

12/10/2020

# HTML 5 – Partie 2

# HTML



MODULE B1.PROG

Les liens Hypertextes : balise <a> .....	2
Comprendre les chemins absolus et relatifs .....	2
Lien vers une autre page du même site .....	3
Le lien à l'intérieur d'une page .....	4
Le lien vers un autre site .....	4
Le lien vers une adresse email .....	4
Lien vers un numero de telephone .....	5
Le lien pour télécharger un fichier .....	5
Attributs utiles pour la balise <A> .....	5
Conseils de mise en œuvre .....	5
Le style sur les liens .....	6
Sélecteur AVEC PSEUDOS classes.....	6
pseudos classe pour les liens .....	7
Pseudos classes de paragraphe .....	8
Les pseudo-classes d'insertion .....	8

## LES LIENS HYPERTEXTES : BALISE <A>

L'essence même du Web est de pouvoir naviguer sur différentes pages en suivant des liens qui peuvent être :

- Un endroit de la page en cours.
- Une autre page du site.
- Un endroit dans une autre page du site.
- Une page d'un autre site situé sur le Web.
- Une adresse électronique.
- Un fichier à télécharger.

Les liens apparaissent par défaut soulignés et de couleur bleue pour le lien non visité, de couleur pourpre pour le lien visité et rouge pour le lien actif. Le soulignement est une convention majeure du Web pour indiquer les liens hypertextes. Il est toutefois possible de modifier le style des liens en CSS.

La balise utilisée pour cela est la balise <a> qui signifie **anc**re

## COMPRENDRE LES CHEMINS ABSOLUS ET RELATIFS

Rappelons ici que le protocole principal utilisé pour le Web est **HTTP (Hyper Text Transfer Protocol)**, basé sur le modèle client-serveur dont le fonctionnement est décrit de façon simplifiée ici :

1. Le Client demande une page par l'intermédiaire de son navigateur (Chrome, Firefox, autres). La page demandée ne se trouve pas en local sur son propre poste, mais sur le serveur. La page est demandée sous forme d'URL. Par exemple l'utilisateur demande la page <http://www.monsite.fr/test.html>.

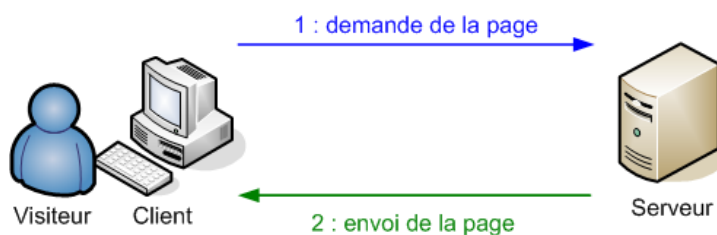
Cette URL se compose en réalité de 2 parties :

- Le serveur : <http://www.monsite.fr>
- La page demandée : /test.html

La demande revient à formuler la requête http suivante : sur le serveur **www.monsite.fr** **GET /test.html**

**Le / correspond à la racine du serveur.** Notons ici que l'on ne sait pas où physiquement se trouvent les fichiers sur le serveur.

2. Le serveur reçoit la requête et l'analyse. Il vérifie qu'il possède la page demandée (/test.html) dans son système de fichiers. Si c'est le cas, la page est renvoyée via le réseau local ou internet vers le client. Le navigateur Client n'a plus qu'à interpréter le code HTML de la page et afficher celle-ci.



Souvent source d'erreurs, de pages introuvables (erreur 404), d'images manquantes ou de styles non appliqués, le format d'écriture des liens est très important. Une erreur classique est de référencer une page web par rapport à un chemin d'accès physique.

Par exemple, le lien suivant n'a aucune chance d'aboutir `<a href="C:\Inetpub\wwwroot\test.html"> page 1 </a>`  
Comme nous l'avons vu précédemment le client, qui va recevoir cette page avec ce lien, va chercher à accéder sur son propre disque le chemin `C:\Inetpub\wwwroot\test.html`

Deux types d'écritures cohabitent, chacune avec leur spécificité propre :

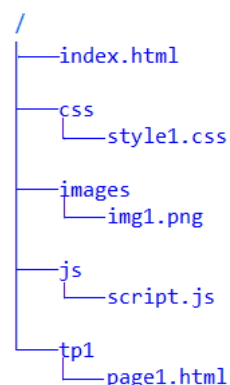
- **Les chemins absolus** : chemin d'accès à une ressource à partir de la racine.
- **Les chemins relatifs** : chemin d'accès à une ressource à partir du répertoire courant.

Supposons l'arborescence suivante. Le "/" est la racine du site. Physiquement, les fichiers se trouvent sur un support de stockage. L'emplacement physique où se trouvent les fichiers nous importe peu.

Accès aux différents fichiers. La racine du site est <http://@IP/>

- En utilisant les chemins absolus :

Ressource	Chemin absolu (1)	Chemin absolu (2)
Index.html	<a href="http://@IP/index.html">http://@IP/index.html</a>	/index.html
Style1.css	<a href="http://@IP/css/style1.css">http://@IP/css/style1.css</a>	/css/style1.css
Script.js	<a href="http://@IP/script/script.js">http://@IP/script/script.js</a>	/script/scripts.js
Page1.html	<a href="http://@IP/tp1/page1.html">http://@IP/tp1/page1.html</a>	/tp1/page1.html
Img1.png	<a href="http://@IP/images/img1.png">http://@IP/images/img1.png</a>	/images/img1.png



- En utilisant les chemins relatifs. Le chemin relatif dépend du répertoire courant où se trouve le fichier HTML ou JS qui contient le lien vers une ressource. Le ".." signifie répertoire parent, et le "." signifie répertoire courant.

Répertoire courant	Ressource	Chemin relatif (1)	Chemin relatif (2)
/	Index.html	index.html	./index.html
tp1/	Index.html	../index.html	
/	Page1.html	tp1/page1.html	./tp1/page1.html
Tp1	Img1.png	../images/img1.png	

## LIEN VERS UNE AUTRE PAGE DU MÊME SITE

Les premiers liens consistent à pointer vers une autre page située dans le site. Il est recommandé d'utiliser ici des liens relatifs.

On utilise l'attribut href pour définir l'adresse (URL) de la cible du lien.

Exemple :

```

<!--lien vers page2.htm se trouvant dans le même répertoire-->
<a href="page2.htm">Lien vers page 2</a>
<a href="./page2.htm">Lien vers page 2</a>

<!--lien vers une page1.htm se trouvant dans dossier1-->
<a href="dossier1/page1.htm">Texte du lien </a>
<a href="./dossier1/page1.htm">Texte du lien </a>

<!--Lien vers page2.htm se trouvant dans le dossier2 du répertoire parent-->
<a href=".. /dossier2/page2.htm">Texte du lien</a>

<!--Lien vers index.htm se trouvant dans le répertoire parent-->
<a href=".. /index.htm">Texte du lien</a>

```

La balise <a> peut contenir d'autres types de balises (h1, div, li, img...)

Exemple :

```

<a href="page2.htm">
  <div>
    <p>les premiers liens consistent à pointer vers une autre page située dans le site. Il est
    recommandé d'utiliser ici des liens relatifs. On utilise l'attribut href pour définir l'adresse (URL) de
    la cible du lien.
    </p>
  </div>
</a>

```

## LE LIEN À L'INTÉRIEUR D'UNE PAGE


<https://codepen.io/samuel-gibert/pen/ExbVXwZ>

Il peut être intéressant de mener l'utilisateur à un endroit précis du document notamment si cette page est une page de texte importante, pour des renvois à des notes ou autres. Dans ce cas on place dans notre page ce que l'on appelle une ancre. Le lien nous mènera à cette ancre.

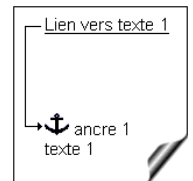
Pour définir notre ancre :

```
<a id="nomAncre"></a>
```

Noter que l'on évite des caractères spéciaux (un espace par exemple) pour définir l'attribut **id**. Autre particularité, il ne peut y avoir dans une même page 2 éléments html ayant le même id

Pour définir le lien vers cette ancre :

```
<a href="#nomAncre">Texte du lien</a>
```



Exemple :

```
<a href="#ch2"> Accès aux chapitres 2</a>
<a id="ch1"></a>
<h1>Chapitre 1</h1>
<p> le texte du chapitre 1 <br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
</p>
<a id="ch2"></a>
<a href="#ch1"> Retour au chapitre 1</a>
<h1>Chapitre 2</h1>
```

Il est également possible de faire référence à une ancre d'une autre page. Exemple :

```
<a href="dossier1/pagel.htm#ancre">Texte du lien</a>
```

Enfin il existe une ancre par défaut **#top** permettant de retourner en haut de la page :

```
<a href="#top">Retourner en haut</a>
```

## LE LIEN VERS UN AUTRE SITE


<https://codepen.io/samuel-gibert/pen/XWzmgVN>

Un lien peut faire référence à des pages d'autres sites, situées à une autre adresse sur le Web. On utilisera ici un adressage absolu. Attention à ne pas oublier le "http://" devant href sinon le lien ne fonctionnera pas.

Exemple :

```
<a href="http://www.w3.org">Lien vers W3C</a>
```

Dans l'exemple précédent, la page vers le site W3C s'ouvre dans la page actuelle. C'est le comportement par défaut de la balise **<a>**. Il est possible d'ouvrir cette page dans un nouvel onglet en utilisant la touche "CTRL+clic" ou clic droit sur le lien puis ouvrir dans un nouvel onglet ou une nouvelle fenêtre.

Il est également possible de modifier le comportement d'ouverture de votre lien avec l'attribut "target". Voici les différentes possibilités :

- `target="_self"` (défaut). Le lien s'ouvre dans la même fenêtre.
- `target="_blank"`. Le lien s'ouvre dans une nouvelle fenêtre du navigateur ou un nouvel onglet.
- `target="_top"` et `"_parent"` : seront étudiés plus tard.

Exemple :

```
<a href="http://www.w3.org" target="_blank">Lien vers W3C</a>
```

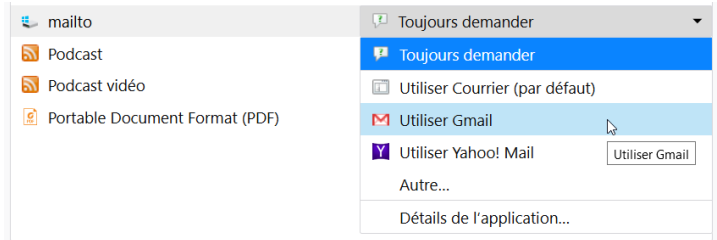
## LE LIEN VERS UNE ADRESSE EMAIL

Le lien ne sera pas ici une page, mais une adresse mail selon la syntaxe suivante :

```
<a href="mailto:samuel.gibert@gmail.com">M'envoyer un mail </a>
```

Le clic sur le lien ouvrira l'application de messagerie électronique par défaut du visiteur : Microsoft Outlook, Thunderbird...

La plupart des navigateurs proposent de modifier l'application qui gère les mails par défaut. Exemple sous Firefox (voir copie d'écran). Il est également possible de modifier les applications par défaut sous Windows.



Il est aussi possible de prédéfinir un objet du message ainsi envoyé ou prévoir l'envoi d'une copie à un autre destinataire.

- Pour prédéfinir un objet à l'e-mail :

```
<a href="mailto:samuel.gibert@gmail.com?subject=Html5"> m'envoyer un mail </a>
```

- Pour définir une copie du message :

```
<a href="mailto:samuel.gibert@gmail.com?cc=autres@gmail.com"> m'envoyer un mail </a>
```

- Pour combiner les possibilités :

```
<a href="mailto:samuel.gibert@gmail.com?subject=Html5&cc=autres@gmail.com">m'envoyer un mail </a>
```

## LIEN VERS UN NUMÉRO DE TÉLÉPHONE

Dans même style que pour l'adresse mail il est possible d'inclure dans l'attribut href un numéro de téléphone. Utile dans la navigation sur un smartphone.

```
<a href="tel:+6000000000">+6 15 70 15 64</a>
```

## LE LIEN POUR TÉLÉCHARGER UN FICHIER

Pour permettre de télécharger un fichier, il suffit de spécifier le nom (avec l'extension) dans l'attribut href.

Exemple avec un fichier PDF.

```
<a href="./pdf/Cours_HTML_5.pdf" target="_blank"> téléchargez le cours </a>
```

Dans le cas de fichier PDF, le navigateur propose d'ouvrir le fichier dans le navigateur. Il en sera de même pour tous les autres fichiers reconnus. Quand il n'y a pas d'application définie par défaut pour l'extension du fichier, le navigateur télécharge le fichier.

Exemple avec un fichier ZIP:

```
<a href="./archive/Cours_HTML_5.zip"> téléchargez le cours </a><br>
```

## ATTRIBUTS UTILES POUR LA BALISE <A>

Nous avons vu l'attribut "href", et "target". Il existe d'autres attributs bien utiles.

- title="valeur" : permet d'afficher une info bulle au-dessus ou en dessous du lien
- download : permet de spécifier que le fichier doit obligatoirement être téléchargé quel que soit le type du fichier (PDF, html, zip, exe ,...)
- hreflang="valeur" : cet attribut indique de façon purement indicative la langue de la ressource liée.
- Type = "type mime" : type de document du lien : information purement indicative.

## CONSEILS DE MISE EN ŒUVRE

Ce chapitre est un résumé de l'article suivant avec pour chaque conseil un bon et un mauvais exemple :

[https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Cr%C3%A9er\\_un\\_hyperlien](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Cr%C3%A9er_un_hyperlien)

- Le texte d'un lien doit être compréhensible seul :

```
<a href="https://firefox.com/"> Télécharger Firefox </a>
<a href="https://firefox.com/"> Cliquez ici</a> pour télécharger Firefox
```

- N'ajoutez pas « lien vers » dans le texte du lien:

```
<a href="https://developer.mozilla.org/Firefox_OS"> En savoir plus sur Firefox OS</a>
Voici un <a href="https://developer.mozilla.org/Firefox_OS"> lien sur Firefox OS </a>
```

- Indiquez la cible du lien plutôt que de copier-coller une URL

```
<a href="https://careers.mozilla.org/university/"> Candidatez pour un stage chez Mozilla </a>
<a href="https://careers.mozilla.org/university/"> https://careers.mozilla.org/university/ </a>
```

- Le texte d'un lien doit être concis

```
Voici une page où vous pouvez <a href="https://www.mozilla.org/firefox/all/"> télécharger
Firefox </a>
Voici une page où <a href="https://www.mozilla.org/firefox/all/"> vous pouvez télécharger
Firefox en 90 langues différentes, pour quatre systèmes d'exploitation différents </a>
```

## LE STYLE SUR LES LIENS

Dans ce cours vous trouverez les propriétés CSS permettant de mettre en forme les **liens** HTML.

Tous ce que nous avons vu comme style sur le texte d'applique également sur la balise `<a>` puisqu'un lien est aussi du texte. Les propriétés **background-color**, **color**, **text-xxxx**, **font-xxxxx**, et **autres**, propriétés peuvent être utilisés sur cette balise.

Afin de permettre la modification du style des liens, il faut comprendre la notion de pseudoclasses en CSS.

Une pseudoclasse est un mot-clé qui peut être ajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel l'élément doit être pour être ciblé par la déclaration.

## SÉLECTEUR AVEC PSEUDOS CLASSES



<https://codepen.io/samuel-gibert/pen/yLPYGqz>

Les pseudo classes vont nous permettre de changer le style d'un élément HTML selon son **état**.

Par exemple, la pseudoclasse **:hover**, permettra d'appliquer une mise en forme spécifique lorsque l'utilisateur survole l'élément ciblé par le sélecteur.

L'élément HTML sera alors dans l'**état** "survolé".

La syntaxe des pseudoclasses est : (le sélecteur peut être : type, identification, class, hiérarchique)

```
Sélecteur : pseudoclasse {
    déclaration(s) de style
}
```

Exemple :

```
div : hover { //Quand une balise div est survolé on modifie le style
    background-color: #F89B4D;
}
```

Voici une liste des pseudos classes standard : <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

:active	:any	:any-link	:checked	:default	:dir()
:disabled	:empty	:enabled	:first	:first-child	:first-of-type
:fullscreen	:focus	:hover	:indeterminate	:in-range	:invalid
:lang()	:last-child	:last-of-type	:left	:link	:not()
:nth-child()	:nth-last-child()	:nth-last-of-type()	:nth-of-type()	:only-child	:only-of-type
:optional	:out-of-range	:read-only	:read-write	:required	:right
:root	:scope	:target	:valid	:visited	

Certaines de ses pseudo-class sont spécifique à certains éléments HTML et d'autres, comme par exemple **:hover** sont utilisables sur quasiment tous les éléments HTML.

**En résumé** : une pseudoclasse cible un élément dans un certain état, comme si nous avions ajouté une classe sur cet élément.

### PSEUDOS CLASSE POUR LES LIENS

Ces pseudoclasses sont généralement bien utiles pour les liens (<a>). Pour obtenir un résultat correct, il est indispensable d'énoncer les apparences de lien dans l'ordre suivant :

- **:link** permet de définir l'apparence d'un lien qui n'a pas encore été sélectionné.
- **:visited** permet de définir l'apparence d'un lien qui a déjà été visité.
- **:hover** permet de définir l'apparence d'un lien au passage du curseur de la souris.
- **:focus** (état d'un élément qui a reçu «l'attention»), par exemple un lien lorsqu'on y accède grâce au clavier, ou un champ texte d'un formulaire lorsqu'on clique dedans)
- **:active** permet de définir l'apparence d'un lien au moment où il est cliqué par le visiteur (entre le clic et le relâchement du clic)

Notons que ici que **:link** et **:visited** sont réservés aux liens (élément HTML a)

HTML	CSS externe
<pre>&lt;a href="http://www.google.fr" target="_blank"&gt; vers google &lt;/a&gt;</pre>	<pre>a:link {     color: red;     text-transform: uppercase; } a:visited {     text-decoration: underline red;     color:blue; } a:hover { font-weight: bold;           font-style: italic;           font-size: 1.5em; } a:active {     text-decoration: underline overline green; }</pre>
VISUEL	
<p>Link : <u><b>VERS GOOGLE</b></u>      hover : <u><b>VERS GOOGLE</b></u>      active : <u><b>VERS GOOGLE</b></u></p> <p>Visited : <u>VERS GOOGLE</u></p>	



## PSEUDOS CLASSES DE PARAGRAPHE

- `:first-letter` permet de définir l'apparence de la première lettre d'un paragraphe.
- `:first-line` permet de définir l'apparence de la première ligne d'un paragraphe.

Exemple :

HTML	CSS externe
<pre>&lt;p&gt;Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ... anim id est laborum.&lt;/p&gt;</pre>	<pre>p:first-letter {     text-transform: uppercase;     font-size: 1.5em;     color: blue; }  p::first-line{     font-weight: bold; }</pre>
<b>VISUEL</b> <p><b>L</b>orem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>	

## LES PSEUDO-CLASSES D'INSERTION

- La pseudoclasse `:before`, associée à la propriété `content`, permet d'insérer un contenu avant un élément.
- La pseudoclasse `:after`, associée à la propriété `content`, permet d'insérer un contenu après un élément.

Ces classes peuvent se combiner avec d'autre pseudoclasses

Exemple:

HTML	CSS externe
<pre>&lt;p class="tel"&gt; 06012345 67 &lt;/p&gt; &lt;p class="mail"&gt; samuel.gibert@ac-grenoble.fr &lt;/p&gt;</pre>	<pre>.tel:before {     content: ' ☎ - '; } .mail:after {     content: ' -✉ '; } a:hover:before {     content: ' ➡ '; }</pre>
<b>VISUEL</b> <p>☎- 06 01 23 45 67</p> <p>samuel.gibert@ac-grenoble.fr ✉</p> <p style="text-align: right;">:hover ➡ <u><b>VERS GOOGLE</b></u></p>	