

# Interaction Homme - Machine sur le WEB

---

## Partie 1 - Le HTML

---

### Introduction

Le `World Wide Web`, plus communément appelé `web` a été développé au CERN (Conseil Européen pour la Recherche Nucléaire) par le Britannique Sir Timothy John Berners-Lee et le Belge Robert Cailliau au début des années 90.

À cette époque les principaux centres de recherche mondiaux étaient déjà connectés les uns aux autres, mais pour faciliter les échanges d'information Tim Berners-Lee met au point le `système hypertexte`.

Le `système hypertexte` permet, à partir d'un document, de consulter d'autres documents en cliquant sur des mots clés. Ces mots "cliquables" sont appelés `hyperliens` et sont souvent soulignés et en bleu. Ces `hyperliens` sont plutôt connus aujourd'hui sous le simple terme de `liens`.

La première page web créée est toujours consultable à l'adresse suivante :

<http://info.cern.ch/hypertext/WWW/TheProject.html>

Tim Berners-Lee développe le premier navigateur web (logiciel permettant de lire des pages contenant des hypertextes), il l'appelle simplement `WorldWideWeb`. Il faudra attendre 1993 et l'arrivée du navigateur web `NCSA Mosaic` pour que le web commence à devenir populaire en dehors du petit monde de la recherche.

Techniquement le web se base sur trois choses : le protocole HTTP (HyperText Transfert Protocol), les URL (Uniform Resource Locator) et le langage de description HTML (HyperText Markup Language). Nous aurons, très prochainement l'occasion de revenir sur ces trois éléments.

Une chose très importante à bien avoir à l'esprit : beaucoup de personnes confondent "web" et "internet". Même si le "web" "s'appuie" sur internet, les deux choses n'ont rien à voir puisque "internet" est un "réseau de réseau" s'appuyant sur le protocole IP alors que, comme nous venons de le voir, le web est la combinaison de trois technologies : HTTP, URL et HTML. D'ailleurs on trouve autre chose que le "web" sur internet, par exemple, les emails avec le protocole SMTP (Simple Mail Transfert Protocol) et les transferts de fichiers avec le protocole FTP (File Transfert Protocol).

Plus d'informations : <https://home.cern/fr/science/computing/birth-web/short-history-web>

Pour créer des pages web, nous utilisons (entre autre) les langage HTML (Hyper Text Markup Langage), CSS (Cascading Style Sheets) et d'autres langages qui vont permettre une interaction entre l'homme et la machine comme par exemple le javascript (côté client) ou le PHP (côté serveur)

Les navigateurs (Google Chrome, Firefox, Edge, Safari...) interprètent le HTML, le CSS et le javascript afin d'afficher les pages web.

Rendez-vous sur le site du lycée <https://edouard-branly-boulogne-sur-mer.enthdf.fr/> avec le navigateur de votre choix.

Faire un clic-droit sur la page d'accueil du site et affichez le code source de la page (raccourci CTRL + U).

Vous observez le code source de la page qui est interprété par le navigateur afin d'afficher le site web.

Dans ce code, on trouve différents langages : HTML, CSS, javascript...

Dans cette première partie, nous allons découvrir le langage HTML.

## Le langage HTML

Le langage `HTML` permet de structurer et de mettre en forme les éléments d'une page web. Il s'écrit avec des `balises` qui fonctionne quasiment toujours par deux : la balise d'ouverture et la balise de fermeture.

Grâce au HTML vous allez pouvoir, dans votre navigateur (Firefox, Chrome, Opera,...), afficher du texte, afficher des images, proposer des hyperliens (liens vers d'autres pages web), afficher des formulaires et même afficher des vidéos.

HTML n'est pas un langage de programmation (comme le Python par exemple), c'est un langage de description (pas de conditions, de boucles....).

Pour aborder le HTML, nous allons utiliser l'éditeur de code `notepad++` qui permet d'écrire du code HTML avec une coloration syntaxique (il met en couleur le code HTML pour en simplifier la lecture). Il existe d'autres éditeur de texte comme Atom, SublimText...

**Remarque :** Il existe aussi des éditeurs en ligne comme [Repl.it](https://repl.it), [Thimble](https://thimble.org), ou encore [Trinket](https://trinket.io).

## La structure de base d'une page html (activité 1)

Dans votre espace personnel, créez un dossier `web`

Dans `notepad++`, créez un fichier vide nommé `index.html` et l'enregistrer dans votre dossier `web` (n'oubliez pas de choisir comme langage le HTML dans les menus du logiciel)

Recopiez le code ci-dessous dans votre fichier `index.html`

```
<!DOCTYPE html>
<html lang="fr">
<html>
  <head>
    <meta charset="utf-8">
    <title>Une page web minimaliste</title>
  </head>
  <body>
    <h1>Le HTML</h1>
    <p>Le HTML est un <strong>langage de balisage </strong>,
      les balises vont par deux (en général)</p>
  </body>
</html>
```

Enregistrez vos modifications et visualisez le résultat obtenu dans le navigateur de votre choix.

Vous venez de créer votre première page web, ce code constitue le code source minimum.

Observons ce code, le fichier `index.html` commence par :

```
<!DOCTYPE html>
```

Cette ligne indique au navigateur que nous utiliserons la dernière version du HTML (le HTML5)

Puis vient l'indication de la langue utilisée :

```
<html lang="fr">
```

À la troisième ligne, on ouvre une balise `<html>` qui sera fermée avec `</html>` tout à la fin du code, c'est entre ces deux balises que sera codée la page web (on parle d'élément racine) :

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

L'élément racine est ensuite séparé en deux autres parties `head` et `body` délimitées par des balises HTML :

- L'en-tête `<head>` qui comporte des informations (codage des caractères, titre, liens vers des feuilles de style CSS, liens vers des scripts javascript, etc...) qui ne seront pas affichées par le navigateur.

- Le corps `<body>` qui contient tout le contenu de la page (textes, images, liens, etc...) tout votre code HTML décrivant votre page web devra se trouver entre ces 2 balises.

## Notion de balise HTML

---

Comme déjà évoqué ci-dessus, en HTML tout est une histoire de balise que l'on ouvre et que l'on ferme. Une balise ouvrante est de la forme `<nom_de_la_balise>`, les balises fermantes sont de la forme `</nom_de_la_balise>`.

Dans l'exemple précédent, trois balises sont utilisées :

- La balise `h1` qui permet de définir un titre de niveau 1.
- La balise `p` qui permet de définir un paragraphe.
- La balise `strong` qui permet de mettre en gras du texte.

En observant attentivement le code, vous devriez forcément remarquer que toute balise ouverte doit être refermée à un moment ou un autre. La balise ouvrante et la balise fermante peuvent être sur la même ligne ou pas, cela n'a aucune espèce d'importance, la seule question à se poser ici est : ai-je bien refermé toutes les balises que j'ai ouvertes ?

**Remarque :** pour plus de lisibilité dans votre code, il ne faut pas hésiter à alignement les balises ouvrantes et fermantes en utilisant les tabulations (cela ressemble à l'indentation utilisée avec Python)

### Modifier une page web (activité 2)

Modifiez le titre de la page web qui est actuellement "Une page web minimaliste" avec "Découverte du HTML"

Ajoutez un deuxième paragraphe avec comme titre "Exemples de balise HTML" et comme texte "La balise `p` permet de délimiter un paragraphe"

Il existe plusieurs niveaux de titre (de `h1` à `h6`), modifiez la page pour avoir des titres de paragraphes de niveau `h3`.

Enregistrez vos modifications, et observez le résultat dans le navigateur de votre choix.

**Remarque :** Il est important de bien comprendre la règle d'imbrication des balises dans notre exemple nous respectons bien cette règle avec la balise `<p>` et `<strong>`.

## D'autres balises

---

Il est important de comprendre que chaque balise a une signification qu'il faut bien respecter (on parle de la sémantique des balises).

## Une balise pour mettre du commentaire

Il peut être utile de mettre du commentaire dans le code d'une page web, ce qui sera mis en commentaire n'apparaîtra pas sur la page web (il ne sera pas interprété) mais uniquement dans le code source.

```
<!-- mettre du commentaire -->
```

## Une balise pour passer à la ligne

Vous devez aussi savoir qu'il existe des balises qui sont à la fois ouvrantes et fermantes, par exemple la balise permettant de sauter une ligne, la balise `<br/>`.

```
<p> Ceci est un paragraphe<br/> avec un passage à la ligne après le mot  
paragraphe</p>
```

## Une balise pour créer des listes à puces

Pour créer des listes à puces, il faut utiliser la balise `<ul>` afin de délimiter le début et la fin de la liste. Ensuite chaque item de la liste sera encadré par la balise `<li>`.

```
<ul>  
  <li>Pomme</li>  
  <li>Poire</li>  
  <li>Fraise</li>  
</ul>
```

## Une balise pour créer des liens hypertextes

La balise `a` permet de créer des liens hypertextes (qui sont par défaut soulignés en bleue), ces liens qui vous permettent de "surfer" entre les pages d'un site ou entre les sites.

La balise `a` possède un attribut `href` qui a pour valeur l'adresse du site cible ou le chemin du fichier que l'on cherche à atteindre (exemple : [Cliquez ici pour vous rendre sur google.fr](#)).

Entre la balise ouvrante et fermante, on trouve le texte qui s'affichera à l'écran (c'est ce texte qui est souligné en bleu). La balise peut sans problème se trouver en plein milieu d'un paragraphe.

## Ajouter des liens (activité 3)

Observez le code ci-dessous :

```

<!DOCTYPE html>
<html lang="fr">
<html>
  <head>
    <meta charset="utf-8">
    <title>Une page web minimaliste</title>
  </head>

  <body>
    <h1>Le HTML</h1>
    <p>Le HTML est un langage de balisage,<br/>les balises vont
    par deux (en général)</p>

```

```

<p>Liste de fruit :
  <ul>
    <li>Pomme</li>
    <li>Poire</li>
    <li>Fraise</li>
  </ul>
</p>
<a href="https://www.w3schools.com" target="_blank" title="Cliquez ici !">Lien

```

```

</body>

```

...

Vous y trouvez des exemples correspondants aux balises présentées au-dessus.

Ajoutez en dessous du lien vers le site W3schools (qui vous servira de modèle pour la syntaxe) un nouveau lien vers l'ENT de l'établissement.

Recherchez l'utilité de l'attribut `target="_blank"` lors de la création d'un lien.

Recherchez l'utilité de l'attribut `title` et l'ajouter à votre lien précédemment créé.

Créez une liste de trois liens (que vous présenterez sous forme d'une liste à puces) comprenant :

- Un lien vers le site public du lycée
- Un lien vers le moteur de recherche Qwant
- Un lien vers le site de l'Onisep

Enregistrez votre travail et observez le résultat.

**Remarque :** Vous trouverez plus d'informations sur la création de liens sur le site [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction\\_%C3%A0\\_HTML/Creati](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Creati)

## Des images, du son et de la vidéo

---

### Une balise pour insérer une image

La balise `<img>` est à la fois ouvrante et fermante comme la balise `<br/>`, cette balise permet d'insérer une image sur votre page web :

```
<img src='images/HTML_logo.png' alt='logo HTML'>
```

Elle possède 2 attributs :

- `src` qui doit être égal au nom du fichier image (ou au chemin si le fichier image se trouve dans un autre dossier).
- `alt` qui doit être égal à une description de votre image (cet attribut est utilisé notamment par les systèmes de description des pages web utilisées par les non-voyants), il faut donc systématiquement renseigner cet attribut.

### Ajouter des images (activité 4)

Dans votre dossier `web`, créez un sous-dossier `images`

Recherchez sur internet des images représentant les logos du HTML, CSS, Python en vérifiant que vous avez bien le droit de les utiliser et les placer dans votre dossier `images`.

Ajoutez le code nécessaire pour faire apparaître les images sur votre page `index.html`

Enregistrez votre travail et observez le résultat.

**Remarque :** Vous trouverez plus d'informations sur la balise `<img>` sur le site <https://developer.mozilla.org/fr/>

### Une balise pour insérer du son

La balise `<audio>` permet d'insérer facilement une piste audio, la syntaxe est proche de la balise `<img>` :

```
<audio controls="controls" src="audio/jingle.mp3" />Votre navigateur ne supporte pas la balise AUDIO.</audio>
```

Ne pas oublier de mettre l'attribut `controls` afin d'afficher le panneau de commande permettant de lancer ou stopper la piste son.

## Ajouter du son (activité 5)

Dans votre dossier `web`, créez un sous-dossier `audio`

Recherchez sur internet une piste audio libre de droit au format mp3 et la placer dans votre dossier `audio`.

Ajoutez le code nécessaire pour permettre la lecture sur votre page `index.html` de la piste audio.

Enregistrez votre travail et observez le résultat.

Plus d'infos : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/audio>

## Une balise pour insérer une vidéo

La balise `<video>` permet d'insérer facilement une vidéo, la syntaxe est proche de la balise `<audio>` :

```
<video controls="controls" src="video/codage.mp4" />Votre navigateur ne supporte pas la balise VIDEO.</video>
```

Ne pas oublier de mettre l'attribut `controls` afin d'afficher le panneau de commande permettant de lancer ou stopper la vidéo.

## Ajouter une vidéo (activité 6)

Dans votre dossier `web`, créez un sous-dossier `video`

Recherchez sur internet une piste vidéo libre de droit au format mp4 et la placer dans votre dossier `video`.

Ajoutez le code nécessaire pour permettre la lecture sur votre page `index.html` de la piste vidéo.

Enregistrez votre travail et observez le résultat.

Plus d'infos : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/video>

## Création d'une page HTML (activité 7)

Maintenant que vous disposez de quelques balises de base, réalisez une page web (ou plusieurs pour les plus avancés) dans laquelle nous trouverons au moins trois paragraphes avec des titres, quelques liens, quelques images... et pourquoi une piste audio et/ou vidéo.

Vous choisirez un thème de votre choix en respectant la législation sur le droit d'usage des médias utilisés.



Vous rendrez votre travail sous la forme d'un fichier site\_nom\_prenom.zip.