

Cours :
Introduction au langage HTML
(HTML signifie « *HyperText Markup Language* »)

Objectifs :

- Connaître la structure de base d'une page web ;

Compétences :

- Aucune compétence directe avec le référentiel.
- Cette introduction constitue un rappel du programme de SNT permettant à terme de traiter le chapitre liés à « Interactions entre l'homme et la machine sur le Web ».

Préambule

- Le langage HTML n'est pas un langage de programmation mais de structuration d'un document ;
- Une page Web n'est pas composée uniquement de HTML. Il faut y ajouter du code CSS (Cascading Style Sheets / Feuille de style en cascade) pour ajouter une couche de style (apparence) ainsi qu'éventuellement du JavaScript pour obtenir de l'interactivité avec le l'utilisateur.

Comment fonctionnent les sites web ?

Pour consulter les sites web, vous lancez un programme appelé le navigateur web en cliquant sur l'une des icônes représentées à la figure suivante (Chrome, Safari, Internet Explorer et Firefox).



Les langages HTML et CSS sont à la base du fonctionnement de tous les sites web. Quand vous consultez un site avec votre navigateur. L'ordinateur se base sur ce qu'on lui a expliqué en HTML et CSS pour savoir ce qu'il doit afficher. Chaque navigateur a sa propre interprétation du code HTML et CSS ce qui peut engendré des affichages parfois différents.

Syntaxe et référence du langage

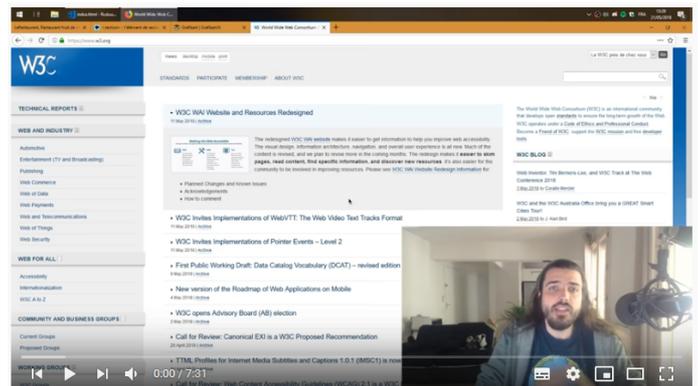
Les éléments du langage sont extrêmement nombreux. Aussi, il est vivement conseillé d'utiliser les sites de référence tels que :
w3schools (<https://www.w3schools.com>) en anglais

Developper Mozilla (<https://developer.mozilla.org/fr/>)

Un outils très utile pour vérifier la conformité de votre code avec les exigences du W3C, le W3C validator (<https://validator.w3.org>)

Ci-contre un tutoriel vidéo sur la façon de l'utiliser :

Dans le but d'un apprentissage autonome, nous vous conseillons l'excellent site de [Grafikart](#) et le site d'[OpenClassRoom](#) ([ici](#) la version pdf du cours d'OpenClassRoom) relatifs entre autres au HTML, CSS, JavaScript et PHP.



Qu'est ce que le HTML réellement ? ⇒ [Lien](#)

HTML est un *langage de balises* qui définit la structure de votre contenu. HTML se compose d'une série d'[éléments](#) utilisés pour entourer, ou envelopper, les diverses parties du contenu pour les faire apparaître ou agir d'une certaine façon.

Les [balises](#) (ou Tags en anglais) entourantes peuvent être abondées d'attributs permettant différentes choses :

- lien hypertexte vers d'autres contenus ;
- mettre un texte en italique ;
- mettre une police plus grande ou plus petite, et ainsi de suite.

Par exemple, la ligne de contenu suivante :

```
1 | Mon chat est très grincheux
```

se décrira de la manière suivante :

```
1 | <p>Mon chat est très grincheux</p>
```

Anatomie d'un élément HTML ⇒ [Lien](#)



Les éléments vides

Certains éléments n'ont pas de contenu. Ces éléments sont appelés **éléments vides**.

Exemple : l'élément `` qui permet d'afficher des images.

```
1 | 
```

Structure de base d'une page Web

```
1 | <!DOCTYPE html>
2 | <html>
3 |   <head>
4 |     <meta charset="utf-8">
5 |     <title>Ma page de test</title>
6 |   </head>
7 |   <body>
8 |     
9 |   </body>
10| </html>
```

Figure 1

- **<!DOCTYPE html>** Permet aux navigateurs de savoir qu'il s'agit de HTML et d'ainsi de l'interpréter en tant que tel.
- **<html></html>** Cet élément encadre tout le contenu de la page. Cet élément est parfois appelé l'élément racine.
- **<head></head>** Cet élément est utilisé comme un container pour toutes les choses qui font partie de la page HTML **mais qui ne sont pas du contenu affiché**. C'est dans cet élément qu'on mettra des mots-clés, une description de la page qui apparaîtra sur les moteurs de recherche, les liens vers les fichiers CSS à utiliser pour la mise en forme, les déclarations des jeux de caractères à utiliser et ainsi de suite.
- **<body></body>** Cet élément est celui qui contient **tout** le contenu que vous souhaitez **afficher** pour qu'il soit vu par les visiteurs : cela peut être du texte, des images, des vidéos, des jeux, des pistes audio jouables, et ainsi de suite.
- **<meta charset="utf-8">** Cet élément définit le jeu de caractères qui devrait être utilisé pour le document. utf-8 regroupe l'ensemble des caractères connus utilisés dans les différents langages humains. Généralement, utf-8 permet de gérer n'importe quel texte que vous pourriez utiliser sur la page.
- **<title></title>** Cet élément définit le titre de votre page. C'est ce titre qui apparaîtra sur l'onglet lorsque la page sera chargée. C'est également ce titre qui sera utilisé pour décrire la page lorsque vous la placez dans vos marques-pages.



- Copié le répertoire « IntroHTML » mis à disposition sur le réseau du lycée dans votre espace classe puis coller le dans votre espace « mes Devoirs ».
- Ouvrir un éditeur de texte de type Notepad++ (surtout pas un traitement de texte de type Word ou LibreOffice Writer...) et créer un fichier que vous nommerez Doctype.html et enregistrerez sous la racine du répertoire « IntroHTML » ;
- Saisissez dans ce fichier la structure de base d'un fichier HTML (figure 1) puis enregistrez-le. ;
- Ouvrez-le avec un navigateur (Firefox ou Chrome).
- Une le fichier ouvert par le navigateur Web, faites un clic droit et sélectionnez « Examiner l'élément ou Inspecter). Vous pouvez ainsi visualiser le code HTML interprété par le navigateur et constater que le contenu de l'élément <head> ne s'affiche pas sur la page utilisateur. Par ailleurs seul l'attribut « alt » s'affiche car l'image déclarée dans l'élément n'est pas trouvée ! C'est normal car le lien vers l'image est faux.
- Changer le nom de l'onglet (actuellement « Ma page de test » dans le fichier source par « Structure de base d'une page HTML »,supprimer la totalité de l'élément et rafraichir la page du navigateur (F5) ;
- Vous avez à présent un document avec la structure de base d'une page html.
- Enregistrez votre fichier source.
- Enregistrez ce fichier source sous un autre nom (par exemple : « Divers_essais » et renommer l'onglet par le même nom. Vous avez à présent un document de base vous permettant de commencer à coder en HTML, document que vous pourrez utiliser pour divers essais dans la suite de ce document.

Baliser le texte ⇒ [lien](#)

Les titres

Les éléments de titre permettent de définir certains textes comme des titres ou sous-titres pour le contenu. D'une certaine façon, ceux-ci fonctionnent comme pour un livre : on a le titre du livre (le plus important) puis les titres des différents chapitres et parfois des sous-titres au sein de ces chapitres. HTML contient des éléments pour 6 niveaux de titres : [<h1>-<h6>](#). La plupart du temps, 3-4 niveaux suffisent amplement :

```
1 | <h1>Mon titre principal</h1>
2 | <h2>Mon titre de section</h2>
3 | <h3>Mon sous-titre</h3>
4 | <h4>Mon sous-sous-titre</h4>
```

Les paragraphes

Les éléments [<p>](#) sont utilisés pour contenir des paragraphes de texte. Vous les utiliserez fréquemment pour placer du texte sur une page

```
1 | <p>Voici un paragraphe</p>
```

Les listes

Une grande partie du contenu sur le Web est présente sous forme de listes. HTML a donc des éléments utilisés pour représenter ces listes. Le balisage des listes contient toujours au moins deux éléments. Les types de listes utilisés fréquemment sont les listes ordonnées et les listes non-ordonnées :

1. **Les listes non-ordonnées** sont des listes pour lesquelles l'ordre des éléments n'a pas d'importance (par exemple une liste d'emoji). La balise utilisée pour ces listes est l'élément `` (**ul** signifie *unordered list* liste non-ordonnée en anglais)
2. **Les listes ordonnées** sont des listes pour lesquelles l'ordre des éléments est important (par exemple une recette). La balise utilisée pour ces listes est l'élément `` (**ol** signifie *ordered list* liste ordonnée en anglais)

Chaque élément d'une liste est balisé avec un élément `` (list item).

Par exemple, si on souhaite modifier un paragraphe en une liste :

```
<p>Mozilla est une communauté mondiale composée de technologues, chercheurs, bâtisseurs travaillant ensemble... </p>
```

On pourrait faire :

```
1 <p>Mozilla est une communauté mondiale composée de </p>
2
3 <ul>
4   <li>technologues</li>
5   <li>chercheurs</li>
6   <li>bâtisseurs</li>
7 </ul>
8
9 <p>travaillant ensemble...</p>
```

Les liens ⇒ [lien](#)

Pour créer un lien, il suffit d'utiliser l'élément `<a>` (le **a** est un raccourci pour « ancre »).

```
<a href="https://www.mozilla.org/fr/about/manifesto/">Manifeste Mozilla</a>
```

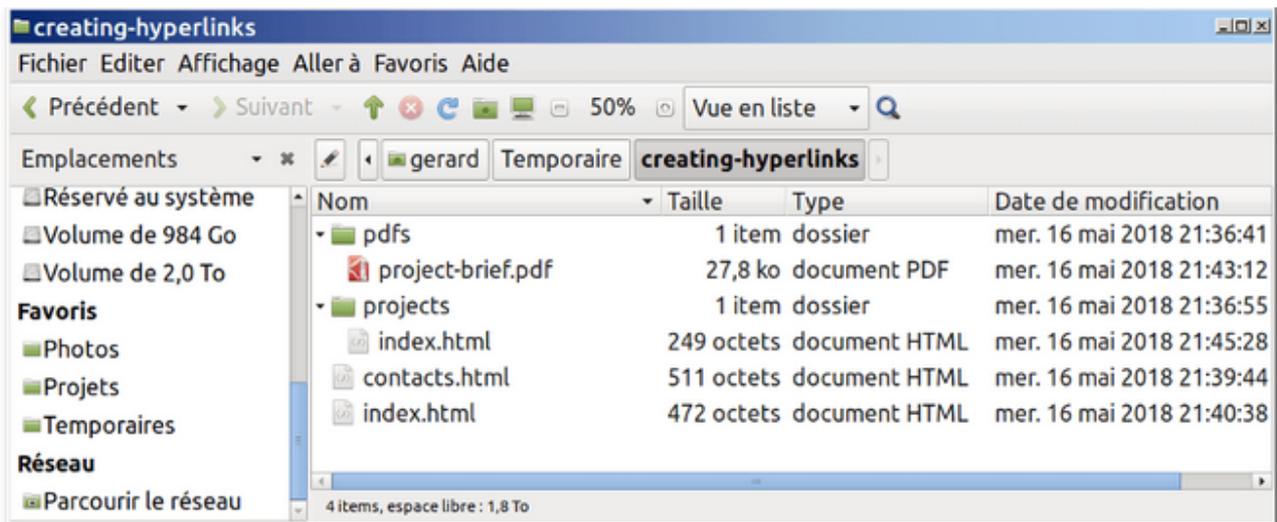
URL et chemins ⇒ [lien](#)

Pour bien maîtriser les liens, vous avez besoin d'avoir compris ce que sont les URL et les chemins. Cette section vous donne les informations pour y parvenir.

Une URL, ou **Uniform Resource Locator**, est simplement une chaîne textuelle qui définit où se situe quelque chose sur le Web. Par exemple, la page d'accueil en anglais de Mozilla est située à l'adresse `https://www.mozilla.org/en-US/`.

Les URL utilisent des chemins pour trouver des fichiers. Les chemins indiquent où dans le système de fichiers, se trouve celui qui vous intéresse.

Regardons un exemple simple de structure de répertoires



La **racine** de cette structure de répertoires s'appelle **creating-hyperlinks**. Dans la racine, nous avons un fichier **index.html** et un **contacts.html**.

Remarques : sur un site réel, `index.html` serait notre page d'accueil ou portail (page web servant de point d'entrée à un site web ou à une section particulière d'un site web).

Il y a aussi deux répertoires dans la racine - **pdfs** et **projects**. Chacun d'eux comporte un seul fichier respectivement un PDF (**project-brief.pdf**) et un fichier **index.html**.

- Pointer vers un fichier dans le même répertoire :

```
<a href="contacts.html">Contacts</a>
```

- Pointer vers un fichier dans un sous-répertoire :

```
<a href="projects/index.html">page d'accueil</a>
```

- Pointer vers un répertoire en montant dans les répertoires parents :

La syntaxe « `../` » permet de remonter au répertoire parent. Pour remonter de deux répertoire parents il faut noter « `../../` »

```
<a href="../pdfs/project-brief.pdf">sommaire de projet</a>
```

Les images ⇒ [Lien](#)

L'élément `` permet d'intégrer une image dans la page, à l'endroit où l'élément apparaît. L'image utilisée est définie via l'attribut `src` (pour source) qui contient le chemin vers le fichier de l'image.

```
1 | 
```

L'attribut `alt` (pour alternatif) permet une description de l'image à l'intention des utilisateurs qui ne peuvent pas voir l'image pour quelle que raison que ce soit.



- Collez le bout de code précédent dans la partie `<body>` de votre programme et actualisé votre navigateur Web (F5). Pourquoi l'image ne s'affiche t'elle pas ?
- Téléchargez une image prise sur le Web et enregistrez dans le sous-répertoire « images » présent dans le répertoire « introHTML » ;
- Faites en sorte que votre image s'affiche sur le navigateur Web.

Utilisation des références du langage

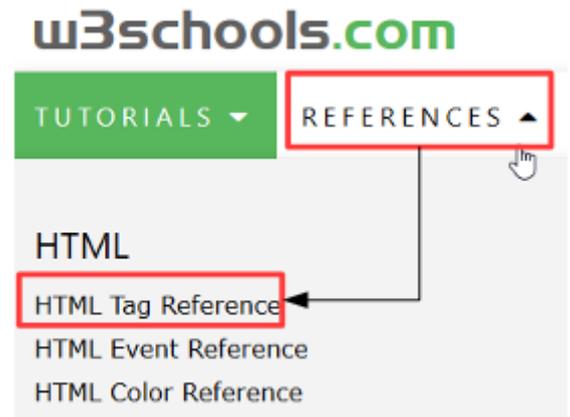
Vous venez d'avoir un très bref aperçu de la syntaxe HTML.

Les langages Web étant un puit sans fond, l'idée est de vous faire utiliser la documentation de référence.

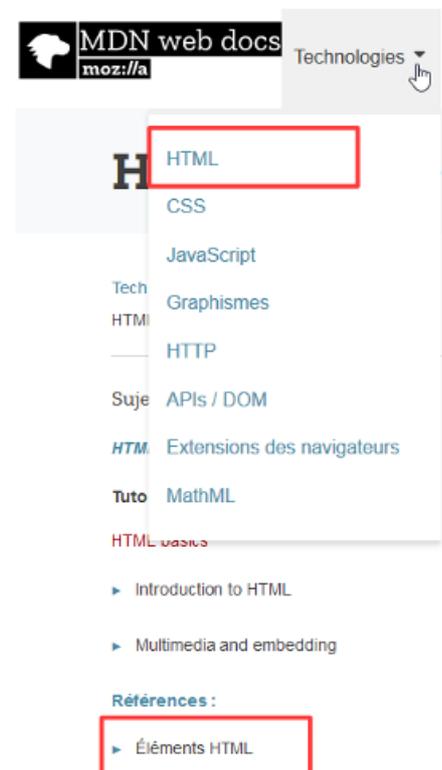
Les deux sites de références à privilégier sont :

- w3schools (<https://www.w3schools.com>) en anglais
- Developer Mozilla (<https://developer.mozilla.org/fr/>)

Pour commencer et ne pas trop mélanger le HTML et le CSS, nous utiliserons pour la partie à venir le site **w3schools** et en particulier les éléments de du langage HTML (figure ci-contre)



A titre d'information, vous pouvez retrouver les explications équivalentes sur Developer Mozilla mais avec une surcroupe CSS donc à éviter d'utiliser dans un premier temps (figure ci-après).





A partir du fichier « Rudiments_html.html » présent dans le répertoire « introHTML3, on vous demande d'enrichir ce fichier en explorant quelques éléments supportés par le HTML5. Nous rappelons que vous devez utiliser le site w3schools (<https://www.w3schools.com>).

Vous vérifierez systématiquement qu'ils sont supportés par les principaux navigateurs Web



[
](#) - [](#) - [](#) - [<abbr>](#) - [<hr>](#) - [<pre>](#) - [<select>](#) - [<sub>](#) - [<sup>](#) - [<table>](#) - [<textarea>](#) - [<div>](#) - [](#) - [<button>](#) - [<form>](#)

Les attributs

Les **attributs** sont utilisés pour **personnaliser les balises**. D'une manière ou d'une autre, un jour vous voulez modifier la taille d'une image, d'un tableau ou changer la couleur d'une police. Tout ceci est possible avec l'aide des attributs.

Dans l'exemple ci-après, l'attribut **class** permet d'utiliser un nom pour identifier l'élément et ce nom pourra être utilisé plus tard pour la mise en forme ou autre chose.



Nous avons déjà rencontrés quelques attributs :

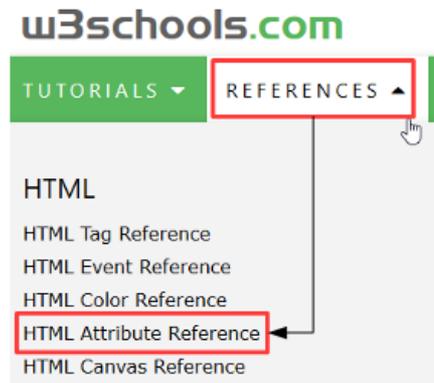
Avec l'élément `<a>` (`href` est un attribut)

```
<a href="contacts.html">Contacts</a>
```

Avec l'élément `` (`src` et `alt` sont des attributs)

```
1 | 
```

Les attributs sont très nombreux... pour preuve, cliquer sur l'image ci-contre.



Certains attributs sont globaux (valides pour tous les éléments) et d'autres sont valides pour certains éléments seulement. Pour exemple ci-dessous, l'attribut « accesskey » est global alors que « accept » est valable pour l'élément <input>.

Attribute	Belongs to	Description
accept	<input>	Specifies the types of files that the server accepts (only for type="file")
accept-charset	<form>	Specifies the character encodings that are to be used for the form submission
accesskey	Global Attributes	Specifies a shortcut key to activate/focus an element

Vous retrouverez [ici](#) une liste des attributs globaux.

Attribut « title »

([Lien](#))

```
<p><abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
<p title="Free Web tutorials">W3Schools.com</p>
```

Résultat :

WHO was founded in 1948.

World Health Organization

La description mise pour « title » s'affiche dans le navigateur lorsque la souris est positionnée sur le mot WHO.

Attribut «style»

```
<h1 style="color:blue;text-align:center">This is a header</h1>  
<p style="color:green">This is a paragraph.</p>
```

Résultat :

This is a header

This is a paragraph.

Cet attribut « style » fait référence aux propriétés du CSS (cf. [mémento HTML / CSS](#))



A partir du fichier « Attributs – devoir maison.html », vous devez le modifier afin d’obtenir le résultat final de l’annexe 1 (fichier « Attributs - devoir maison.pdf » présent dans le répertoire « introHTML »).

Annexe 1

J'apprends à manipuler quelques attributs

Modifier cette ligne pour qu'elle apparaisse en vert

Modifier ce contenu pour que le mot **NSI** seul s'affiche en rouge

Modifier ce contenu afin que l'info bulle "Numérique Sciences Informatiques" s'affiche lorsque l'on positionne la souris sur NSI.

Modifier la <div> suivante pour que sa couleur soit bleue claire. (*indice : il faut le faire avec l'attribut "Style"*).

Numérique Sciences Informatiques

This is a heading in a div element

This is some text in a div element.

Modifier l'élément suivant pour faire apparaître l'image "image1.jpg" présente dans le sous-répertoire "images" du répertoire "introHTML". L'image aura une largeur de 250 pixels par 150 pixels de hauteur.



Modifier l'élément <input> pour qu'un message d'aide à la saisie précise ceci "Veuillez entrer votre nom"

Nom :

Ajouter une question à cocher de votre choix et faites en sorte que la case "Vous avez choisi la spécialité NSI soit cochée par défaut"

Cocher les bonnes réponses :

- Vous avez plus de 18 ans
 - Vous avez choisi la spécialité NSI
 - Vous avez des bases en HTML
 - Vous habitez Rennes.
-

Illustrer par des exemples personnels les différents types suivants de l'élément <input> :

- Radio ;
- Password
- Email (essayer d'entre un email sans @ et constater)
- File
- Range
- Color