . Néanmoins, pour ne pas les confondre avec le premier type de balise, les webmasters recommandent d'ajouter ce / (*slash*) à la fin des balises orphelines. Vous me verrez donc mettre un / aux balises orphelines et je vous recommande de faire de même, c'est une bonne pratique.

## Les attributs

Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires. L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```

À quoi cela sert-il ? Prenons la balise `<image />` que nous venons de voir. Seule, elle ne sert pas à grand-chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher :

```
<image nom="photo.jpg" />
```

L'ordinateur comprend alors qu'il doit afficher l'image contenue dans le fichier photo.jpg .

Dans le cas d'une balise fonctionnant « par paire », on ne met les attributs que dans la balise ouvrante et pas dans la balise fermante. Par exemple, ce code indique que la citation est de Neil Armstrong et qu'elle date du 21 juillet 1969 :

```
<citation auteur="Neil Armstrong" date="21/07/1969">
C'est un petit pas pour l'homme, mais un bond de géant pour l'humanité.
</citation>
```

Toutes les balises que nous venons de voir sont fictives. Les vraies balises ont des noms en anglais (eh oui !) ; nous allons les découvrir dans la suite de ce cours.

## Structure de base d'une page HTML5

Reprenons notre éditeur de texte (dans mon cas, Sublime Text). Je vous invite à écrire ou à copier-coller le code source ci-dessous dans votre éditeur de texte. Ce code correspond à la base d'une page web en HTML5 :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>
  <body>
  </body>
```

```
</html>
```

Attention : J'ai mis des espaces au début de certaines lignes pour « décaler » les balises. Ceci est obligatoire et car cela rend le code source plus lisible. On appelle cela l'**indentation**. Dans votre éditeur, il suffit d'appuyer sur la touche Tab pour avoir le même résultat.

Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis. Par exemple, la balise `<html>` est la première que l'on ouvre et c'est aussi la dernière que l'on ferme (tout à la fin du code, avec `</html>` ). *Les balises doivent être fermées dans le sens inverse de leur ouverture*. Un exemple :

- `<html><body></body></html>` : **correct**. Une balise qui est ouverte à l'intérieur d'une autre doit aussi être fermée à l'intérieur.
- `<html><body></html></body>` : **incorrect**, les balises s'entremêlent.

## Le doctype

```
<!DOCTYPE html>
```

La toute première ligne s'appelle le **doctype**. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML.

Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation). Vous pouvez considérer que c'est un peu l'exception qui confirme la règle.

## La balise `</html>`

```
<html>  
</html>
```

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante `</html>` se trouve tout à la fin du code !

## L'en-tête `<head>` et le corps `<body>`

Une page web est constituée de deux parties :

- L'en-tête `<head>` : cette section donne quelques informations générales sur la page, comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !
- Le corps `<body>` : c'est là que se trouve la partie principale de la page. Tout ce que nous écrivons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrivons la majeure partie de notre code.

Pour le moment, le corps est vide (nous y reviendrons plus loin). Intéressons-nous par contre aux deux balises contenues dans l'en-tête...

## L'encodage ( `charset` )

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

Cette balise indique l'encodage utilisé dans votre fichier `.html` .

Sans rentrer dans les détails, car cela pourrait vite devenir compliqué, l'encodage indique la façon dont le fichier est enregistré. C'est lui qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

Il y a plusieurs techniques d'encodage, portant des noms bizarres, et utilisées en fonction des langues : ISO-8859-1, OEM 775, Windows-1253... Une seule cependant devrait être utilisée aujourd'hui autant que possible : UTF-8. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues de notre planète ! C'est pour cela que j'ai indiqué `utf-8` dans cette balise.

Il ne suffit pas de *dire* que votre fichier est en UTF-8. Il faut aussi que votre fichier soit bien *enregistré* en UTF-8. C'est heureusement le cas désormais par défaut dans la plupart des éditeurs de texte.

Si les accents s'affichent mal par la suite, c'est qu'il y a un problème avec l'encodage. Vérifiez que la balise meta indique bien UTF-8, et que votre fichier est enregistré en UTF-8 (sous Sublime Text, allez dans le menu File > Save with Encoding > UTF-8 pour vous assurer que votre fichier est enregistré en UTF-8.).

## Le titre principal de la page

`<title>`

C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient.

Il est conseillé de garder le titre assez court (moins de 100 caractères, en général).

Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur). Enregistrez votre page web et ouvrez-la dans votre navigateur. Vous verrez que le titre s'affiche dans l'onglet, comme sur la figure suivante.

Il faut savoir que le titre apparaît aussi dans les résultats de recherche, comme sur Google (figure suivante).

Autant vous dire que bien choisir son titre est important !

## Les commentaires

Nous avons appris à créer notre première *vraie* page HTML dans ce chapitre. Avant de terminer, j'aimerais vous présenter le principe des commentaires.

Un **commentaire** en HTML est un texte qui sert simplement de mémo. Il n'est pas affiché, il n'est pas lu par l'ordinateur, cela ne change rien à l'affichage de la page.

Un **commentaire** sert à *vous* et aux personnes qui liront le code source de votre page. Vous pouvez utiliser les commentaires pour laisser des indications sur le fonctionnement de votre page.

Quel intérêt ? Cela vous permettra de vous rappeler comment fonctionne votre page si vous revenez sur votre code source après un long moment d'absence. Ne rigolez pas, cela arrive à tous les webmasters.

## Insérer un commentaire

Un commentaire est une balise HTML avec une forme bien spéciale :

```
<!-- Ceci est un commentaire -->
```

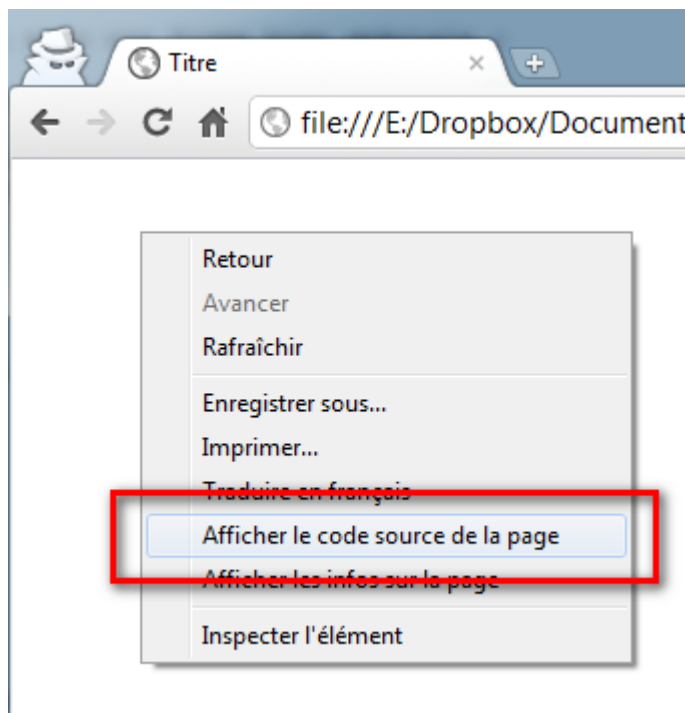
Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page, mais vous pouvez vous en servir pour vous aider à vous repérer dans votre code source (surtout s'il est long).

```
<!DOCTYPE html>
<html>
  <head>
    <!-- En-tête de la page -->
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>

  <body>
    <!-- Corps de la page -->
  </body>
</html>
```

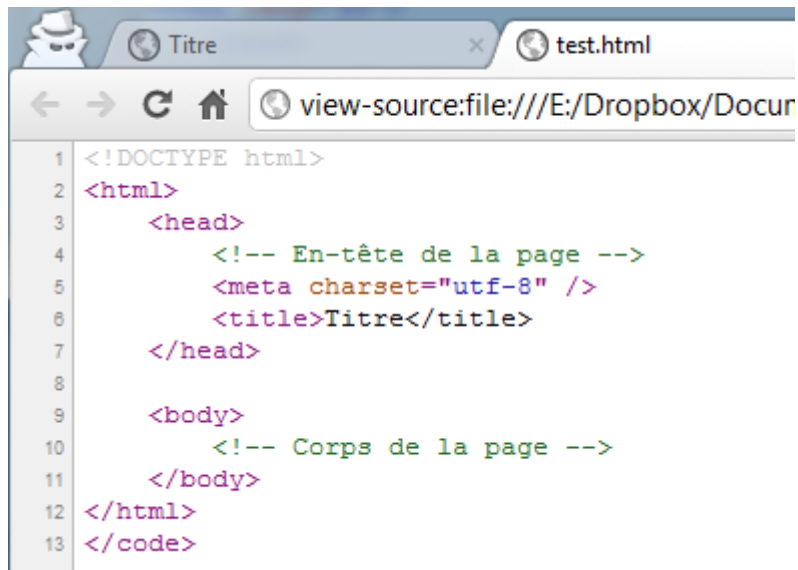
## Tout le monde peut voir vos commentaires... et tout votre code HTML !

Terminons par une remarque importante : *tout le monde peut voir le code HTML de votre page* une fois celle-ci mise en ligne sur le Web. Il suffit de faire un clic droit sur la page et de sélectionner « Afficher le code source de la page » (l'intitulé peut changer selon votre navigateur), comme le montre la figure suivante.



Menu Afficher le code source

Le code source s'affiche alors (figure suivante).



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- En-tête de la page -->
5     <meta charset="utf-8" />
6     <title>Titre</title>
7   </head>
8
9   <body>
10    <!-- Corps de la page -->
11  </body>
12 </html>
13 </code>
```

Affichage du code source

Vous pouvez tester cette manipulation sur n'importe quel site web, cela marche ! Garanti à 100 %. Cela s'explique assez facilement : le navigateur *doit* obtenir le code HTML pour savoir ce qu'il faut afficher. Le code HTML de tous les sites est donc public.

La morale de l'histoire ? Tout le monde pourra voir votre code HTML et vous ne pouvez pas l'empêcher. Par conséquent, ne mettez pas d'informations sensibles comme des mots de passe dans les commentaires...

EN RÉSUMÉ : NOTEZ ICI LES POINTS IMPORTANTS DU CHAPITRE