

Chapitre 1: Le langage HTML

Langage HTML (Hypertext markup Language)

1. HTML C'est quoi?
2. Comment écrire une page HTML?
3. Balises, attributs et caractères spéciaux
4. Balises de base
5. Balises de formatage du texte
6. Paragraphes, conteneurs (ou containers) et listes
7. Images et liens hypertextes
8. Les tableaux
9. Les formulaires
10. Balises spéciales
11. Les balises META
12. Validation et compatibilité HTML - Les moteurs de rendu
13. Exercice - Présentation d'une page HTML basique

Chapitre 1: Le langage HTML

HTML (HyperText Markup Language) est un langage de description (dit de marquage) de pages Web. Il permet de présenter les documents hypertextes destinés à être affichés sur le navigateur. Il s'agit d'un langage côté client (tout comme CSS et Javascript). Il est supporté et développé par W3C*.

L'origine du HTML remonte au début du Web. En effet, il a été inventé vers les années 1989 afin qu'il puisse présenter les documents qui circulent sur la toile et établir des liens entre eux à travers les liens hypertextes (ou hyperliens).

- Versions du langage

En 1989 HTML1.0 est apparu, il se basait sur les spécifications du standard SGML (Standard Generalized Markup Language) qui est aussi un langage de description à balises. Puis HTML2.0 a vu le jour en 1995. En 1997, HTML3 puis HTML4 sont apparus.

HTML4 est la version du langage la plus populaire, elle est toujours supportée de nos jours malgré l'apparition du standard HTML5.

* Le World Wide Web Consortium, abrégé par le sigle W3C, est un organisme de standardisation chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML5, HTML, XHTML, XML, RDF, SPARQL, CSS, PNG...

Chapitre 1: Le langage HTML

L'objectif de ce cours est d'apprendre à utiliser le langage HTML.

Les outils dont vous aurez besoin

Pour bien maîtriser le langage HTML, il faut pratiquer. Il vous faudra donc un [éditeur](#) de code et un navigateur pour tester ce que vous avez codé.

Il existe des éditeurs WYSIWYG (pour What You See Is What You Get) qui permettent de construire facilement un document HTML sans avoir besoin de connaître la syntaxe.

Mais pour devenir un bon développeur il faut essayer d'écrire le code source par soi-même.

Je vous recommande donc de vous servir d'un éditeur texte comme : Notepad++, Sublime, Visual Studio Code, ...
N'importe quel navigateur fera l'affaire. De préférence utiliser chrome.



Chapitre 1: Le langage HTML

Format d'une page HTML

Une page HTML est un fichier texte (après tout, le code HTML est du texte) avec l'extension .html ou .htm. Il n'y a absolument aucune différence entre les deux extensions. La première est l'extension normale alors que la deuxième est plus courte et aussi plus adaptées aux anciens systèmes d'exploitation qui comprenaient seulement les extensions en 3 caractères.

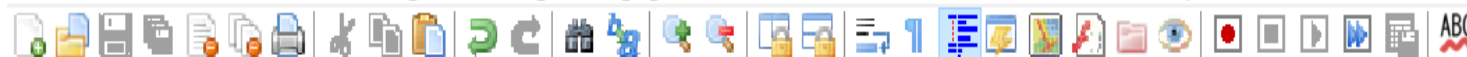
Les fichiers HTML sont déposés dans un serveur de fichiers. Il ne nécessite donc aucune exécution par le serveur car HTML est un langage côté client. Par conséquent, quand vous créez votre fichier HTML, il n'y a aucun problème à le mettre n'importe où dans votre ordinateur (au bureau par exemple) car c'est le navigateur qui l'exécutera après ouverture.

Les développeurs ont pour coutume de nommer la page d'accueil de leur site Web "index", est puisque c'est un document HTML alors son nom complet sera "index.html". Essayez de faire de même.

Chapitre 1: Le langage HTML

A quoi ressemble un fichier HTML?

Fichier Édition Recherche Affichage Encodage Langage Paramètres Outils Macro Exécution Compléments Documents ?



```
index.html x
1 <html>
2   <head>
3     <title>Ma première page HTML</title>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8">
5   </head>
6   <body>
7     <h1>Il s'agit d'un exemple de HTML</h1>
8     <div>
9       <font color="#DD7700"><b>HTML</b></font>
10      est un langage de description de pages Web.
11    </div>
12  </body>
13 </html>
```

Ma première page HTML x +

← → ↻ Fichier | file:///C:/Users/oumay/Desktop/index.html

Il s'agit d'un exemple de HTML

HTML est un langage de description de pages Web.

Un document HTML contient du texte, mais vous avez certainement remarqué la présence des symboles < et >. Ces deux chevrons (ouvrant et fermant) avec le mot qu'ils contiennent constituent ce que l'on appelle une balise.

Un document HTML est structuré sous forme de balises, chaque balise a un sens que le navigateur comprend et exécute pour que le document soit présenté tel que les balises le dictent.

Chapitre 1: Le langage HTML

Balises HTML

Les **balises** constituent l'élément de base du langage HTML. Chaque balise a une signification que le navigateur connaît et applique au contenu. C'est comme ça que la présentation du document est effectuée.

Une balise est représentée par: **le chevron ouvrant + le nom de la balise + le chevron fermant**. Exemple `` ou `<body>`.

Il existe deux types de balises: des balises ouvrantes et des balises fermantes. Elles vont souvent ensemble de telle sorte que la balise ouvrante applique un effet au contenu qui la suit et la balise fermante cesse l'effet en question. La balise fermante se distingue par un slash (/) qui vient juste avant le nom de la balise. Exemple `` est la balise fermante de `` et `</body>` est la balise fermante de `<body>`.

Néanmoins, il existe des balises ouvrantes qui n'ont pas de balises fermantes associées. On les appelle des balises **auto fermantes** ou des balises **orphelines**. Ce sont des balises qui s'ouvrent et se ferment au même moment et elles appliquent leur effet à l'endroit où elles sont déclarées (et pas au contenu qui les suit comme les balises qu'on a vu ci haut). Exemple: `
`. (Saut de ligne)

Chapitre 1: Le langage HTML

Imbrication des balises

Un contenu peut subir l'effet de plusieurs balises à la fois. Si c'est le cas, alors une règle s'impose: En cas d'imbrication de plusieurs balises, la première balise ouverte est la dernière à fermer. Il faut donc fermer les balises dans l'ordre inverse de leur ouverture.

Le code ressemblera à ceci:

```
<balise1><balise2><balise3>Contenu</balise3></balise2></balise1>
```

Les attributs

La plupart des balises HTML peuvent appliquer des effets différents selon les préférences du développeur. C'est comme si on peut les personnaliser à notre guise. C'est là où les attributs interviennent.

Les attributs sont déclarés dans la balise ouvrante. Ils représentent des paramètres qui personnalisent la balise où ils sont définis et ils possèdent des valeurs. Une balise peut renfermer plusieurs attributs à la fois, chacun avec sa valeur.

```
<balise attribut1="valeur1" attribut2="valeur2" attribut3="valeur3"...>Contenu</balise>
```

```
<font color="#DD7700"><b>HTML</b></font>  
est un langage de description de pages Web.
```

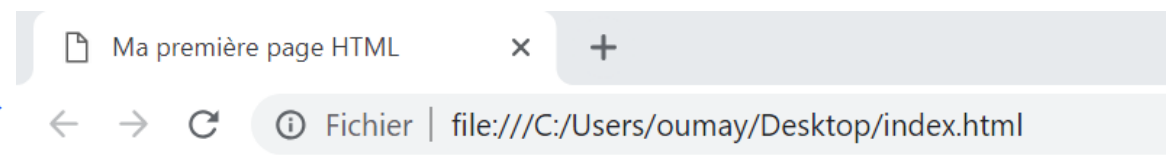
HTML est un langage de description de pages Web.

Chapitre 1: Le langage HTML

Pour que l'écriture du code HTML soit correcte il faut vérifier les points suivants:

- Il ne doit pas y avoir d'espace entre le chevron d'ouverture et le nom de la balise
- Il doit impérativement y avoir un espace (ou plus) entre le nom de la balise et l'attribut
- Il doit impérativement y avoir un espace (ou plus) entre les attributs successifs
- La valeur de l'attribut doit être déclarée entre des guillemets (doubles quote ou simple quote). Cependant les navigateurs peuvent tolérer l'absence des guillemets, mais votre code reste invalide du point de vue de W3C(*).

```
index.html x
1 <html>
2   <head>
3     <title>Ma première page HTML</title>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8">
5   </head>
6   <body>
7     <h1>Il s'agit d'un exemple de HTML</h1>
8     <br>
9     <div>
10      <font color="#DD7700"><b>HTML</b></font>
11      est un langage de description de pages Web.
12    </div>
13
14    <br>
15    <br>
16
17    <font color="red" face="arial">Bonjour à tous</font>
18
19
20  </body>
21 </html>
```



Il s'agit d'un exemple de HTML

HTML est un langage de description de pages Web.

Bonjour à tous

Chapitre 1: Le langage HTML

Les caractères spéciaux

HTML est constitué principalement de balises. Mais ce n'est pas tout car il y a des séquences, autres que les balises, que le navigateur reconnaît et remplace par leurs significations. On appelle ces séquences: caractères spéciaux.

Le code HTML des caractères spéciaux commence par le symbole "&" et fini par ";". Par exemple © signifie "©".

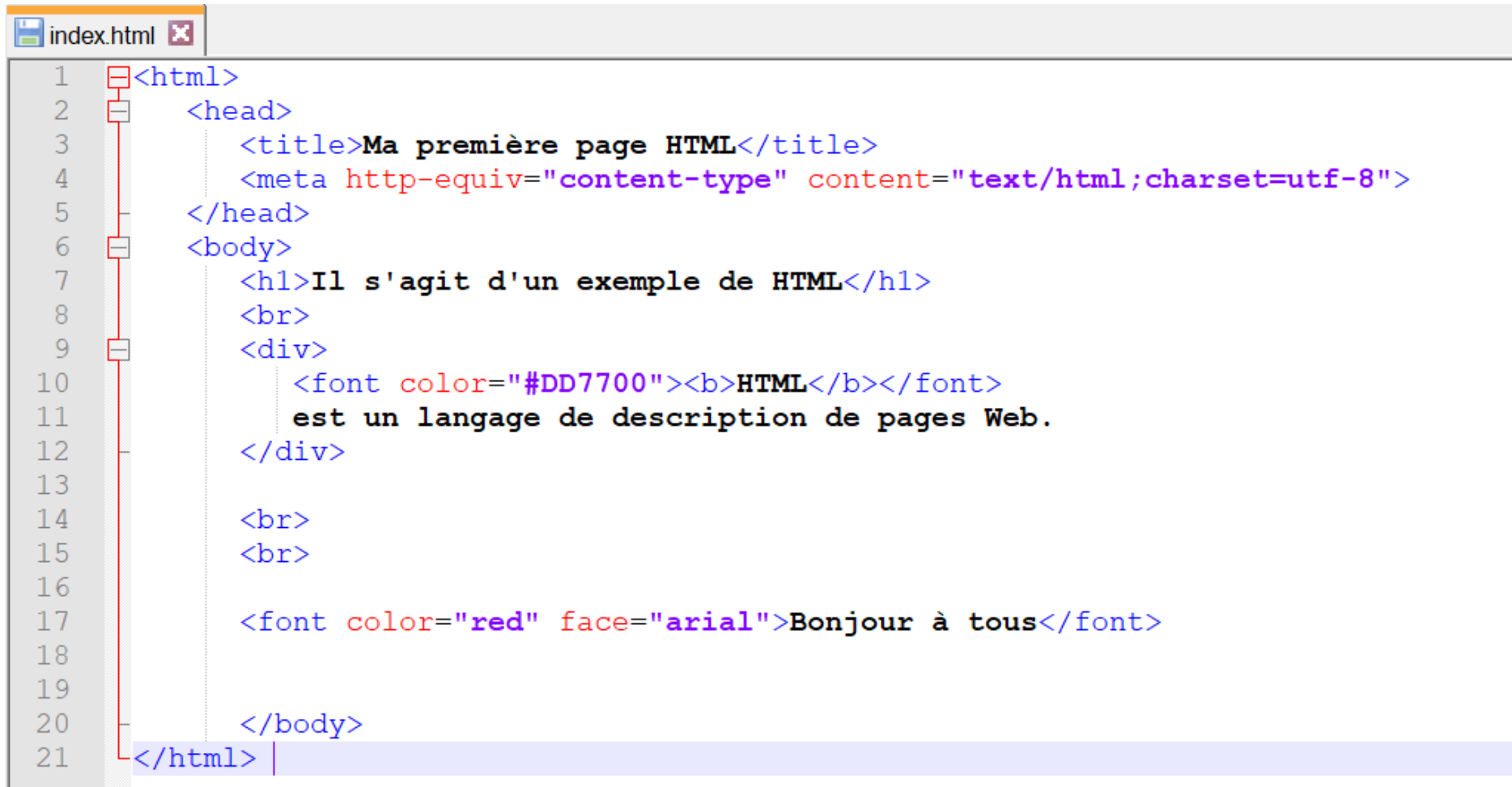
Carctère	Code HTML
"	"
&	&
<	<
>	>
oe	œ
Espace	
£	£
©	©

®	®
±	±
μ	µ
½	½
Ç	Ç
æ	æ

Chapitre 1: Le langage HTML

Les balises de bases

Les trois balises HTML minimales sont: <html>, <head> et <body>.

A screenshot of a code editor window titled 'index.html'. The editor displays a valid HTML document structure. On the left side, there is a vertical line with square markers indicating the nesting of the HTML tags: a red square for the root <html> tag, a blue square for the <head> tag, and a blue square for the <body> tag. The code itself is as follows:

```
1 <html>
2   <head>
3     <title>Ma première page HTML</title>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8">
5   </head>
6   <body>
7     <h1>Il s'agit d'un exemple de HTML</h1>
8     <br>
9     <div>
10      <font color="#DD7700"><b>HTML</b></font>
11      est un langage de description de pages Web.
12    </div>
13
14    <br>
15    <br>
16
17    <font color="red" face="arial">Bonjour à tous</font>
18
19
20  </body>
21 </html>
```

Chapitre 1: Le langage HTML

Les balises de bases

Les trois balises HTML minimales sont: `<html>`, `<head>` et `<body>`.

- Balise `<html>`

La balise `<html>` est la balise de premier niveau. Elle renferme tout le contenu de la page Web. Elle est déclarée au début du document. La balise fermante `</html>` vient clôturer le document en la déclarant à la fin.

- Balise `<head>`

La balise `<head>` représente l'entête du document. Elle contient des données supplémentaires qui ne sont pas nécessairement affichées sur le navigateur, mais elle peuvent fournir des informations complémentaires à l'exécution de la page par celui-ci, ou des informations utiles aux outils de recherche en vue de référencement.

Une des balises les plus utilisées dans la balise `<head>` est la balise `<title>`. Cette balise indique le titre du document. Celui-ci sera affiché sur la barre de titre du navigateur. C'est une balise qui doit être fermée pour indiquer la fin du titre.

On trouve aussi la balise `<meta>` qui fournit des informations supplémentaires sur le contenu.

Chapitre 1: Le langage HTML

Les balises de bases

Les trois balises HTML minimales sont: <html>, <head> et <body>.

- Balise <body>

C'est la plus importante parmi les balises vues précédemment. En effet, la balise <body> renferme le contenu du corps. C'est ce qui sera réellement visible sur le navigateur par les internautes.

La balise <body> doit aussi être fermée pour indiquer la fin du corps du document. Elle peut contenir des attributs, mais la plupart d'entre elles ne sont pas compatibles.

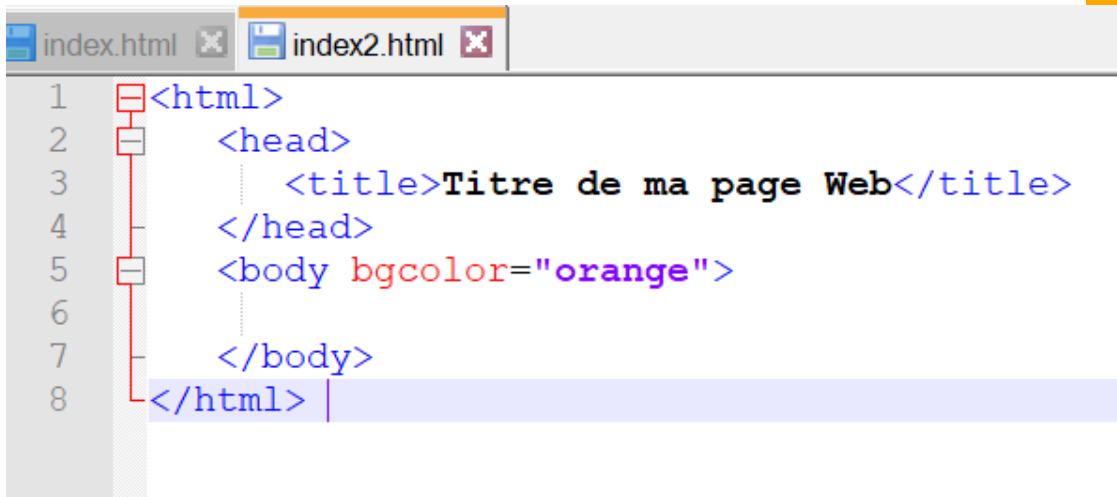
Cependant <body> peut contenir un attribut plus au moins utilisé, il s'agit de bgcolor qui permet de colorer l'arrière-plan de la page Web en entier. La balise est donc déclarée ainsi:

```
<body bgcolor="orange"></body>
```

La couleur d'arrière plan sera donc orange. Il suffit de mentionner le nom de la couleur en anglais pour que celle-ci soit reconnue en HTML. Cependant il existe une manière plus efficace pour déclarer les couleurs qu'on va voir après.

Chapitre 1: Le langage HTML

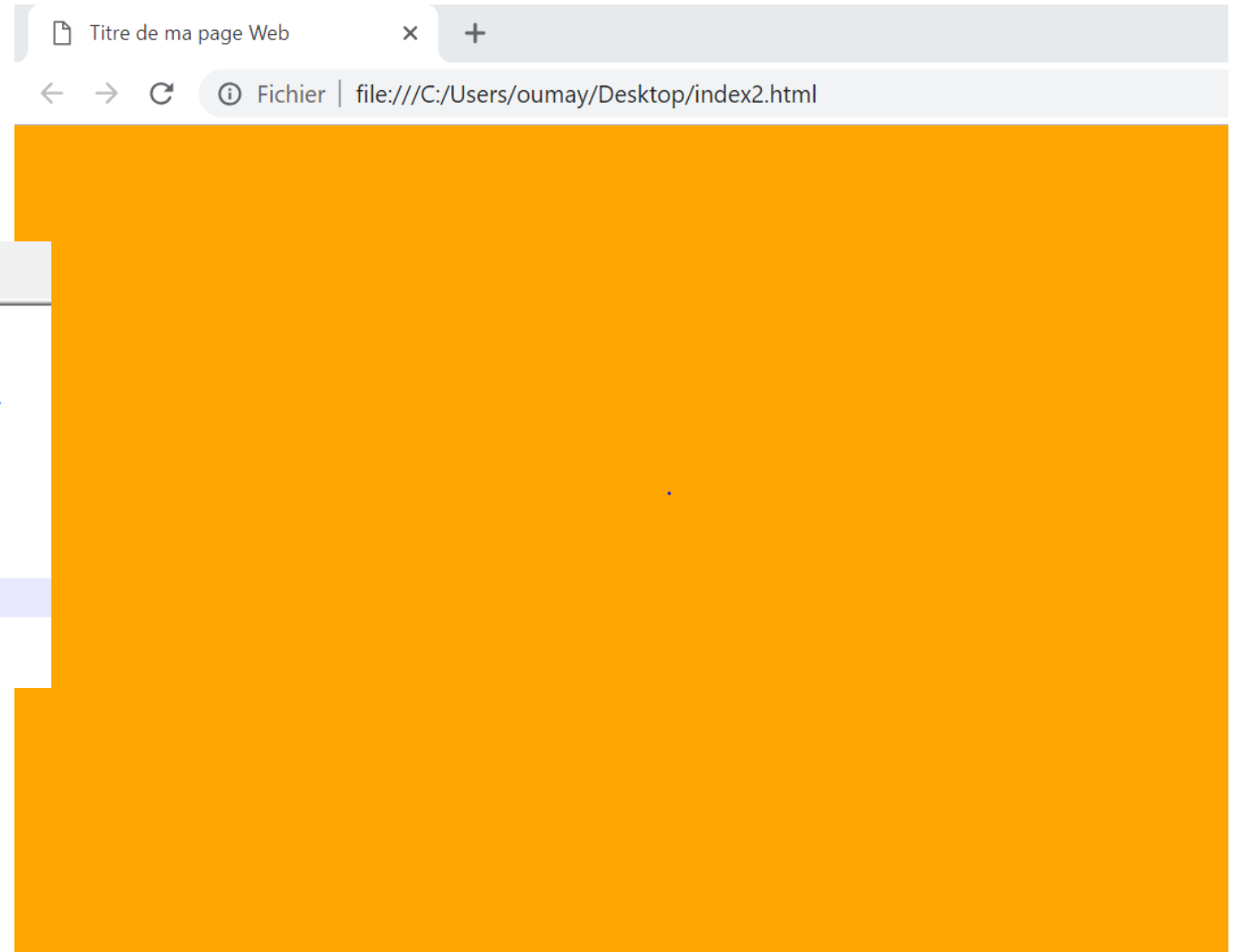
Les balises de bases



A screenshot of a code editor with two tabs: 'index.html' and 'index2.html'. The 'index2.html' tab is active, showing the following HTML code:

```
1 <html>
2   <head>
3     <title>Titre de ma page Web</title>
4   </head>
5   <body bgcolor="orange">
6
7   </body>
8 </html> |
```

The code is color-coded: tags are blue, attributes are red, and text is black. A vertical line on the left indicates line numbers 1 through 8. The cursor is at the end of line 8.



Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Balise `` et ``

La balise `` met le texte qu'elle contient en gras (bold). C'est utile pour les textes importants ou les titres.

Par exemple:

```
<b>Ce texte est écrit en gras</b>
```

Ce qui donne:

Ce texte est écrit en gras

La balise `` met aussi le texte en gras, elle peut remplacer la balise ``, bien que celle-ci est plus populaire.

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Balise <u>

La balise <u> (pour underline) souligne le texte qu'elle entoure. C'est utile pour attirer l'attention à un mot ou une phrase.

Par exemple:

```
<u>Texte souligné</u>
```

Ce qui donne:

Texte souligné

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Balise <i>

La balise <i> (pour italic) met le texte en italique. C'est pratique pour les citations ou les notes.

Par exemple:

```
<i>Texte en italique</i>
```

Ce qui donne:

Texte en italique

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Balise <s>

La balise <s> (pour strike) barre le texte. C'est pratique pour marquer un contenu obsolète ou invalide (par exemple: sur les sites E-commerce, on barre généralement les anciens prix d'articles en promotion, pour mettre en valeur le nouveau prix).

Par exemple:

```
<s>Texte barré</s>
```

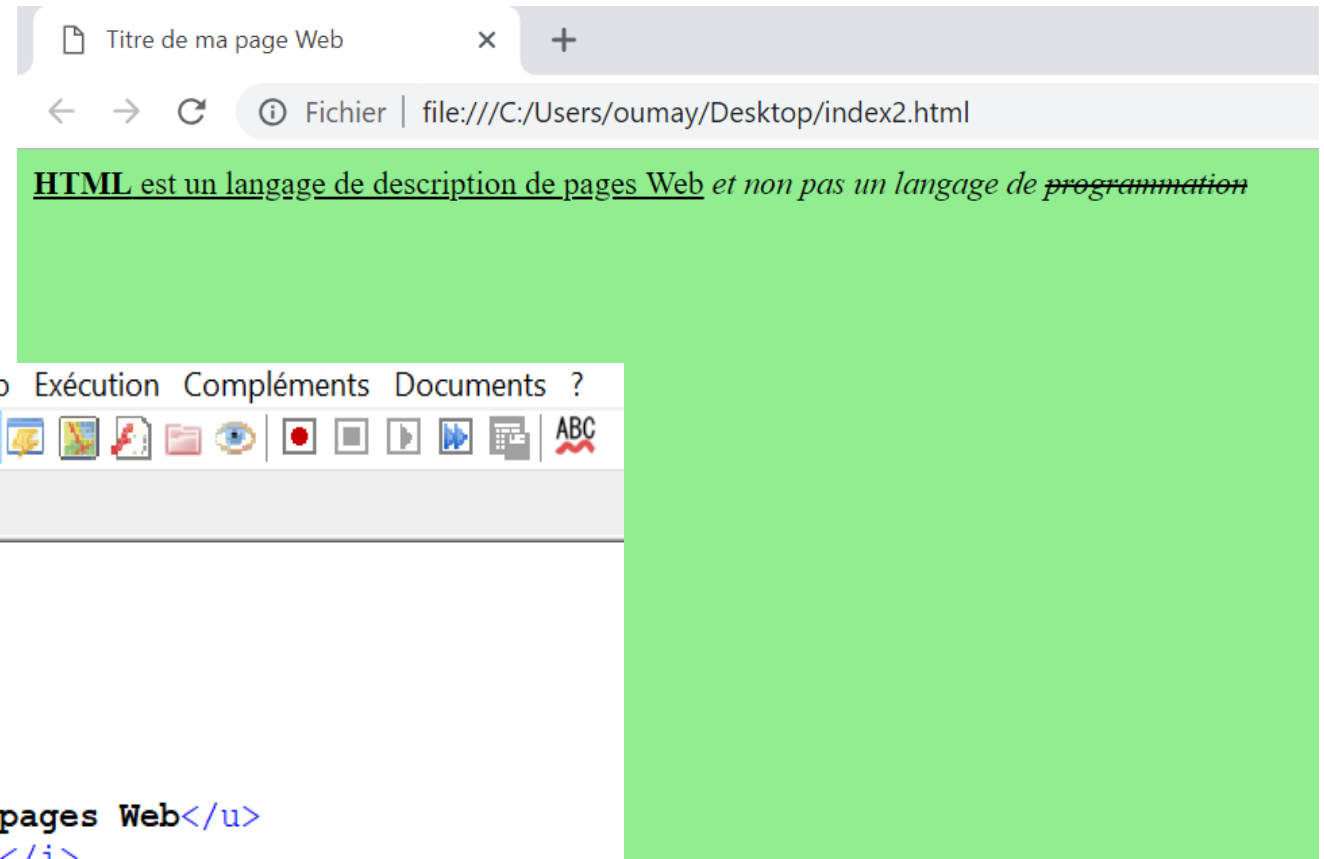
Ce qui donne:

~~Texte barré~~

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:



```
Fichier  Édition  Recherche  Affichage  Encodage  Langage  Paramètres  Outils  Macro  Exécution  Compléments  Documents  ?  
index.html  index2.html  
1  <html>  
2  <head>  
3    <title>Titre de ma page Web</title>  
4  </head>  
5  <body bgcolor="lightgreen">  
6  
7    <u><b>HTML</b> est un langage de description de pages Web</u>  
8    <i>et non pas un langage de <s>programmation</s></i>  
9  
10  </body>  
11 </html>  
12
```

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Balises <h1>, <h2>...<h6>

La balise <h1> (heading) désigne un titre de premier niveau. Le texte qu'elle entoure est de grande taille et mis en gras. Il est utile pour créer les grands titres d'une pages. La balise <h2> désigne un titre de deuxième niveau, elle applique le même effet que la balise précédente mais avec une taille de caractères légèrement plus petite. C'est pratique pour les sous-titres. Il existe aussi <h3>, <h4>, <h5> et <h6> avec une taille de caractères de plus en plus petite.

Par exemple:

```
<h1>Titre de premier niveau</h1>
<h2>Titre de deuxième niveau</h2>
<h3>Titre de troisième niveau</h3>
<h4>Titre de quatrième niveau</h4>
<h5>Titre de cinquième niveau</h5>
<h6>Titre de sixième niveau</h6>
```

Ce qui donne:

Titre de premier niveau

Titre de deuxième niveau

Titre de troisième niveau

Titre de quatrième niveau

Titre de cinquième niveau

Titre de sixième niveau

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Balises `<sup>` et `<sub>`

La balise `<sup>` met un texte en exposant et la balise `<sub>` le met en indice.

Par exemple, le code suivant:

```
x<sup>2</sup>
```

donne:

x^2

et ce code

```
H<sub>2</sub>O
```

donne

H_2O

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Retour à la ligne: Balise

Pour retourner à la ligne à n'importe quel moment, il suffit de déclarer la balise `
`. Il s'agit d'une balise orpheline (donc pas de `</br>`).

Exemple:

```
Ligne 1 <br>
Ligne 2 <br>
Et ligne 3
```

Ce qui donne le résultat suivant:

```
Ligne 1
Ligne 2
Et ligne 3
```

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Commentaire HTML `<!-- -->`

Un commentaire HTML est déclaré comme ceci: `<!-- Commentaire -->`. Il est visible dans le code source mais ignoré par le navigateur. Il sert à marquer un bloc de code pour que celui-ci soit facilement trouvé et compris lors de sa prochaine manipulation.

Exemple:

```
<!-- Ceci est un commentaire -->  
<i>Le commentaire n'est pas visible sur le navigateur.</i>
```

Le résultat eu sur le navigateur est:

Le commentaire n'est pas visible sur le navigateur.

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage sans attributs:

Balise <pre>

Nous avons vu précédemment que pour intégrer un retour à la ligne il faut déclarer la balise **
, et pour mettre plusieurs espaces ou tabulations il faut passer par le caractère spécial ** **. Cependant, il existe une méthode très simple pour afficher un contenu tel qu'il a été édité dans le code. C'est à dire, si vous mettez une tabulation dans le code, celle ci sera automatiquement reprise dans le navigateur, et si vous sautez une ligne, ce saut de ligne sera aussi visible sur le navigateur. La solution magique s'appelle la balise **<pre>.

Exemple:

```
<pre>
  Là-haut sur la montagne, l'était un vieux chalet.
  Murs blancs, toit de bardeaux,
  Devant la porte un vieux bouleau.
  Là-haut sur la montagne, l'était un vieux chalet.
</pre>
```

Le résultat du code est le suivant:

```
Là-haut sur la montagne, l'était un vieux chalet.
Murs blancs, toit de bardeaux,
Devant la porte un vieux bouleau.
Là-haut sur la montagne, l'était un vieux chalet.
```

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage avec attributs:

Avec la même balise on peut réussir plusieurs effets différents. C'est grâce aux attributs que celle ci renferme. Une des balises les plus populaires de HTML est la balise ****.

Balise ****

La balise de formatage de texte la plus connue est sans doute la fameuse ****. Elle doit sa notoriété aux effets qu'elle peut apporter aux textes qu'elle inclue.

La balise **** dispose de trois attributs principaux qui permettent d'appliquer des effets non négligeables aux textes. Ces trois attributs sont: **face**, **size**, et **color**.

L'attribut **face**:

L'attribut **face** permet de spécifier la police à utiliser pour afficher le texte inclus dans la balise ****. Par défaut c'est la police "Times New Roman" qui est appliquée par la plupart des navigateurs si aucune police n'est spécifiée. "Times New Roman" est une police de la famille "Sérim" et elle n'est pas très adaptée aux pages Web. Cependant d'autres polices pourront bien faire l'affaire comme "verdana" ou "arial" qui sont assez populaires et sont supportés par de nombreux systèmes d'exploitation.

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage avec attributs:

L'attribut size:

L'attribut **size** change la taille du texte. Par défaut le navigateur applique la taille 12 points (écrit 12pt) aux polices, sauf quelques exceptions comme "Netscape Navigator" qui applique par défaut la taille 10pt. (Le point est l'unité de mesure des polices.

HTML ne supporte (malheureusement) que 7 tailles de police seulement, elle sont déclarées par des indices allant de 1 à 7. Le tableau suivant détaille la valeur de chaque indice:

Indice (valeur de l'attribut size)	Valeur réelle en pt
1	8pt
2	10pt
3 (par défaut)	12pt
4	14pt
5	18pt
6	24pt
7	36pt

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage avec attributs:

L'attribut color:

Dans un texte, il n'y a pas que la police et la taille, il y a aussi la couleur. L'attribut **color** sert à appliquer une couleur au texte entouré par la balise ****. La méthode la plus simple pour définir une couleur c'est par son nom anglais (red pour rouge, yellow pour jaune...).

Un problème surgit aussitôt à la surface. Combien de couleurs peut on définir de cette manière? sans doute pas beaucoup (un peu plus de 100 couleurs). Cependant un écran peut afficher environ 16 millions de couleurs. Pour couvrir toutes ces nuances, la seule solution consiste à utiliser les codes des couleurs.

Gestion des couleurs en HTML (codes des couleurs)

Pour qu'un écran affiche une image en couleur, il combine 3 couleurs dites **primaires**. Ces couleurs sont **Rouge**, **Vert** et **Bleu**. On parle alors de la base RVB (RGB en anglais). Le fait de mélanger ces trois couleurs à des proportions différentes donne naissance aux 16 millions de couleurs dont on a parlé au paragraphe précédent.

Techniquement, chaque couleur primaire est codée sur un octet (8 bits). Les trois couleurs sont donc codées sur 24 bits et du coup on peut avoir jusqu'à 16777216 couleurs.

Chapitre 1: Le langage HTML

Balises de formatage

- Balises de formatage avec attributs:

Le tableau suivant illustre les couleurs basiques:

couleur	Code HTML
Rouge	#FF0000
Vert	#00FF00
Bleu	#0000FF
Blanc	#FFFFFF
Noir	#000000
Jaune	#FFFF00
Magenta	#FF00FF
Cyan	#00FFFF
Gris moyen	#888888
Orange	#FF8800

```
<font color="#FF0000" face="verdana" size="2">  
  Ce texte a été formaté grâce à la balise font  
</font>
```

Le résultat obtenu est:

Ce texte a été formaté grâce à la balise font

```
1 <html>  
2   <head>  
3     <title>Titre de ma page Web</title>  
4   </head>  
5   <body bgcolor="00FFFF">  
6     <u><b>HTML</b> est un langage de description de pages Web</u>  
7     <i>et non pas un langage de <s>programmation</s></i>  
8   </body>  
9 </html>
```

Titre de ma page Web

Fichier | file:///C:/Users/oumay/Desktop/index2.html

HTML est un langage de description de pages Web et non pas un langage de programmation

Chapitre 1: Le langage HTML

1. Le code suivant est-il correct?

```
<b><i><u>Bonjour à tous</u></b></i>
```

2. Si on appliquait la balise `<u>` à un texte dans la balise `<title>`, est ce que celui-ci serait souligné?

3. Par défaut, combien mesure la taille de caractères d'un titre `<h1>`?

- ☐ 24pt
- ☐ 36pt
- ☐ 18pt
- ☐ 20pt

4. Que fait le code suivant?

```
<font color="#0000FF" face="arial" size="2">Texte formaté</font>
```

- ☐ Affiche le texte "Texte formaté" en vert avec une taille de caractères de 2 points
- ☐ Affiche le texte "Texte formaté" en vert avec une taille de caractères de 10 points
- ☐ Affiche le texte "Texte formaté" en bleu avec une taille de caractères de 2 points
- ☐ Affiche le texte "Texte formaté" en bleu avec une taille de caractères de 10 points

5. Quelle est l'erreur commise dans ce code?

```
<html>
  <head>
  </head>
  <body>
    <b>Texte en gras</b><br>
    <u>Texte souligné</u></br>
  </body>
</html>
```

- ☐ La balise `<title>` a été oubliée
- ☐ La balise `<body>` ne contient pas assez de contenu
- ☐ La balise `
` ne doit pas être fermée

Chapitre 1: Le langage HTML

Les paragraphes

Nous avons vu l'intérêt de la balise `
` qui sert à sauter la ligne à n'importe quel moment au milieu d'un contenu HTML. Déjà avec cette balise on peut dire qu'on arrive à séparer les paragraphes entre elles. Mais il existe une balise plus adaptée à cet usage. Il s'agit de `<p>`.

Balise `<p>`

- La balise `<p>` est une balise de type **block**, c'est à dire qu'elle crée un bloc et engendre automatiquement un retour à la ligne. Elle sert à définir un paragraphe. On peut la doter de l'attribut **align** qui permet d'aligner, à sa guise, le contenu du paragraphe. Les différentes valeurs de l'attribut **align** sont:

left: C'est la valeur par défaut. Elle permet d'aligner le contenu du paragraphe à gauche de la page (ou à gauche du conteneur qui renferme la balise `<p>`).

• **right:** Elle permet d'aligner le contenu du paragraphe à droite de la page (ou à droite du conteneur qui renferme la balise).

• **center:** Elle permet de centrer le paragraphe.

• **justify:** Elle permet de justifier le contenu du paragraphe (prolonge le texte pour qu'il occupe toute la ligne).

Chapitre 1: Le langage HTML

Les paragraphes

Balise <p>

```
<h1>Les paragraphes</h1>
<p align="left">
  La balise <p> permet de définir un paragraphe au sein d'un code HTML.
  Il engendre un retour à la ligne avant et après.
</p>
```

L'exécution du code sur le navigateur donne:

Les paragraphes

La balise <p> permet de définir un paragraphe au sein d'un code HTML. Il engendre un retour à la ligne avant et après.

Chapitre 1: Le langage HTML

Les conteneurs (ou containers)

On entend par **conteneur** (ou container) des balises qui peuvent renfermer d'autres éléments comme du textes ou des images. Les conteneurs les plus célèbres sont les balises **<div>** et ****.

Balise <div>

La balise **<div>** est une balise de type **block**. Elle permet de définir un conteneur ou un bloc qui contient d'autres éléments. Il sert généralement à mieux diviser la page Web pour placer le bon contenu au bon endroit. Cependant, la balise **<div>** nécessite des styles CSS pour qu'elle soit dotée de toute sa force.

En HTML cette balise ne sert pas à grand chose si ce n'était que pour aligner du texte ou définir un paragraphe (Elle se comporte à peu près comme la balise **<p>**). Les attributs dont on peut la doter sont les même que ceux de la balise **<p>**.

Exemple:

```
<h1>Les conteneurs</h1>
<div align="left">
  La balise &lt;div&gt; permet de définir un conteneur de type block au sein d'un code HTML.
  Il engendre un retour à la ligne avant et après.
</div>
```

L'exécution du code sur le navigateur donne:

Les conteneurs DIV

La balise **<div>** permet de définir un conteneur de type block au sein d'un code HTML. Il engendre un retour à la ligne avant et après.

Chapitre 1: Le langage HTML

Les conteneurs (ou containers)

Balise ``

La balise `` est aussi un container, mais de type **inline** (elle occupe juste assez d'espace pour afficher ce qu'elle contient et pas toute la ligne comme la balise `<div>`). Par conséquent elle n'engendre pas de retour à la ligne.

Exemple:

```
HTML est un <span>langage</span> de description de pages Web.
```

Sur le navigateur cela donne:

HTML est un langage de description de pages Web.

Chapitre 1: Le langage HTML

Les listes HTML

Pour afficher plusieurs entrées, l'une en dessous de l'autre, il n'y a pas mieux que les listes. Il y'en a deux types: des listes ordonnées et des listes non ordonnées.

Listes ordonnées: Balise ``

La balise `` permet d'avoir des listes ordonnées (ou numérotées). Elle doit être fermée (par ``) et elle contient une autre balise qui est ``. La balise `` permet de définir une entrée de la liste (Il y 'a donc autant de balises `` que d'entrées).

La balise `` peut avoir des attributs dont les plus importants sont:

- **type**: sert à définir le type de marqueur de la liste. Les différentes valeur de l'attribut sont:

- 1 pour les chiffres classiques (il s'agit d'ailleurs de la valeur par défaut),

- A pour l'alphabet majuscule,

- a pour l'alphabet minuscule,

- I pour les chiffres romains majuscules

- i pour les chiffres romains minuscule.

- **start**: indique la valeur du marqueur du début. Par défaut la liste commence par 1 (si elle est ordonnée). Si vous mettez `start="5"` alors la première entrée sera marquée par 5 et non par 1, la deuxième par 6...

Chapitre 1: Le langage HTML

Exemple:

```
Les noms des balises de bases à déclarer dans un document HTML sont:  
<ol type="1">  
  <li>HTML</li>  
  <li>HEAD</li>  
  <li>BODY</li>  
</ol>
```

On obtient sur le navigateur:

Les noms des balises de bases à déclarer dans un document HTML sont:

1. HTML
2. HEAD
3. BODY

Chapitre 1: Le langage HTML

Liste non ordonnées: Balise

• La balise **** permet de créer des listes non ordonnées. Elle contient aussi la balise **** qui permet de définir les éléments de la liste.

Tout comme la balise ****, la balise **** possède des attributs qui permettent de personnaliser la liste. Le plus important d'entre eux est l'attribut **type** qui définit le type de marqueur et qui peut avoir une des valeurs suivantes:

- **disc**: le marqueur est un cercle plein. Il s'agit de la valeur par défaut.
- **circle**: le marqueur est un cercle creux.
- **square**: le marqueur est un carré plein.

Le résultat obtenu ressemble à ceci:

On reprend l'exemple précédent avec la balise **** cette fois:

```
Les noms des balises de bases à déclarer dans un document HTML sont:  
<ul type="square">  
  <li>HTML</li>  
  <li>HEAD</li>  
  <li>BODY</li>  
</ul>
```

Les noms des balises de bases à déclarer dans un document HTML sont:

- HTML
- HEAD
- BODY

Chapitre 1: Le langage HTML

Les images

Pour afficher une image en HTML on fait appelle à la balise ``. Toutefois on peut afficher les images en tant qu'arrière plan (CSS) .

Balise ``

- La balise `` permet d'afficher une image sur le navigateur. Il s'agit d'une balise orpheline (donc pas de ``).

La balise `` peut avoir plusieurs attributs. Les plus importants sont:

- **src**: désigne la source de l'image. La valeur associée correspond au chemin vers elle (ou son emplacement). Le chemin peut être absolu ou relatif. Le chemin absolu représente l'[URL](#) de l'image (par exemple: `http://www.logistiqua.com/wp-content/uploads/2016/02/logo-EST-FES.png`). Le chemin relatif, quant à lui, désigne l'emplacement de l'image à partir de l'emplacement actuel (par exemple: `images/Logo_ESTF.png`).
- **border**: accepte une valeur numérique (1,2...). Cette valeur désigne l'épaisseur de la bordure qui encadre l'image. L'unité utilisée pour la bordure est le pixel (px) mais elle est implicite, seule la valeur numérique est déclarée. Par défaut la bordure est de couleur noire.

Chapitre 1: Le langage HTML

Les images

- **width** et **height**: signifient, respectivement, la largeur et la hauteur de l'image. L'unité peut être le pixel (qui est implicite) ou le pourcentage (qu'il faut expliciter à l'aide du symbole universel %). Si par exemple on donne à la largeur la valeur 100%, cela veut dire que l'image occupe toute la largeur de la page (ou du conteneur qui accueille la balise ****).

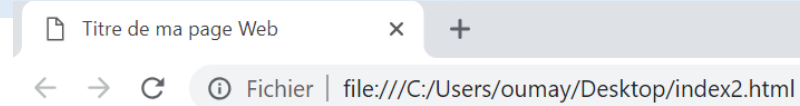
Si width ou height ne sont pas spécifiée alors l'image sera affichée en taille réelle.

Si seule la valeur d'une dimension (width ou height) est déclarée, l'autre dimension sera automatiquement recalculée par le navigateur de telle sorte à conserver les proportions originales de l'image.

- **alt**: représente le texte alternatif à l'image. Il accepte comme valeur un mot ou une phrase qui sera affichée à la place de l'image si celle-ci n'est pas chargée correctement.

- **title**: il s'agit d'un message qui sera affiché dans une infobulle si on survole l'image avec la souris.

Chapitre 1: Le langage HTML



Ecole Supérieure de Technologie de FES



```
index.html x index2.html x
1 <html>
2   <head>
3     <title>Titre de ma page Web</title>
4   </head>
5   <body >
6
7     <b> Ecole Supérieure de Technologie de FES</b>
8
9     <br>
10    <br>
11    <br>
12    <center>
13      
15    </center>
16  </body>
17 </html>
18
```

Chapitre 1: Le langage HTML

Les liens hypertextes

Un lien hypertexte ou hyperlien est un objet HTML (texte, image ou autre) sur lequel on peut cliquer pour aller vers une autre page (ou un autre emplacement dans la même page). Il est considéré comme l'un des éléments les plus importants du langage HTML puisqu'il permet de relier les pages les unes aux autres pour constituer ainsi le **Web** tel que nous le connaissons.

Balise <a>

- La balise <a> (pour **anchor** ou ancre) permet de créer un lien hypertexte pointant vers un autre document ou autre emplacement. Un lien classique est reconnaissable par sa couleur bleue et un style souligné (ces propriétés peuvent être changées à l'aide du langage CSS).

La balise <a> dispose des attributs suivants:

- **href**: désigne le chemin vers la page de destination après avoir cliqué sur le lien. Comme pour l'attribut **src** de la balise , le chemin peut être absolu ou relatif.

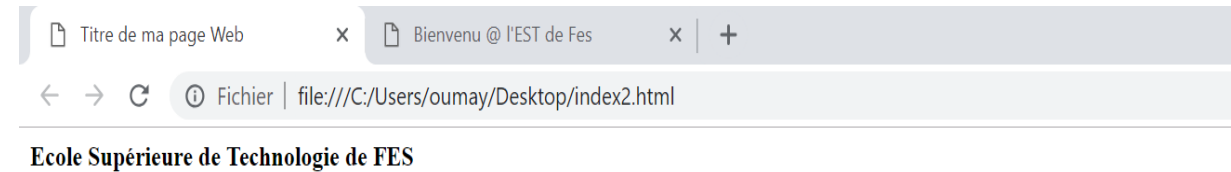
Si par exemple la valeur de l'attribut **href** vaut "mailto:adresse@mail" alors, en cliquant sur le lien, le logiciel de messagerie par défaut installé chez le client s'ouvrira en l'invitant à écrire un mail à "adresse@mail".

- **target**: désigne où la page de destination sera ouverte. Il peut contenir plusieurs valeurs mais une seule est fréquemment

Chapitre 1: Le langage HTML

Les liens hypertextes

- **target:** désigne où la page de destination sera ouverte. Il peut contenir plusieurs valeurs mais une seule est fréquemment utilisée il s'agit de **_blank** (n'oubliez pas le souligné avant le mot blank). La valeur **_blank** signifie que le lien sera ouvert dans une nouvelle fenêtre (ou nouvel onglet). C'est utile pour les liens pointant vers d'autres sites Web. Si l'attribut **target** n'est pas spécifié, alors la page de destination sera ouverte dans la même fenêtre.
- **title:** décrit un message qui sera affiché dans une infobulle si on survole le lien avec la souris.



```
<a href="http://www.est-usmba.ac.ma/" title="Aller vers ESTF" target="_blank">
Lien vers le site de l'ESTF
</a>
```

[Lien vers le site de l'ESTF](http://www.est-usmba.ac.ma/)



Chapitre 1: Le langage HTML

Liens internes

Il arrive des fois qu'en cliquant sur un lien, on se déplace vers un autre emplacement dans la même page déjà ouverte. On l'appelle alors un lien interne.

Le principe est simple. Supposons que vous voulez prévoir un lien qui déplace l'internaute vers une image dans la même page. Ce qu'il faut faire c'est ajouter une **ancree** près de l'image (de préférence avant la balise ``). L'ancree aura la forme suivante:

```
<a name="toto"></a>
```

(Vous être libre de mettre ce que vous voulez dans l'attribut **name**). Notez que cet ancre ne sera pas visible car la balise `<a>` ne contient aucun texte. Ensuite on va créer le lien qui nous mènera vers l'ancree créé comme ceci:

```
<a href="#toto">Aller vers toto</a>
```

Notez que le `#` fait référence à la page courante. Après le `#` on déclare le nom de l'ancree (la valeur de l'attribut **name** de tout à l'heure).

En cliquant sur le texte "Aller vers toto" on sera déplacé automatiquement vers l'ancree et par conséquent vers l'image souhaitée.

Chapitre 1: Le langage HTML

1. Une page Web peut contenir un nombre illimité d'images

- ☐ Oui
- ☐ Non

2. Si on applique la balise `<i>` à une image, cette dernière sera inclinée

- ☐ Oui
- ☐ Non

3. La balise `` permet d'intégrer des images JPEG et PNG seulement

- ☐ Oui
- ☐ Non

4. Bien que l'on peut redimensionner les images avec les attributs **width** et **height**, il est peu recommander de recourir à cette pratique.

- ☐ Oui
- ☐ Non

Chapitre 1: Le langage HTML

Les tableaux HTML

Nous avons déjà vu deux balises qui servent de conteneur à savoir **<div>** et ****. C'est à dire qu'ils peuvent contenir d'autres balises afin de donner une structure au document. Mais j'ai précisé que ces deux balises ne servent pas à grand chose si nous ne les dotons pas de styles CSS. Cependant, il existe un autre objet HTML qui peut servir de conteneur sans être obligé de faire appel au CSS. C'est le tableau HTML.

Les tableaux peuvent servir de grille qui contiennent des données structurées en ligne et en colonne. Comme par exemple une liste d'articles avec leurs prix. Mais ils peuvent servir également de conteneurs qui renferment d'autres objet HTML afin de structurer sa page Web sous forme de cellules et faciliter ainsi la mise en page.

Pour créer un tableau nous avons besoin, au moins, de trois balises de base: **<table>**, **<tr>** et **<td>**.

Balise **<table>**

- La balise **<table>** est le conteneur principal. Elle permet de déclarer un tableau et peut accueillir de nombreux attributs dont les plus fréquents sont:
 - **border**: définit l'épaisseur de la bordure du tableau (et de ses cellules). Elle est exprimée en pixels (px) mais nous nous contentons de déclarer la valeur sans l'unité. Celle ci étant implicite et connue par le navigateur.
 - **width**: définit la largeur du tableau. Elle est exprimée en pixel ou en pourcentage. Si on entend les pixels alors on met la valeur sans l'unité. Si c'est du pourcentage alors on met la valeur suivie du symbole %.

Chapitre 1: Le langage HTML

Les tableaux HTML

Balise <table>

- **height**: définit la hauteur du tableau en pixel. Mais il vaut mieux ne pas déclarer cet attribut et laisser le tableau s'étirer en fonction de la taille de son contenu.
- **bgcolor**: désigne la couleur de l'arrière plan du tableau. Cette couleur est exprimée en nom anglais ou en code de couleur.
- **cellspacing**: signifie l'espacement entre les différentes cellules du tableau. Sa valeur est exprimée en pixel (implicite).
- **cellpadding**: définit la marge interne des cellules. Elle est exprimée en pixel (implicite).

Balise <tr>

Même si vous déclarez la balise <table> dans le code, rien ne s'affichera sur le navigateur.

En effet, ce qui peut être affiché ce sont les cellules d'un tableau, et une cellule c'est au moins une ligne et une colonne.

La balise <tr> permet d'ajouter une ligne à un tableau. Elle est directement déclarée dans la balise <table>.

Il y a donc autant de balises <tr> que de lignes.

La balise <tr> peut accueillir des attributs comme **height** ou **bgcolor**, mais la plupart des développeurs préfèrent la déclarer sans attribut.

Chapitre 1: Le langage HTML

Les tableaux HTML

Balise `<td>`

• La balise `<td>` représente une colonne. C'est elle en fait le vrai conteneur, car c'est elle qui peut accueillir du contenu. Elle est déclarée dans la balise `<tr>`. Si vous voulez avoir un tableau avec une ligne et deux colonnes, alors vous devez déclarer une balise `<tr>` et dans celle ci, deux balises `<td>`.

Si je vous ai conseillé de laisser la balise `<tr>` sans attributs, c'est parce que c'est la balise `<td>` qui doit en avoir. Les principaux attributs qu'on peut déclarer sur la balise `<td>` sont:

- **width**: définit la largeur de la colonne. Elle est exprimé en pixel ou en pourcentage par rapport à la largeur du tableau.
- **height**: définit la hauteur de la colonne en pixel.
- **bgcolor**: désigne la couleur de l'arrière plan de la colonne. Cette couleur est exprimée en nom anglais ou en code de couleur. Si la couleur d'arrière plan n'est pas déclarée, alors la colonne est transparente, et c'est la couleur de l'arrière plan du tableau qui sera visible.
- **align**: permet de spécifier l'alignement des objets contenus dans la colonne. Elle peut avoir une des valeurs: **left** (valeur par défaut), **right**, **center** ou **justify**.
- **valign**: définit l'alignement vertical des objets contenus dans la colonne. Elle peut avoir une des valeurs: **middle** (valeur par défaut), **top** ou **bottom**.

Chapitre 1: Le langage HTML

Les tableaux HTML

```
<html>
  <head>
    <title>Titre de ma page Web</title>
  </head>
  <body > |
<table border="1" width="100%" cellspacing="0" cellpadding="6">
  <tr>
    <td width="50%" bgcolor="#000000"><font color="#FFFFFF">Type de matériel</font></td>
    <td width="50%" bgcolor="#000000"><font color="#FFFFFF">Quantité</font></td>
  </tr>
  <tr>
    <td width="50%">PC de bureau</td>
    <td width="50%">26</td>
  </tr>
  <tr>
    <td width="50%">PC portables</td>
    <td width="50%">32</td>
  </tr>
  <tr>
    <td width="50%">Imprimantes</td>
    <td width="50%">12</td>
  </tr>
</table>
</body>
</html>
```

← → ↺ ⓘ Fichier | file:///C:/Users/oumay/Desktop/index2.html

Type de matériel	Quantité
PC de bureau	26
PC portables	32
Imprimantes	12

Chapitre 1: Le langage HTML

Les tableaux HTML

Balise `<caption>`

La balise `<caption>` est déclarée directement après la balise `<table>`. Elle contient le titre du tableau et elle peut accueillir l'attribut **align** qui peut prendre une des valeurs suivante: **top** (valeur par défaut) permet d'avoir le titre en dessus du tableau et **bottom** permet d'avoir le titre en dessous du tableau.

Balise `<thead>`

La balise `<thead>` permet de déclarer les cellules d'entête. Elle est déclarée dans la balise `<table>` et peut contenir des lignes (balise `<tr>`) qui contiennent des colonnes (balise `<td>`). Les cellules d'entête sont incluses en haut.

Balise `<th>`

Quand la balise `<thead>` est déclarée, il est préférable de remplacer ses balises `<td>` par `<th>`. La balise `<th>` désigne donc une légende pour les colonnes d'entête.

Balise `<tbody>`

Il s'agit d'une balise qui est sensée accueillir les cellules du corps du tableau. `<tbody>` est déclarée dans la balise `<table>` est peut contenir les balises `<tr>` puis `<td>`.

Balise `<tfoot>`

Il s'agit d'une balise qui est sensée accueillir les cellules du pied du tableau. `<tfoot>` est déclarée dans la balise `<table>` après la balise `<thead>` est peut contenir les balises `<tr>` puis `<td>`. Les cellules de la balises `<tfoot>` sont automatiquement intégrées à la fin du tableau.

Chapitre 1: Le langage HTML

```
<html>
  <head>
    <title>Titre de ma page Web</title>
  </head>
  <body>
    <table border="1" width="100%" cellspacing="0" cellpadding="6">
      <caption>Liste du matériel informatique disponible</caption>
      <thead>
        <tr>
          <th width="50%">Type de matériel</th>
          <th width="50%">Quantité</th>
        </tr>
      </thead>
      <tfoot>
        <tr>
          <th width="50%">Total</th>
          <th width="50%">70</th>
        </tr>
      </tfoot>
      <tbody>
        <tr>
          <td width="50%">PC de bureau</td>
          <td width="50%">26</td>
        </tr>
        <tr>
          <td width="50%">PC portables</td>
          <td width="50%">32</td>
        </tr>
        <tr>
          <td width="50%">Imprimantes</td>
          <td width="50%">12</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

← → ↻ ⓘ Fichier | file:///C:/Users/oumay/Desktop/index2.html



Liste du matériel informatique disponible

Type de matériel	Quantité
PC de bureau	26
PC portables	32
Imprimantes	12
Total	70

Chapitre 1: Le langage HTML

Fusionnement de cellules d'un tableau

Si on suppose qu'on veut créer un tableau qui contient deux lignes, et si la ligne du dessus contient deux colonnes, alors ce qu'on a fait jusqu'ici ne nous permet pas d'avoir un nombre différent de colonnes dans la ligne du dessous. Alors cette dernière ne peut comporter que deux colonnes si on tient à avoir un affichage correcte. Néanmoins, il existe des cas (d'ailleurs très nombreux) où on a besoin d'avoir un nombre de cellules différent dans chaque ligne. La solution magique est le **fusionnement des cellules**.

On peut fusionner les cellules horizontalement ou verticalement. Mais le plus fréquent c'est de fusionner à l'horizontale grâce à l'attribut **colspan**.

L'attribut colspan

L'attribut **colspan** est inclus dans la balise de colonne **<td>**. Il indique à celle ci combien de colonnes il faut fusionner pour en avoir qu'une seule à la fin.

Un exemple sera en mesure de clarifier les idées. Alors, supposons que je veux créer un tableau qui contient deux lignes et chaque ligne contient deux colonnes. Bien entendu, y a rien de nouveau à cela. Voici le code:

Chapitre 1: Le langage HTML

```
<table>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

Cellule 1	Cellule 2
Cellule 3	Cellule 4

```
<table>
  <tr>
    <td colspan="2">Cellule 1</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

Cellule 1	
Cellule 3	Cellule 4

Chapitre 1: Le langage HTML

L'attribut rowspan

Si **colspan** fusionne les colonnes à l'horizontale, **rowspan** quant à elle, elle les fusionne à la verticale. Le principe est le même: Enlevez la (les) colonnes en plus, et sur celle qui reste mettez **rowspan="le nombre de colonnes à fusionner"**. Par exemple. avant le fusionnement:

```
<table>
  <tr>
    <td rowspan="2">Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 4</td>
  </tr>
</table>
```

Cellule 1	Cellule 2
	Cellule 4

```
<table>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

Cellule 1	Cellule 2
Cellule 3	Cellule 4

Chapitre 1: Le langage HTML

1. Que fait le code suivant?

```
<table>
  <tr>
    <td colspan="2"></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
  </tr>
</table>
```

- ☐ Crée un tableau à deux lignes et deux colonnes
- ☐ Crée un tableau à deux lignes, la première contient une seule colonne et la deuxième en contient deux
- ☐ Crée un tableau à deux lignes, la première contient deux colonnes et la deuxième contient une seule
- ☐ Crée un tableau à trois lignes.

2. Le code suivant est-il correct?

```
<table>
  <tr>
    <font color="red">Mon tableau</font>
    <td></td>
  </tr>
</table>
```

3. Un tableau doit impérativement contenir autant de colonnes dans chaque ligne.

4. Par défaut, le contenu de la balise <caption> est placé

- ☐ En dessus du tableau
- ☐ En dessous du tableau

- ☐ Oui
- ☐ Non

Chapitre 1: Le langage HTML

Les formulaires

Qu'est ce qu'un formulaire?

Un **formulaire HTML** permet de rendre la page Web plus interactive en la rendant capable de dialoguer avec l'internaute à travers des champs de saisie et boutons. Un formulaire permet à l'internaute de saisir du texte ou de valider un choix ou encore de sélectionner une entrée. Ces informations seront ensuite (dans la plupart des cas) envoyées au serveur pour les traiter. C'est très utile surtout quand il s'agit d'une [page Web dynamique](#).

Balise <form>

La balise **<form>** délimite les champs du formulaire (zones de texte, boutons...). Elle peut avoir plusieurs attributs dont voici les plus fréquents:

L'attribut name

L'attribut **name** permet d'identifier le formulaire par un nom. On peut mettre n'importe quel nom (à conditions qu'il ne contienne que les caractères conventionnels comme les lettres et les chiffres). Le nom du formulaire est unique. En effet, nous pouvons déclarer plusieurs formulaires dans la même page, pour les distinguer les uns des autres il suffit de leur donner des noms différents.

Chapitre 1: Le langage HTML

L'attribut **method**

L'attribut **method** permet de spécifier la méthode à utiliser pour envoyer le formulaire au serveur. En effet, l'objectif d'un formulaire c'est de transférer les données saisies par le client au serveur (on dit généralement **poster** ou **soumettre** le formulaire). En HTML, il existe deux méthodes pour envoyer un formulaire: **GET** et **POST**.

La méthode **GET**

La méthode **GET** (method="get") est la valeur par défaut si on ne précise aucune valeur pour l'attribut **method**. Cette méthode permet d'envoyer les données du formulaire à travers l'URL (Uniform Resource Locator) en utilisant les symboles **?** pour préciser la suite de données et **&** pour séparer les données entre elles.

Pour mieux comprendre, imaginons que sur un site il existe une page nommée **infos.html** qui renferme un formulaire utilisant la méthode **GET** et qui contient deux zones de textes nommées respectivement **ville** et **pays**. Au chargement du formulaire, le client a saisi dans les deux champs les valeurs **Fès** et **Maroc**. Quand le client ordonnera l'envoi du formulaire, les données saisies seront envoyées via l'URL et celui ci aura la forme suivante:

`http://www.site.ma/infos.html?ville=Fès&pays=Maroc`

Chapitre 1: Le langage HTML

L'attribut method

La méthode **POST** (method="post") permet d'envoyer les données du formulaire au serveur à travers une entête. Pour simplifier, on va dire que les données et la page ne font qu'un. Les informations ne seront donc pas visibles sur l'URL. La méthode **POST** permet aussi de poster un volume de données plus important que celui transmis par **GET**. D'ailleurs on peut même poster un fichier à travers un formulaire via cette méthode. C'est ce que l'on appelle le "Upload" (C'est comme si vous "uploadez" une vidéo sur Youtube, ou votre photo de profil sur Facebook).

L'attribut action

Quand un formulaire est rempli par l'internaute, il est posté au serveur pour le traitement. Il faut donc désigner la page qui se chargera du traitement des données. Cette page est spécifiée dans l'attribut **action**.

Si l'attribut **action** est vide ou n'est pas mentionné, alors c'est la page courante (celle qui contient le formulaire) qui sera la page **action**.

Chapitre 1: Le langage HTML

Les formulaires

Supposons que nous voulons créer un formulaire nommée "inscription" utilisant la méthode "post" et faisant appel à la page "inscription.html" pour assurer le traitement après envoi. Le code HTML ressemblera à ceci:

```
<form name="inscription" method="post" action="inscription.html">  
</form>
```

Si vous exécutez ce code sur le navigateur, rien ne s'affichera. En effet, ce qui sera visible c'est ce que l'on intégrera dans la balise **<form>**.

Que contient donc la balise **<form>**? Elle contient ce que l'on appelle **les champs de formulaire**. Ce sont des objets comme des zones de texte ou des boutons. Nous allons maintenant voir comment intégrer ces champs.

Chapitre 1: Le langage HTML

Les formulaires

Balise `<input>`

La **zone de texte** est le champ de formulaire le plus célèbre. Il permet d'écrire un texte sur une seule ligne (comme le nom ou le login par exemple). Pour créer une zone de texte on fait appel à la balise `<input>`. Il s'agit d'une balise orpheline (pas de `</input>`).

Notez que la plupart des champs de formulaire existants sont déclarés aussi à l'aide de la même balise `<input>`. Pour distinguer une zone de texte d'un bouton par exemple, on fait appel à l'attribut **type**. Détaillons les attributs les plus utilisées pour les **inputs**.

Attribut **type**

L'attribut **type** permet de définir le type de champ à intégrer. Voici les valeurs à mettre pour cet attribut et qui permettent d'avoir les champs les plus fréquents:

- **text**: Permet d'avoir une zone de texte normale (adapté pour les noms, emails, login...)
- **password**: Permet d'avoir une zone de mot de passe où les caractères sont masqués.

Chapitre 1: Le langage HTML

- **radio**: Pour créer un bouton radio. Il s'agit d'un petit cercle qu'on peut cocher et qui permet de faire un choix unique. Le fait de cocher un autre bouton radio décoche le premier.
- **checkbox**: Permet de créer une case à cocher. C'est un petit carré qu'on peut cocher ou décocher. Il est destiné à faire des choix multiples.
- **file**: Permet de créer le champ de chargement de fichier (comme une pièce jointe d'un email).
- **hidden**: Pour créer un champ caché. C'est comme une zone de texte mais qui n'est pas visible sur le navigateur. Il sert à stocker provisoirement des données jusqu'à la phase de traitement.
- **button**: Permet de créer un bouton simple. A la base, ce bouton ne fait aucune opération. Il faut donc le programmer (généralement avec Javascript).
- **reset**: Permet de créer un bouton d'annulation. Si le formulaire contient des champs qu'on a déjà saisi, ce bouton permet de tout initialiser.
- **submit**: C'est le bouton d'envoi. C'est lui qui permet de poster le formulaire vers la page définie dans l'attribut **action** de la balise **<form>**.

Chapitre 1: Le langage HTML

Attribut name

L'attribut **name** permet de donner un nom au champ de formulaire. Le nom sert d'identifiant du champ et doit être unique.

Attribut value

L'attribut **value** sert à définir la valeur par défaut d'un champ. S'il s'agit d'une zone de texte, le fait de déclarer l'attribut **value** force le navigateur à initialiser ce champ avec le texte faisant office de valeur de l'attribut. Si le champs est un bouton, alors le texte de l'attribut **value** constitue l'étiquette (ou label). C'est le texte visible sur le bouton. Pour les boutons radio ou les cases à cocher, le texte de l'attribut **value** n'est pas visible sur le navigateur mais il est très utile (si on fait appel à des programmes comme Javascript ou PHP).

Attribut size

L'attribut **size** contient un entier comme valeur. Il définit la largeur du champ en caractères. Par défaut, une zone de texte mesure 20 caractères (cela veut dire qu'on peut voir jusqu'à 20 caractères à la fois même si le texte est plus grand).

Chapitre 1: Le langage HTML

Attribut **tabindex**

L'attribut **tabindex** permet de définir l'ordre de tabulation. En pratique, si on veut saisir un grand formulaire, on se déplace d'un champ à un autre en appuyant sur la touche "tabulation" du clavier. Par défaut le curseur se déplace dans l'ordre où sont déclarés les champs. On peut rompre cet ordre en définissant les valeurs de **tabindex**. La valeur est un entier qui commence de 1. Le curseur se déplacera dans l'ordre croissant de **tabindex**.

Attribut **checked**

L'attribut **checked** est un attribut booléen. Explicitement il est déclaré comme ceci: **checked="true"**, mais il suffit de déclarer **checked** tout seul pour que le navigateur sache qu'il est activé. Cet attribut s'applique uniquement sur les boutons radio (**type="radio"**) et les cases à cocher (**type="checkbox"**). S'il est déclaré alors le bouton est automatiquement coché au chargement de la page.

Chapitre 1: Le langage HTML

Balise <textarea>

La balise <code>textarea</code> permet de définir un espace de texte. Un espace de texte est une grande zone de texte qui permet d'écrire des paragraphes entiers.

Les attributs **name** et **tabindex** sont aussi applicables sur l'espace de texte mais, en plus, il dispose des attributs suivants:

- **cols**: définit la largeur en caractères de l'espace de texte. Elle désigne combien de caractère une ligne peut-elle contenir.

Le fait de dépasser la largeur définie entraîne un retour automatique à la ligne.

- **rows**: définit la hauteur de l'espace de texte. La hauteur désigne combien de ligne on peut voir à la fois. Si le texte dépasse le nombre de lignes défini, alors une barre de défilement qui permet d'atteindre le reste des lignes apparaît.

Chapitre 1: Le langage HTML

Les formulaires

```
<form name="inscription" method="post" action="">
  Civilité<br>
  <input type="radio" name="civilite"> Mlle<br>
  <input type="radio" name="civilite"> Mme<br>
  <input type="radio" name="civilite"> M.<br>
  Nom et prénom<br>
  <input type="text" name="nom" value=""><br>
  Email<br>
  <input type="text" name="email" value=""><br>
  Login<br>
  <input type="text" name="login" value=""><br>
  Mot de passe<br>
  <input type="password" name="pass" value=""><br>
  Vous être<br>
  <select name="niv">
    <option>Etudiant</option>
    <option>Fonctionnaire</option>
    <option>Employé au secteur privé</option>
  </select><br>
  <input type="button" name="envoyer" value="S'inscrire">
</form>
```

Civilité

- ☐ Mlle
☐ Mme
☐ M.

Nom et prénom

Email

Login

Mot de passe

Vous être

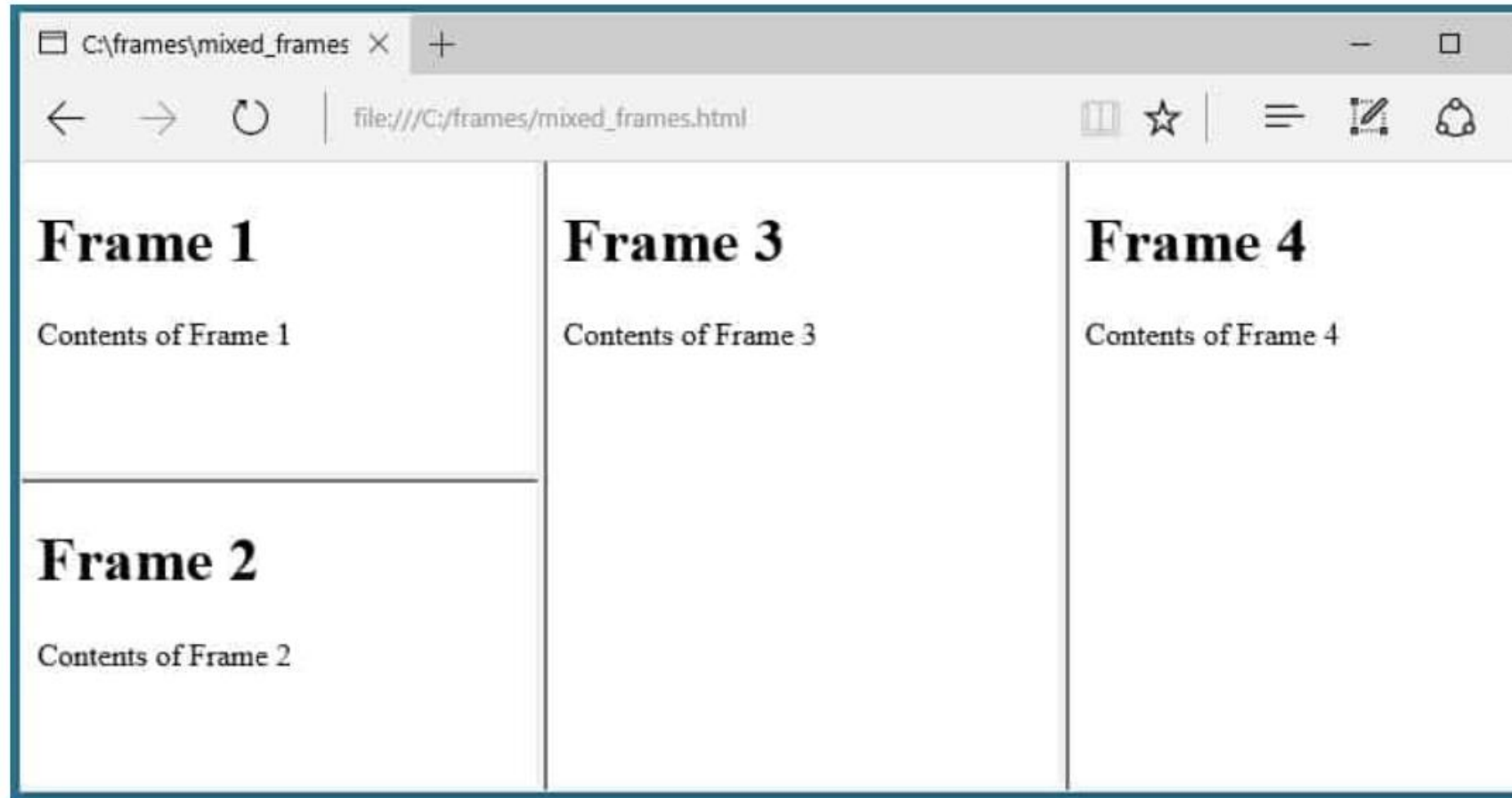
Etudiant ▼

S'inscrire

Chapitre 1: Le langage HTML

Les framesets

```
<frameset cols="*,*,*">
  <frameset rows="*,*">
    <frame src="frame_1.html">
    <frame src="frame_2.html">
  </frameset>
  <frame src="frame_3.html">
  <frame src="frame_4.html">
</frameset>
```



Chapitre 1: Le langage



```
<frameset rows="150px,*">
  <frame noresize src="frame_1.html" marginheight="15">
  <frameset cols="20%,*,20%">
    <frame src="frame_2.html" frameborder="0">
    <frame src="frame_3.html" name="mid_col" frameborder="0">
    <frame src="frame_4.html" frameborder="0">
  </frameset>
</frameset>
```


Chapitre 1: Le langage HTML

```
<!DOCTYPE html>
<html>
<body>
  <h1>Frame 2</h1>
  <p>Contents of Frame 2</p>
  <ul>
    <li><a href="frame_1.html" target="mid_col">Load frame_1.html</a></li>
    <li><a href="frame_2.html" target="mid_col">Load frame_2.html</a></li>
    <li><a href="frame_3.html" target="mid_col">Load frame_3.html</a></li>
    <li><a href="frame_4.html" target="mid_col">Load frame_4.html</a></li>
  </ul>
</body>
</html>
```