

# 1 Introduction au World Wide Web

## 1.1 L'Internet et le Web, quelle différence ?

**Internet** est un réseau informatique mondial.

Le **World Wide Web**, que l'on appelle le plus souvent **Web**, est un des services disponibles sur Internet.

D'autres services disponibles sur Internet sont par exemple la messagerie électronique (**email**) ou le transfert de fichier (**FTP**)

### a) Le Web ou World Wide Web, à quoi ça sert ?

Le Web ("la toile" en français) est un système permettant de visualiser et d'échanger des informations à distance.

Les informations sont présentées sur des pages Web et reliées les unes aux autres par des liens hypertextes.

### b) Qu'est-ce qu'une page Web ? Qu'est-ce qu'un site Web ?

Une page Web est généralement un fichier texte écrit dans un langage informatique baptisé HTML.

Ce fichier peut contenir des images ou d'autres contenus multimédias (audio, vidéo, application).

Un site Web est un ensemble de pages Web reliées les unes aux autres par des liens hypertextes et accessible en ligne à une adresse.

### c) Le langage HTML

**HTML** est l'acronyme d'**H**ypertext **M**arkup **L**anguage qui peut se traduire ainsi : langage hypertexte à balise.

Le HTML est un langage informatique utilisant des **textes** et des **balises**. On peut utiliser un simple éditeur de texte pour créer un fichier HTML.

Pour visionner un fichier en HTML on utilise un navigateur. Un navigateur est un logiciel qui va lire le code HTML et l'interpréter. Chaque balise donne au navigateur des informations sur comment afficher tel ou tel élément.

### d) À propos des balises

Une balise (markup en anglais) est un mot clé qui va donner des indications au navigateur sur ce qu'il doit afficher.

Une balise HTML commence par le caractère « < » et termine par le caractère « > ».

Les caractères qui ne sont pas compris entre les signes « < » et « > » sont donc considérés comme du texte et seront affichés tel quel par le navigateur.

- Il existe des balises ouvrantes et des balises fermantes qui correspondent au début et la fin d'une instruction.
- Les balises de fermeture sont identiques à celles d'ouverture à l'exception de l'ajout d'un caractère « / » (slash) pour signaler la fermeture. Ce caractère se place juste après le signe « > » (inférieur)

### e) Structure d'un fichier HTML

Un fichier HTML (un document HTML) commence par l'ouverture d'une balise <html> et se termine par la balise fermante </html>

Tous les fichiers HTML sont composés d'une entête (balise "head") et d'un corps (balise "body"). L'entête contient des informations sur la page.

Le corps contient le contenu de la page.

Listing 1 – Exemple

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

### f) Principe du Web

La Structure simplifiée d'une adresse, d'une URL est : **protocole** **://** **serveur** **/ressource**

Par exemple : [http://www.univ-nancy2.fr/formations/calendrier\\_licence.html](http://www.univ-nancy2.fr/formations/calendrier_licence.html)

- protocole : **http** pour Hyper Text Transfer (il y en a d'autre comme ftp)
- nom du serveur : nom de domaine de l'ordinateur hébergeant la ressource demandée.
- Emplacement de la ressource sur le serveur : En général on utilise le répertoire et le nom du fichier demandé.

### g) HTTP et HTML

- HTTP=Hypertext Transfer Protocol :
  - Protocole utilisé pour transférer des fichiers (html, mais pas forcément) entre un serveur http et un navigateur
- Que se passe-t-il lorsque je consulte une page web ?
  - Grâce à l'url de la page, mon navigateur consulte un DNS (qui peut lui-même demander à un autre DNS) le serveur qui l'héberge,
  - Mon navigateur demande au serveur (grâce à HTTP) la page
  - Le serveur la renvoie (toujours grâce à HTTP)
  - Mon navigateur affiche la page (grâce à HTML)
- HTML=Hypertext Markup Language :
  - Langage de balisage hypertexte
  - Langage informatique créé pour écrire des pages web

## 2 HTML

- <https://www.pierre-giraud.com/html-css/cours-complet/cours-html-css-presentation.php>

### 2.1 Préparation

**HTML** est l'abréviation de **HyperText Markup Language**, soit en français « langage de balisage hypertexte ». Ce langage a été créé en 1991 et a pour fonction de structurer et de donner du sens à du contenu.

Grâce au HTML, on va par exemple pouvoir indiquer au navigateur que tel texte doit être considéré comme un simple paragraphe ou que tel autre est un titre.

Pour écrire les fichiers html on a deux solutions.

1. Écrire directement le code avec un éditeur de texte (Attention à ne pas confondre avec un traitement de texte comme openoffice, word ...).

→ Lancer *notepad++* ou *pyzo*.

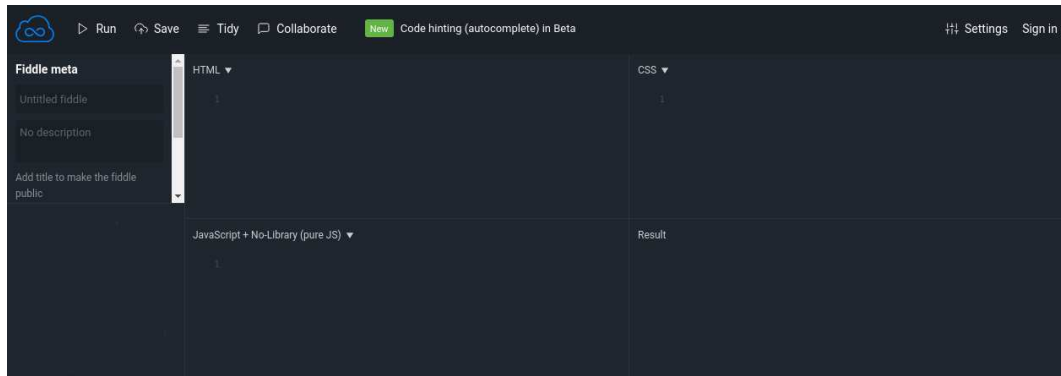
→ Écrire le code suivant :


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

→ Enregistrer le fichier dans votre session, ouvrez un navigateur de votre choix (Firefox, chrome ...), et ouvrez le fichier.

2. Utiliser un éditeur en ligne. Nous utiliserons le site [jsfiddle.net](https://jsfiddle.net).

→ Lancer le navigateur de votre choix (Firefox ou chrome), taper <http://jsfiddle.net/> dans la barre d'adresse. Vous devriez voir apparaître ceci :



Si vous avez déjà du texte alors cliquez sur  → Empty

→ Recopier le texte écrit entre les balises `<body>` dans la section **HTML**

```
1 <!-- Corps de la page -->
2 <p>Hello</p>
```

→ Pour visualiser cliquez sur l'icone **Run**.

À noter :

- La ligne numéro 6 sert pour préciser l'encodage utf8 (ainsi les accents « français » sont bien pris en compte). Pour HTML5 l'encodage par défaut des caractères est UTF-8 .
- Lorsque vous réaliserez une nouvelle page html, vous ferez un « copier-coller » de cet en-tête.
- Tous les documents html que vous enregistrerez devront avoir l'extension : **.html**

## 2.2 Organiser une page :

### a) 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 >`. Elles indiquent la nature du texte qu'elles encadrent (un titre, une image ...).

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

- Les balises en paires s'ouvrent, contiennent du texte, et se ferment plus loin.  
Voici à quoi elles ressemblent : `< title > Ceci est un titre < /title >`.
- Les balises orphelines servent souvent à insérer un élément à un endroit précis, par exemple une image.  
Voici à quoi elles ressemblent `<image />`.

### b) Base d'une page html

- `<!DOCTYPE html >` signifie que la page est écrite en **HTML5**
- `<html> ... </html>` englobe tout le contenu de la page
- `<head> ... </head>` : (header) fournit des informations générales (métadonnées) sur le document, incluant son titre et des liens vers des scripts et feuilles de style.

```
3 <head>
4 <!-- En-tête de la page -->
5 <title>Titre</title>
6 <meta charset="utf-8" />
7 </head>
```

Cette balise encadre des informations générales sur la page comme le titre, l'encodage (gestion des caractères spéciaux comme les accents en français).

Enlever la ligne 6 pour voir ce que ça donne.

- `<body> ... </body>` encadrent le contenu du site web.

## 2.3 Organiser son texte

### a) Les paragraphes

- `<p> ... </p>` pour encadrer un paragraphe
- `<br />` pour sauter une ligne

Écrire quelques lignes utilisant ces balises avec *notepad++* et tester le résultat.

### b) Les titres

En HTML, on peut utiliser six niveaux de titres différents. On a donc six balises de titres différentes :

- `<h1> ... </h1>` : signifie « titre très important ».
- `<h2> ... </h2>` : signifie « titre important ».
- `<h3> ... </h3>` : pareil, c'est un titre un peu moins important.
- `<h4> ... </h4>` : titre encore moins important.
- `<h5> ... </h5>` : titre pas important.
- `<h6> ... </h6>` : titre pas important du tout.

Écrire quelques lignes utilisant ces balises avec *notepad++* et tester le résultat.

### c) Les listes

Les listes nous permettent souvent de mieux structurer notre texte et d'ordonner nos informations.

Il existe plusieurs types de listes dont :

- les listes à puces (**unordonné** en anglais) : `<ul> ... </ul>`.
- les listes numérotées (**ordonné** en anglais) : `<ol> ... </ol>`. Dans les deux cas, les différentes lignes sont encadrées par les balises `<li> ... </li>`.

Exemples :

```
<ul>
  <li> Fraises </li>
  <li> Framboises </li>
  <li> Cerises </li>
</ul>
```

```
<ol>
  <li> Fraises </li>
  <li> Framboises </li>
  <li> Cerises </li>
</ol>
```

Tester ces lignes de code avec *notepad++*.

### d) Les tableaux

Pour créer un tableau simple, nous allons avoir besoin de trois éléments HTML :

- La balise `<table>` ( « tableau » en français) va définir le tableau ;
- La balise `<tr>`, pour « table row » va nous servir à introduire une nouvelle ligne ;
- La balise `<td>`, pour « table data » ou « donnée de tableau » en français va nous permettre d'ajouter des cellules.
- La balise `<th>` pour « table header » est utilisé pour définir les titres d'un tableau html.

Exemples :

```
<table>
  <tr><th>Nom</th><th>prénom</th><th>âge</th></tr>
  <tr><td>Onette</td><td>Camille</td><td>15</td></tr>
  <tr><td>Vanille</td><td>Douglas</td><td>16</td></tr>
  <tr><td>Bijoba</td><td>Joe</td><td>16</td></tr>
</table>
```

Tester ces lignes de code avec *notepad++*.

Pour obtenir les lignes il faut utiliser du code **CSS** que nous verrons donc plus tard.

- **colspan** Spécifie le nombre de colonnes sur lequel s'étendra la cellule. Cela permet de scinder les colonnes.
- **rowspan** spécifie le nombre de lignes sur lequel s'étendra la cellule. Cela permet de scinder les lignes.

Exemple :

```
<table cellpadding="3px" cellspacing="0px" rules="all" style="border:solid 1px black;
border-collapse:collapse; text-align:center;">
  <tr>  <th>Titre 1</th> <th>Titre 2</th> </tr>
  <tr>  <td>Valeur1</td> <td>Valeur2</td> </tr>
  <tr>
    <td colspan="2" style="background-color:lightyellow;">Cellule avec colspan</td>
  </tr>
  <tr>
    <td rowspan="2" style="background-color:lightsteelblue;">Cellule avec rowspan</td>
    <td>Valeur 3</td>
  </tr>
  <tr>
    <td>Valeur 4</td>  </tr>
</table>
```

Tester ces lignes de code avec *notepad++*.

#### e) Insérer des images

L'insertion d'images va se faire au moyen de la balise orpheline `<img .../>`

- L'attribut **src** (pour source) va prendre comme valeur l'adresse de l'image (adresse relative ou absolue)
- L'attribut **alt** (pour alternative) va contenir un texte alternatif décrivant l'image. Ce texte va être affiché si l'image ne peut pas l'être pour une raison ou pour une autre, et est également très utile pour les non-voyants.
- L'attribut **width** permet de fixer une taille précise.

Exemple :

Listing 2 – Exemple

```
<img alt='Pablo Neruda' src='https://upload.wikimedia.org/wikipedia/commons/8/86/Pablo_Neruda_1963.jpg'
width=200px />
```

Tester ces lignes de code avec *notepad++*.

#### f) Les liens

**Html** (Hyper Text Markup Language) est un langage hypertexte qui vous permet en cliquant sur un mot, généralement souligné (ou une image) de vous transporter ;

##### → Les liens externes.

Les liens externes sont des liens ramenant vers des pages d'autres sites.

Listing 3 – Exemple

```
<h1>Les liens</h1>
<p>Cliquez sur <a href="http://wikipedia.org">ce lien </a>pour aller sur Wikipédia.</p>
```

Tester ces lignes de code avec *notepad++*.

##### → Les liens internes en HTML

Les liens internes vont être des liens ramenant vers d'autres pages au sein d'un même site.

Dans ce cas, nous allons renseigner une **valeur relative** pour notre attribut *href*.

Créez les fichiers suivants :

Listing 4 – test-lien1.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Les liens en HTML</title>
```

```

5  <meta charset="utf-8" />
6  </head>
7  <body>
8  <h1>Les liens</h1>
9  <p>Page de départ</p>
10 <p>Cliquez pour accéder à <a href="test-lien2.html">la deuxième page</a>.</p>
11 <body>
12 </html>

```

Listing 5 – test-lien2.html

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Les liens en HTML</title>
5    <meta charset="utf-8" />
6  </head>
7  <body>
8  <p>Deuxièmes page</p>
9  <body>
10 </html>

```

Ouvrez un navigateur puis ouvrir la page test-lien1.html

## 2.4 Un peu de mise en forme

Tout document *Html* contient en majorité du texte. Voici comment l'agrémenter par quelques balises élémentaires.

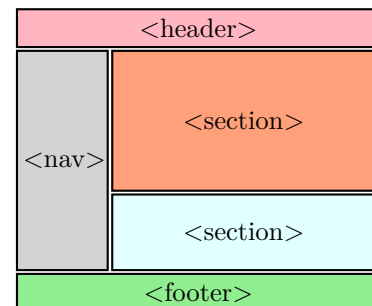
- `<!-- ...-->` : Pour écrire un commentaire. Le texte n'est pas affiché.
- `<b> ... </b>` : pour mettre le texte en Gras (Bold en anglais).
- `<strong> ... </strong>` : indique que le texte a une importance particulière (*bold*).
- `<i> ... </i>` : pour mettre le texte en *Italique*.
- `<em> ... </em>` : est utilisé afin de marquer un texte sur lequel on veut insister (*Italic*).
- `<mark> ... </mark>` : représente un texte marqué à cause de sa pertinence dans le contexte.  
Il peut par exemple être utilisé afin d'indiquer les correspondances d'un mot-clé recherché au sein d'un document.
- `<center> ... </center>` : pour centrer du texte.

## 2.5 Structurer sa page

En général, une page web est constituée d'un en-tête (tout en haut), de menus de navigation (en haut ou sur les côtés), de différentes sections au centre... et d'un pied de page (tout en bas).

HTML5 a introduit des balises permettant de structurer ses pages.

- `<header> ... </header>` : l'en-tête pour le logo, la bannière de notre site ...
- `<footer> ... </footer>` : le pied de page.
- `<nav> ... </nav>` : principaux liens de navigation ou le menu.
- `<section> ... </section>` : permet de créer des sections de page



## 2.6 TP

Je désire avoir un site de cuisine avec trois recettes de cuisine (une entrée, un plat et un dessert) que vous aimez bien. Chaque recette doit être écrite dans un fichier **html** et doit contenir la liste des ingrédients, une image du plat et le texte de la recette.

Un fichier principal **Recettes.html** doit contenir la liste des recettes et des liens pointant vers ces recettes.

- Un page **Recettes.html** avec des liens vers les recettes

- Une page *entree.html* avec la recettes de l'entrée.
- Une page *plat.html* avec la recettes d'un plat.
- Une page *dessert.html* avec la recette d'un dessert.

## 3 CSS

### 3.1 Préparation

**CSS** est le diminutif de **Cascading StyleSheets**, ou feuilles de styles en cascade.

Le CSS a été créé en 1996 et a pour rôle de mettre en forme du contenu en lui appliquant ce qu'on appelle des styles.

Le CSS va nous permettre par exemple de définir la taille, la couleur ou l'alignement d'un texte.

Nous allons donc utiliser le CSS sur notre code HTML, afin d'enjoliver le résultat visuel final.

Pour écrire le code css on a deux solutions.

→ Si on utilise [jsfiddle.net](https://jsfiddle.net) on écrit directement le code dans la partie **CSS**

Listing 6 – HTML

```
<p>Voici un exemple</p>
```

Listing 7 – CSS

```
p {  
    color: red;  
}
```

→ Si on écrit les page dans un fichier **HTML** on peut écrire le code **css** dans la même page mais il est préférable d'écrire le code **css** dans un deuxième fichier avec l'extension **css**.

Écrire le fichier *test-css.html* avec le code suivant :

Listing 8 – test-css.html

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4   <title>Exemple css</title>  
5   <meta charset="utf-8" />  
6   <link rel="stylesheet" type="text/css" href="monstyle.css" />  
7 </head>  
8 <body>  
9   <p>Hello</p>  
10 </body>  
11 </html>
```

Listing 9 – *monstyle.css*

```
1 p {  
2   color: red;  
3 }
```

Ouvrez le fichier *test-css.html* avec un navigateur.

### 3.2 Le principe

#### a) Par sélecteur de balise

Tester ces lignes de code ci-dessous sur <https://jsfiddle.net>.

```
<h1>titre 1</h1>
<p>ligne 1</p>
<p>ligne 2</p>
<p>ligne 3</p>
<h1>titre 2</h1>
<p>ligne 4</p>
<p>ligne 5</p>
```

Regardez le résultat puis dans la partie **css** écrire :

```
p {
    color : red;
}
```

Regardez le résultat.

```
p h1 {
    color : red;
}
```

Regardez le résultat.

### b) Par sélecteur d'identités (id)

Le sélecteur **#id** nous permet de cibler un élément en particulier plutôt qu'un type d'élément.

Une **id** doit être **unique**.

Pour cibler un élément possédant un attribut **id**, en CSS, il faudra préciser la valeur de l'attribut précédée d'un dièse (#).

```
<h1>titre 1</h1>
<p id='rouge'>ligne 1</p>
<p id='bleu'>ligne 2</p>
<p>ligne 3</p>
<h1>titre 2</h1>
<p>ligne 4</p>
<p>ligne 5</p>
```

```
#rouge {
    color : red;
}
#bleu {
    color : blue;
}
```

Regardez le résultat.

### c) Par sélecteur de Classes.

Le sélecteur **.class** nous permet de cibler un ensemble d'éléments possédant les mêmes propriétés.

Pour cibler un élément possédant un attribut **class**, en revanche, il faudra en CSS préciser la valeur de l'attribut précédée d'un point (.).

```
<h1>titre 1</h1>
<p class='rouge'>ligne 1</p>
<p class='bleu'>ligne 2</p>
<p>ligne 3</p>
```



```
<h1 class='rouge'>titre 2</h1>
<p class='rouge'>ligne 4</p>
<p class='bleu'>ligne 5</p>
```

```
.rouge {
    color : red;
}
.bleu {
    color : blue;
}
```

Regardez le résultat puis modifier par :

```
.rouge p{
    color : red;
}
.bleu {
    color : blue;
}
```

Regardez le résultat.

Il y a de nombreuses propriétés en **CSS**. Voici plusieurs liens intéressants.

- Les propriétés CSS les plus utilisées sur [cssdebutant.com](https://cssdebutant.com)
- Mémento des propriétés CSS sur [openclassrooms](https://openclassrooms.com)
- Toutes les propriétés CSS sur [css-faciles.com](https://css-faciles.com)

### 3.3 Livre recette

Modifier votre livre de recettes à l'aide de règle **CSS** de votre choix.

## 4 Résumé html

Le **World Wide Web**, que l'on appelle le plus souvent **Web**, est un des services disponibles sur Internet. D'autres services disponibles sur Internet sont par exemple la messagerie électronique (**email**) ou le transfert de fichier (**FTP**)

Listing 10 – Structure générale

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

- `<p> ... </p>` pour encadrer un paragraphe
- `<br />` pour sauter une ligne
- `<h1> ... </h1>` : signifie « titre très important ».
- `<h2> ... </h2>` : signifie « titre important ».
- `<h3> ... </h3>` : pareil, c'est un titre un peu moins important. etc...

Listing 11 – Listes à puces (*unordonned*)

```
<ul>
  <li> Fraises </li>
  <li> Framboises </li>
  <li> Cerises </li>
</ul>
```

Listing 12 – Listes numérotés (*ordonned*)

```
<ol>
  <li> Fraises </li>
  <li> Framboises </li>
  <li> Cerises </li>
</ol>
```

Listing 13 – Un tableau

```
<table>
  <tr><th>Nom</th><th>prénom</th><th>âge</th></tr>
  <tr><td>Onette</td><td>Camille</td><td>15</td></tr>
  <tr><td>Vanille</td><td>Douglas</td><td>16</td></tr>
  <tr><td>Bijoba</td><td>Joe</td><td>16</td></tr>
</table>
```

Listing 14 – Une image

```
<img alt='Pablo Neruda' src='https://upload.wikimedia.org/wikipedia/commons/8/86/Pablo_Neruda_1963.jpg'
      width=200px />
<img alt='Texte alternatif' src='Monimage.jpg' width=200px />
```

Listing 15 – Un lien

```
<h1>Les liens</h1>
<p>Cliquez sur <a href="http://wikipedia.org">ce lien externe</a>pour aller sur Wikipédia.</p>
<p>Cliquez sur <a href="page2.html">ce lien interne</a>pour aller sur une deuxième page html.</p>
```