

1 HTML

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

1.1 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*.

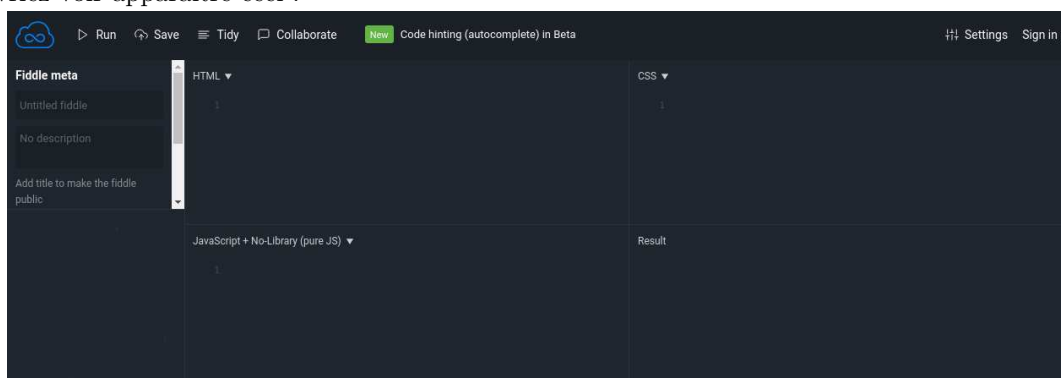
→ Écrire le code suivant :


```
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.

→ 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**

1.2 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.

1.3 Organiser son texte

a) Les paragraphes

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

Écrire quelques lignes utilisant ces balises sur jsfiddle.net

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 sur jsfiddle.net

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) :
- les listes numérotées (**ordonné** en anglais) : Dans les deux cas, les différentes lignes sont encadrées par les balises

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 sur jsfiddle.net.

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 sur jsfiddle.net

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 sur jsfiddle.net

e) Insérer des images

L'insertion d'images va se faire au moyen de la balise orpheline ``

- 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 :

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

Tester ces lignes de code sur jsfiddle.net.

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.

```
<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 sur jsfiddle.net.

→ 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 1 – 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 2 – 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

1.4 Un peu de mise en forme

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

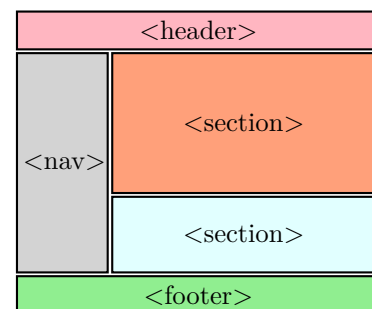
- `<!-- ...-->` : Pour écrire un commentaire. Le texte n'est pas affiché.
- ` ... ` : pour mettre le texte en Gras (Bold en anglais).
- ` ... ` : indique que le texte a une importance particulière (*bold*).
- `<i> ... </i>` : pour mettre le texte en *Italique*.
- ` ... ` : est utilisé afin de marquer 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.

1.5 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



1.6 TP

Je désire avoir un livre de cuisine avec quatre recettes de cuisine (une entrée, un plats et deux desserts) que vous aimez bien.

Chaque recettes doit contenir la liste des ingrédients, une image du plat et le texte de la recette.

Le fichier *Recettes.html* doit contenir :

- Première partie : Les entrée
- Deuxième partie : Les plats
- Troisième partie : Les desserts.

Me faire vérifier puis après transformer ce site en 4 pages.

- Un page *Recettes.html* avec des liens vers les recettes
- Une page *entree.html* avec les recettes des entrées.
- Une page *plat.html* avec les recettes des plats.
- Une page *dessert.html* avec les recettes des desserts.

2 CSS

2.1 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 on écrit directement le code dans la partie **CSS**

Listing 3 – HTML

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

Listing 4 – 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 5 – test-css.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Exemple css</title>
```

```
5 <meta charset="utf-8" />
6 <link rel="stylesheet" href="monstyle.css" />
7 </head>
8 <body>
9   <p>Hello</p>
10 </body>
11 </html>
```

Listing 6 – *monstyle.css*

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

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

2.2 Le principe

a) Par sélecteur de balise

Tester Les 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;  
}  
#blue {  
    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
- Mémento des propriétés CSS sur [openclassrooms](https://openclassrooms.com)
- Toutes les propriétés CSS sur css-faciles.com

2.3 Livre recette

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

3 Exposés

Voici différents thèmes.

- Faire des groupes de 3.
- Choisir un des thèmes ci-dessous
-
- Faire des recherche sur le thème choisit puis faire un site internet pour expliquer.