



Université Hassan 1^{er}



Faculté des Sciences et Techniques de Settat

COURS HTML

HyperText Markup Language

Professeur Laachfoubi Nabil

Département des Mathématiques et Informatique

Sommaire

| | |
|--|----|
| A. Introduction..... | 4 |
| A.1. Bref historique du HTML | 4 |
| A.2. HTML : un langage de balises | 4 |
| A.3. HTML : règles de base | 5 |
| B. Structure d'un document HTML..... | 6 |
| B.1. Les éléments de base | 6 |
| C. Structurer le contenu d'une page | 9 |
| C.1. Les grandes divisions de la page..... | 9 |
| C.2. Les divisions secondaires..... | 11 |
| C.3. Les divisions de bloc locales | 12 |
| C.4. Les éléments des divisions sémantiques du texte | 12 |
| C.5. Les styles physiques..... | 14 |
| C.6. Les listes..... | 15 |
| D. Insérer des images et du multimédia..... | 17 |
| D.1. Les types d'images | 17 |
| D.2. L'insertion d'images et du multimédia | 18 |
| E. Les liens..... | 19 |
| E.1. L'insertion d'un lien | 19 |
| E.2. Les liens externes..... | 20 |
| E.3. Les liens ciblés : les ancres..... | 20 |
| E.4. Les liens pour télécharger des fichiers | 21 |
| E.5. Lien à partir d'une image ou un élément de type bloc..... | 22 |
| E.6. Le lien vers une adresse email | 22 |
| E.7. Les liens déclenchant un script JavaScript..... | 22 |
| F. Les tableaux | 23 |
| F.1. La structure générale d'un tableau..... | 23 |
| F.2. Les cellules de tableaux | 25 |
| F.3. La fusion des cellules | 26 |
| F.4. Les cellules d'entête | 27 |

| | |
|--|----|
| F.5. La légende d'un tableau..... | 27 |
| F.6. Le groupement de colonnes | 28 |
| F.7. La structuration d'un tableau | 29 |
| G. Les formulaires..... | 30 |
| G.1. La déclaration de formulaire | 30 |
| G.2. Les composants communs | 31 |
| G.3. Les composants de saisie d'informations | 32 |

A. Introduction

A.1. Bref historique du HTML

Depuis les débuts du World Wide Web, de nombreuses versions de HTML (HyperText Markup Language) ont été créées :

| Année | Version |
|-------|---|
| 1989 | Tim Berners-Lee invente WWW |
| 1991 | Tim Berners-Lee invente HTML |
| 1993 | Dave Raggett rédige HTML+ |
| 1995 | HTML Working Group définit HTML 2.0 |
| 1997 | Recommandation W3C : HTML 3.2 |
| 1999 | Recommandation W3C : HTML 4.01 |
| 2000 | Recommandation W3C : XHTML 1.0 |
| 2008 | WHATWG HTML5 First Public Draft |
| 2012 | WHATWG HTML5 Living Standard |
| 2014 | Recommandation W3C : HTML5 |
| 2016 | W3C Candidate Recommendation : HTML 5.1 |

A.2. HTML : un langage de balises

Dans un langage de balisage, tout contenu, qu'il s'agisse de texte, d'images ou d'éléments multimédias les plus divers, doit être enfermé dans un élément. En HTML, chaque élément possède un nom déterminé ; la liste des éléments utilisables est limitative et clairement définie dans la spécification du langage.

Les balises

Les balises, appelées **tags** en anglais, sont des commandes à l'intention du navigateur et saisies entre les signes, inférieur à (<) et supérieur à (>). Ainsi une balise s'écrit <balise>.

En règle générale, à toute balise d'ouverture correspond une balise de fermeture qui marque la fin de la commande annoncée par la balise d'ouverture. La balise de fermeture reprend le même énoncé que la balise d'ouverture mais ce dernier est précédé d'un slash (/). Ainsi à la balise d'ouverture <balise> correspondrait la balise de fermeture </balise>. La syntaxe d'une balise est alors :

```
<balise> Contenu Balise </balise>
```

Les attributs de la balise

Il est parfois nécessaire de compléter une commande par des spécifications plus précises dans l'un ou l'autre domaine. Pour ce faire, le langage HTML dispose des attributs de la balise. L'attribut s'insère dans la balise, entre le mot de commande et le signe > final.

La syntaxe complète d'une balise avec un attribut est :

```
<balise attribut="valeur"> Contenu Balise </balise>
```

L'attribut comporte toujours une valeur, celle-ci s'indique en complément de l'attribut par un signe égal (=) suivi de la valeur mise entre des guillemets. Il est impératif de mettre cette valeur entre guillemets. Le strict respect de la syntaxe veut qu'il n'y ait pas d'espace avant et après le signe égal.

Il est possible d'utiliser plusieurs attributs, séparés par un espace, dans une même balise :

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

Le contenu d'une balise peut être constitué de texte ou d'autres balises, eux-mêmes, peuvent en contenir d'autres, et ainsi de suite. Cet ensemble d'inclusion constitue la hiérarchie du document HTML.

A.3. HTML : règles de base

Un document bien formé

Un document HTML est considéré bien formé s'il respecte les règles simples suivantes :

- Les balises et les attribues sont insensibles à la casse. Ainsi, on peut écrire indifféremment <BALISE>, <Balise> ou <balise>.
- Les balises non vides doivent avoir une balise d'ouverture et une balise de fermeture. Par exemple, au lieu d'écrire :

```
<a>  
  <b>article1  
  <b>article2
```

On utilisera le code suivant :

```
<a>  
  <b>article1</b>  
  <b>article2</b>  
</a>
```

- Pour les balises vides, il serait plus propre de les terminer par les caractères /> précédés d'un espace pour marquer la fin de la balise : <balise />.
- Les balises doivent être correctement imbriquées et donc obéir au principe premier ouvert, dernier fermé. Par exemple, au lieu d'écrire (code incorrect) :

```
<a><b><c>contenu</a></c></b>
```

On doit utiliser le code suivant :

```
<a><b><c>contenu</c></b></a>
```

- Toutes les valeurs des attributs doivent toujours figurer entre des guillemets. Par exemple, on préférera le code suivant :

```
<a attribut1="valeur1" attribut2="valeur2" > contenu</a>
```

Plutôt que d'écrire :

```
<a attribut1=valeur1 attribut2=valeur2 > contenu</a>
```

Un document conforme

Un document HTML est déclaré conforme et valide s'il respecte les règles d'inclusion des éléments, les uns dans les autres, telles qu'elles sont définies dans la spécification. Pour s'assurer de la conformité d'un document HTML, plusieurs validateurs sont disponibles en ligne comme :

- <https://validator.w3.org/>
- <https://html5.validator.nu/>

B. Structure d'un document HTML

B.1. Les éléments de base

Un document HTML se compose d'un ensemble d'éléments de balisage, certains sont obligatoires, d'autres sont optionnels. Il comporte toujours au moins quatre éléments :

- Une déclaration DOCTYPE
- Les balises d'ouverture et de fermeture <html> </html>
- Les balises d'ouverture et de fermeture <head> </head>
- Les balises d'ouverture et de fermeture <body> </body>

La déclaration DOCTYPE

Tout document HTML doit commencer par une déclaration doctype. Cette ligne de balisage spécifie la version HTML du document qui va être créé et permet 5, aux navigateurs de l'interpréter correctement. A partir de sa version 5, Le HTML utilise le doctype simplifié suivant :

```
<!DOCTYPE html>
```

La balise racine <html>

La balise <html> est l'élément racine du document. Elle prend place juste après la déclaration du doctype. Il n'existe que deux éléments enfants de la racine <html> : le <head> et le <body> mais elle constitue le parent de tous les autres éléments.

La balise <html> possède plusieurs attributs dont les plus utilisés sont :

1. L'attribut **lang** : la valeur indique la langue utilisée par défaut dans la page.
2. L'attribut **dir** : il indique le sens de lecture du texte dans la page. Il peut prendre les valeurs **ltr** pour le texte qui se lit de gauche à droite (langues occidentales) ou **rtl** pour le texte qui se lit de droite à gauche (langues orientales comme l'arabe ou l'hébreu)

```
<html lang="fr" dir="ltr" >  
  <!-- Les éléments HTML de la page -- >  
</html>
```

L'entête du document : la balise <head>

La balise <head> constitue l'entête du document. Elle contient les balises suivantes :

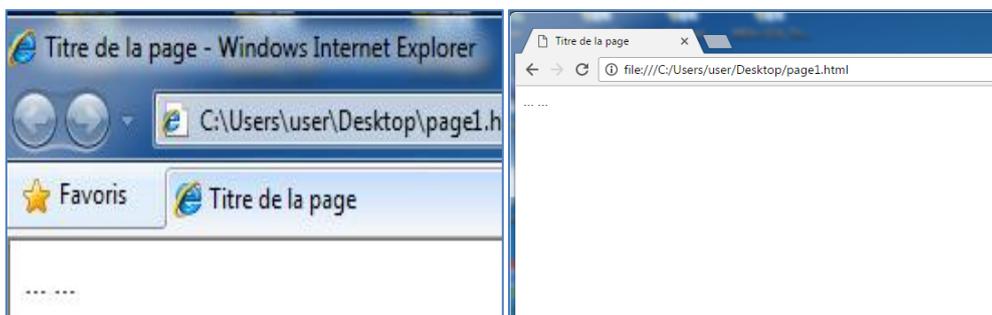
1. La balise <title> : c'est le titre du document,
2. Les balises <style> et <link> : englobent des déclarations ou des appels de feuilles de style CSS,
3. La balise <base> : c'est adresse de base par default de tous les liens de la page,
4. La balise <script> : contient des fonctions ou des liens vers des fichiers JavaScript,
5. Les balises **méta** : servent à placer des métadonnées dans la page HTML. Ce sont des informations, utilisées par les navigateurs web et les moteurs de recherche.

```
<head>  
  <meta charset="utf-8">  
  <title>Titre de la page</title>  
  <link rel="stylesheet" href="style.css">  
  <script src="script.js"></script>  
</head>
```

Le titre de la page : la balise <title>

La balise <title> est la seule balise obligatoire dans l'entête du document. Elle contient le titre du document HTML qui apparaît dans la barre de titre située en haut de la fenêtre du navigateur.

```
<title>Titre de la page</title>
```



L'adresse de base : la balise <base/>

La balise <base/> contient l'adresse de base par default de tous les liens de la page si leur adresse est transmise de façon relative.

```
<base href="http : //www.fsts.ac.ma/pages/" >
```

Les documents liés : la balise <link/>

La balise <link/> permet d'incorporer le contenu d'un fichier externe - une feuille de style CSS ou une icône - dans une page web.

```
<link rel="stylesheet" href="monstyle.css">
<link rel="icon" href="/favicon.ico" type="image/x-icon">
```

Les métadonnées : la balise <meta/>

La balise <meta /> serve à placer des métadonnées dans la page HTML. Ce sont des informations utilisées par les serveurs web, les navigateurs web et les moteurs de recherche comme l'encoding (charset) utilisé, la description de la page, les mots clés, etc.

```
<meta charset="UTF-8">
<meta name="description" content=" cours développement web">
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
<meta name="author" content="Youssef Khalis">
```

Les scripts internes : la balise <script/>

La balise <script /> permet de lier une page HTML avec un fichier externe contenant du code JavaScript.

```
<script type="text/javascript" src=" monscript.js"></script>
```

Il est possible aussi d'incorporer du code JavaScript directement dans une page HTML via la même balise <script>.

```
<script>
  function maFunction
  {
    alert("Bonjour HTML5!");
  }
</script>
```

L'incorporation des styles : la balise <style>

Il est possible de placer directement les styles CSS dans le code HTML en tant que contenu de la balise <style>.

```
<style>
  h1 {color : red ; }
  h2 {color : blue ; }
</style>
```

Le corps du document : la balise <body>

Les balises <body> ... </body> constituent le corps du document. C'est le conteneur de l'ensemble des éléments textuels et graphiques d'une page web.

Un document HTML minimal se présente ainsi :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Titre de la page</title>
```

```
<link rel="stylesheet" href="style.css">
<script src="script.js"></script>
</head>
<body>
  <!-- contenu affichable -->
</body>
</html>
```

C. Structurer le contenu d'une page

Chaque élément HTML a un affichage par défaut dans le navigateur. Il ya deux types d'affichage disponible :

➔ **L'affichage en bloc** : Un élément de type bloc commence toujours sur une nouvelle ligne et prend toute la largeur disponible du navigateur.

Les principaux éléments de type bloc de HTML sont :

- <div>
- <h1>-<h6>
- <p>
- <form>
- <body>
-

- <table>

➔ **L'affichage en ligne** : Un élément de type en ligne correspond à un mot ou à une suite de mots à l'intérieur d'un élément bloc. Il ne commence pas par une nouvelle ligne et ne prend que la largeur nécessaire.

Les principaux éléments de type en ligne de HTML sont :

-
- <a>
-
-
- <i>
- <input>
- <label>
- <select>

C.1. Les grandes divisions de la page

La balise <div>

La balise <div> crée une division dans une page HTML qui permet de grouper dans un seul bloc un ensemble composé soit de texte brut, soit d'autres balises comme les éléments en ligne, les titres, les listes et formulaires, etc...

```
<div id="definition" class="style1" >
  <h2> Settat </h2>
  <p>Settat est une ville et commune du Maroc située dans la région Casablanca-Settat, chef-lieu de la province de Settat. Elle est située dans la région géographique et historique de la Chaouia, à 57 km de Casablanca, sur la route de Marrakech. </p>
</div>
```

La balise <header>

La balise <header> constitue l'en-tête de la page. Elle peut contenir un logo et/ou une liste de liens organisés en menu.

```
<header>
  <h1>Titre principal</h1>
  
</header>
```

La balise <footer>

La balise <footer> a pour rôle de contenir le pied de page qui peut être commun à toutes les pages du même site et inclure des informations comme une adresse de contact, le copyrights, les coordonnées de l'entreprise, etc...

```
<footer>
  <p>Posté par : Ayman Raji</p>
  <p>Contact : <a href="mailto:someone@example.com">someone@example.com</a></p>
</footer>
```

La balise <aside>

La balise <aside> est le conteneur d'informations ou de liens connexes avec le contenu principal. Elle pourrait être utilisée par exemple pour faire référence à des sujets complémentaires ou pour définir des termes spécifiques employés dans le contenu principal.

```
<p>Ma famille et moi ont visité Oukaïmden le weekend dernier.</p>
<aside>
  <h4>Oukaïmden </h4>
  <p>Oukaïmden est une station de ski situé à 80 km de Marrakech.</p>
</aside>
```

La balise <nav>

La balise <nav> contient la liste des liens utiles à la navigation vers différentes pages du site ou vers des pages externes.

```
<nav>
  <ul>
    <li><a href="index.html">Accueil</a></li>
    <li><a href="forum.html">Forum</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

C.2. Les divisions secondaires

Les titres

Les titres divisent un document en sections. Ils permettent de :

- Créer une structure organisée ;
- Décomposer la fluidité du texte sur la page ;
- Constituent des indices visuels du regroupement des sujets abordés dans le contenu.

HTML comprend six éléments pour représenter les différents niveaux de titre dans une page web :

- `<h1>` est le titre le plus important.
- `<h6>` est le titre le moins important.

```
<h1>Titre de niveau 1</h1>
<h2> Titre de niveau 2</h2>
<h3> Titre de niveau 3</h3>
<h4> Titre de niveau 4</h4>
<h5> Titre de niveau 5</h5>
<h6> Titre de niveau 6</h6>
```

Les paragraphes : la balise `<p>`

Le contenu d'une page HTML peut être divisé en différents paragraphes. Chaque paragraphe sera par défaut précédé et suivi d'un saut de ligne pour marquer la séparation avec le contenu précédant et suivant. Chaque paragraphe doit être contenu dans l'élément `<p>`, et donc délimité par les balises `<p>` et `</p>`

```
<p> Ceci est un texte dans un paragraphe.</p>
```

Les articles : la balise `<article>` et les sections

La balise `<article>` définit un contenu indépendant et autonome du document qui possède une identité à part entière dans la page comme l'article d'un blog, un post dans un forum ou un produit dans un site commercial.

Les balises `<section>` et `<article>` ont été introduits avec HTML5 afin de fournir des éléments sémantiques à utiliser pour la division du contenu textuel ou graphique d'une page en plus des éléments traditionnels `<div>` et `<p>`.

```
<article>
  <h1>Marrakech</h1>
  <p> Marrakech, surnommée la Ville rouge ou la ville ocre, est une ville située dans le centre du Maroc, dans
    l'intérieur des terres, au pied des montagnes de l'Atlas.</p>
</article>
```

Les sections : la balise `<section>`

La balise `<section>` détermine les sections d'un document, telles que les chapitres, les en-têtes, les pieds de page ou toute autre section du document.

```
<section>
  <h1>Histoire</h1>
  <p>Marrakech fut fondée en l'an 1071 (an 463 de l'Hégire) par le souverain almoravide Youssef Ibn
    Tachfin10....</p>
</section>
```

C.3. Les divisions de bloc locales

Les blocs de citations : la balise <blockquote>

La balise <blockquote> indique que le texte contenu dans l'élément est une citation longue d'une autre source. Le texte est généralement affiché avec une indentation. Une URL indiquant la source de la citation peut être réalisée grâce à l'attribut cite.

```
<blockquote cite=" https : //fr.wikipedia.org/wiki/Marrakech">
  <p>Ceci est une citation issue de la page Wikipedia de Marrakech.</p>
</blockquote>
```

Le texte préformaté : La balise <pre>

La balise <pre> représente du texte préformaté. Elle permet d'afficher le texte tel qu'il a été encodé dans l'éditeur de texte. Les espaces, les tabulations et les retours chariot sont respectés à l'écran.

```
<pre>
body
{
  color : red ;
  text-decoration : underline ;
}
</pre>
```

L'adresse : La balise <address>

La balise <address> définit les informations de contact pour le propriétaire d'un document ou d'un article.

Si l'élément <address> se trouve dans l'élément <body>, il représente les informations de contact du document.

Si l'élément <address> se trouve dans un élément <article>, il représente les informations de contact de cet article.

```
<address>
  Ecrit par <a href="mailto : webmaster@example.com">Ahmed Raji</a>.<br>
  N'hésitez pas à nous contacter : <br>
  www.fsts.ac.ma <br>
  FST de Settat, Km 3, B.P. : 577 Route de Casablanca <br>
  Maroc
</address>
```

C.4. Les éléments des divisions sémantiques du texte

Les abréviations : <abbr>

La balise <abbr> représente une abréviation ou un acronyme comme RAM ou ONCF. Elle est surtout utilisée par les navigateurs, les systèmes de traduction et les moteurs de recherche.

```
<p> la <abbr title="Faculté des Sciences et Techniques">FST</abbr> de Settat propose plus de 30 filières de formation.</p>
```

Le sens de lecture du texte : <bdo>

La balise <bdo> (pour Bi-Directional Override) est utilisé afin d'imposer une direction spécifique à un texte.

```
<p>Ce texte se lit de gauche à droite.</p>
<p><bdo dir="rtl">Ce texte se lit de droite à gauche</bdo></p>
```

Les citations : <cite>

La balise <cite> contient le titre d'une œuvre telle qu'une peinture, une chanson, une sculpture, un livre, un film, etc. Cet élément peut contenir le titre de l'œuvre, le nom de son auteur ou une URL de référence.

```
<p>
  <cite>La Joconde</cite> est un tableau de l'artiste italien Léonard de Vinci, réalisé entre 1503 et 1506.
</p>
```

L'inclusion de code source : <code>

La balise <code> représente un fragment de code machine.

```
<p>
  Texte normal.
  <code>alert('Bonjour HTML5'); </code>
  Texte normal.
</p>
```

Les définitions en ligne : <dfn>

La balise <dfn> représente la définition d'un terme.

```
<p>
<dfn title="HyperText Markup Language">HTML</dfn> est le format de données conçu pour représenter les pages web. </p>
```

Les exemples en ligne : <samp>

La balise <samp> est un élément qui permet de représenter un résultat produit par un programme informatique.

```
<p>
  Texte normal.
  <samp>Extrait de texte produit par un programme.</samp>
  Texte normal.
</p>
```

Les divisions en ligne :

La balise est un conteneur générique en ligne. Il peut être utilisé pour grouper des éléments afin de les mettre en forme ou parce qu'ils partagent certaines valeurs d'attribut comme [lang](#). Il doit uniquement être utilisé lorsqu'aucun autre élément sémantique n'est approprié. est très proche de l'élément <div>, mais l'élément <div> est un élément de bloc, alors que est un élément en ligne.

```
<p>J'écris avec un stylo <span style="color : blue">bleu</span>.</p>
```

Le conteneur des dates : <time>

La balise <time> permet de définir une date et/ou une heure. L'attribut facultatif `datetime` indique la date et l'heure selon la représentation numérique internationale de la norme ISO 8601.

```
<p>Nous ouvrons à <time>09 : 00</time> chaque matin.</p>  
<p>La journée internationale des travailleurs est <time datetime="2017-05-01"> le 1 mai</time>.</p>
```

C.5. Les styles physiques

HTML fournit des éléments qui permettent de créer des styles physiques pour le contenu. Ils peuvent faire des formatages de texte des plus basiques qu'on trouve couramment dans les logiciels de traitement de texte comme mettre du texte en gras, en italique, en exposant ou en indice, etc.

Mettre un texte en gras

Pour mettre du texte en gras, il suffit d'inclure celui-ci entre les balises

Dans le même ordre d'idée, la balise ... permet également de mettre en gras dans les navigateurs visuels.

La différence entre les balises et est pourtant sensible en HTML. La balise induit un effet purement typographique et visuel. La balise a pour but de mettre en avant, de renforcer le texte en lui ajoutant une importance sémantique.

```
<p>le contenu suivant est en <b>gras</b> mais ceci est <strong>important</strong></p>
```

Mettre un texte en italique

Pour mettre le texte en italique, il faut l'inclure entre les balises <i> ... </i>.

Dans le même registre, la balise produit le même aspect visuel que la balise <i> mais dans le but de mettre en évidence le texte et lui donner une importance sémantique.

```
<p>le contenu suivant est en <i>italique</i> mais ceci est <em>mis en évidence</em></p>
```

Modifier la taille du texte

A l'intérieur d'une division de la page, on peut modifier la taille relative du texte en utilisant la balise <small>. Il permet d'afficher son contenu avec une taille de police plus petit que celle du contexte.

```
<h2>Ce texte est <small>petit</small> mais pas cela</h2>
```

Créer des exposants et des indices

Avec HTML, il est possible de mettre des caractères en exposant ou en indice. Pour mettre un texte en exposant, il faut l'inclure dans la balise <sup>, et pour écrire un texte en indice, il faut l'inclure dans la balise <sub>.

```
<p>
  La molécule d'eau est H<sub>2</sub>O. Son coefficient de compressibilité est de 4,5×10<sup> 10</sup>
  Pa<sup>-1</sup>
</p>
```

Créer un retour à la ligne

La balise `
` permet de créer un retour à la ligne.

Surligner un texte

Afin de pouvoir surligner un texte, HTML a introduit la balise `<mark>` dont le contenu est mis en évidence dans tous les navigateurs par un fond jaune vif comme on le fait couramment avec un surligneur sur du papier.

```
<p><mark>Stabilo</mark> est un des principaux fabricants de surligneurs en Europe.</p>
```

Barrer un texte

La balise ` ... ` permet de marquer un élément de texte comme supprimé ou périmé, par exemple lors d'une promotion dans un site commercial. Le texte apparaît alors barré à l'écran.

```
<p>Promotion : un T-Shirt au prix de <del>150</del> 99DH.</p>
```

C.6. Les listes

Les listes sont une façon efficace pour structurer un contenu textuel. Avec HTML, on peut créer des listes ordonnées, des listes non ordonnées et des listes de définitions.

Les listes ordonnées

Une liste ordonnée est mise en place par la balise ` ... `.

On détermine ensuite, à l'intérieur de ces balises, les éléments ou les items de la liste, soit les balises `... `.

La structure générale d'une liste ordonnée ou numérotée est donc :

```
<ol>
  <li>Les listes ordonnées</li>
  <li>Les listes non ordonnées</li>
  <li>Les listes imbriqués</li>
  <li>Les listes de définition</li>
</ol>
```

On peut aussi utiliser quelques attributs comme : `reverse`, `start`, `type` :

- L'attribut `type` permet de remplacer les chiffres par les lettres (majuscules ou minuscules) ou des chiffres romains.
- L'attribut `start` permet de commencer la numérotation à un autre chiffre que le 1
- L'attribut `reversed` permet une numérotation descendante (5, 4, 3, 2, 1) au lieu de la numérotation ascendante habituelle (1, 2, 3, 4, 5).

```

<ol start="5" reversed="true" type="I" >
  <li>Les listes ordonnées</li>
  <li>Les listes non ordonnées</li>
  <li>Les listes imbriqués</li>
  <li> Les listes de définition</li>
</ol>

```

Les listes non ordonnées

Les listes non ordonnées ou les listes à puces, sont des listes d'éléments sans ordre particulier introduites par la balise ` ... `.

On détermine ensuite, à l'intérieur de cette balise, les éléments ou les items de celle-ci, soit les balises `...`.

La structure générale d'une liste à puces est donc :

```

<ul>
  <li>Les listes ordonnées</li>
  <li>Les listes non ordonnées</li>
  <li>Les listes imbriqués</li>
  <li> Les listes de définition</li>
</ul>

```

Les listes imbriquées

La balise ``, est le seul contenu autorisé pour les éléments `` et ``. Il peut lui-même avoir un contenu très varié, et contenir d'autres éléments `` ou ``. Ainsi, nous pouvons exploiter cette propriété pour créer des listes imbriquées les unes dans les autres.

La structure des listes ordonnées imbriquées est donc la suivante :

```

<ol>
  <li>Café
    <ul>
      <li>Espresso</li>
      <li>Café Latte</li>
      <li>Cappuccino</li>
      <li>Café Mocha</li>
      <li>Americano</li>
    </ul>
  </li>
  <li>Thé
    <ul>
      <li>Thé noir</li>
      <li>Thé vert</li>
    </ul>
  </li>
  <li>Lait</li>
</ol>
<dl>
  <dt>Espresso</dt>
  <dd>il s'agit de café infusé grâce à une petite quantité d'eau très chaude passant rapidement dans des grains de café moulus finement.</dd>
  <dt>Café Latte</dt>
  <dd>à base d'espresso (généralement allongé), on ajoute la même quantité de lait chaud ainsi qu'un peu de mousse.</dd>
  <dt>Cappuccino</dt>

```

```

<dd>d'origine italienne, ce café contient un espresso, un peu de lait chaud et beaucoup de mousse de lait.
    Son goût est généralement plus corsé que celui du café au lait puisque l'espresso est moins
    dilué.</dd>
<dt>Café Mocha</dt>
<dd>un espresso dans lequel on dissout du cacao, du sucre et on ajoute ensuite de la mousse de lait.</dd>
<dt>Americano</dt>
<dd>un espresso qu'on allonge avec de l'eau chaude et qui va ressembler à un café filtre mais avec le goût
    de l'espresso.</dd>
</dl>
</ol>

```

Les listes de définition

Une liste de définitions permet de créer une liste de termes, chacun d'entre eux étant suivi de sa définition. Le conteneur de l'ensemble de la liste est la balise `<dl>` qui ne peut contenir que des balises `<dt>`, `<dd>` ou la balise `<dl>` elle-même et rien d'autre. À l'intérieur d'une balise `<dd>`, vous pouvez mettre des paragraphes, des sauts de ligne, des images, des liens, des listes, etc. Par défaut, la définition est affichée à la ligne et indentée par rapport à la ligne du terme.

La structure générale d'une liste de définition est donc :

```

<dl>
<dt>Espresso</dt>
<dd>il s'agit de café infusé grâce à une petite quantité d'eau très chaude passant rapidement dans des grains
    de café moulus finement.</dd>
<dt>Café Latte</dt>
<dd>à base d'espresso (généralement allongé), on ajoute la même quantité de lait chaud ainsi qu'un peu de
    mousse.</dd>
<dt>Cappuccino</dt>
<dd>d'origine italienne, ce café contient un espresso, un peu de lait chaud et beaucoup de mousse de lait. Son
    goût est généralement plus corsé que celui du café au lait puisque l'espresso est moins dilué. </dd>
<dt>Café Mocha</dt>
<dd>un espresso dans lequel on dissout du cacao, du sucre et on ajoute ensuite de la mousse de lait.</dd>
<dt>Americano</dt>
<dd>un espresso qu'on allonge avec de l'eau chaude et qui va ressembler à un café filtre mais avec le goût de
    l'espresso.</dd>
</dl>

```

D. Insérer des images et du multimédia

D.1. Les types d'images

Les navigateurs actuels n'acceptent qu'un nombre restreint de types d'images. Les trois grands types les plus utilisés sont supportés par tous les navigateurs : GIF, JPEG et PNG.

- **GIF** : Graphics Interchange Format, ayant pour extension.gif. Ce format est limité à 256 couleurs et est donc déconseillé pour les photographies ayant une grande plage de teintes différentes.
- **JPEG** : Joint Photographic Experts Group, ayant pour extensions.jpeg ou.jpg. Ce format permet la création d'images en 24bits (16 millions de couleurs) et est donc très adapté aux photographies réalistes. En contrepartie, les images JPEG ont habituellement un poids plus important en Ko, ce qui ralentit leur chargement.
- **PNG** : Portable Network Graphics, ayant pour extension.png. Ce format est assez récent et a été conçu comme alternative au format GIF qui faisait à l'époque l'objet de droits d'auteurs. Le format PNG créé

à l'initiative du W3C est donc libre de droit et permet normalement la création de graphiques et de photographies.

D.2. L'insertion d'images et du multimédia

La balise

La balise permet d'inclure de images dans une page web. Il s'agit d'un élément vide de type en ligne.

Voici les principaux attributs de cet élément :

- **Src** : le seul attribut obligatoire de cette balise. Il précise l'adresse relative ou absolue du fichier image que l'on désire afficher dans la page.

```
  

```

- **Alt** : il contient un texte fournissant une brève description de l'image. Ce texte apparaîtra à la place de l'image si celle-ci n'est pas disponible.

```

```

- Les attributs **height** et **width** permettent de définir respectivement la hauteur et la largeur qu'aura l'image sur la page web.

```

```

Titre d'une d'image

HTML5 proposent de nouveaux éléments qui permettent de titrer correctement une image, une illustration ou un diagramme. Il s'agit de l'élément <figure> dans lequel sont insérer un élément qui permet l'affichage de l'image et un élément <figcaption> qui crée un titre a syntaxe générale de l'ensemble est la suivante :

```
<figure>  
    
  <figcaption>Super Mario Bros</figcaption>  
</figure>
```

L'insertion du multimédia

Les images ne sont pas les seuls éléments décoratifs d'une page web. A cet effet, nous disposons désormais de multiples éléments multimédias qui vont des animations Flash à la vidéo, en passant par les fichiers son et les applets java. Avec HTML5, on peut incorporer ces fichiers multimédias de manière conforme en oubliant l'élément <applet> désormais obsolète. Deux éléments HTML sont disponible : <object> et <embed />.

L'insertion d'une animation Flash

Afin d'incorporer des fichiers multimédias comme une animation Flash dans du code HTML deux éléments sont disponibles : <object> et <embed />.

```
<object width="400" height="400" data="fic.swf" type="application/x-shockwave-flash" ></object>  
<embed width="400" height="400" src="fic.swf" type="application/x-shockwave-flash" />
```

L'insertion d'une vidéo

Pour insérer une vidéo dans une page, on utilise la balise `<video>`. Actuellement, il existe 3 formats vidéo pris en charge pour l'élément `<video>` : MP4, WebM et Ogg.

La syntaxe de base de l'élément `<video>` est la suivante :

```
<video src="URL fichier" >...</video>
```

L'insertion d'une audio

La balise `<audio>` est utilisée avec HTML afin d'intégrer un contenu sonore dans un document. Il peut contenir une ou plusieurs sources audio représentées avec l'attribut `src` ou l'élément `<source>`. S'il y a plusieurs sources, l'agent utilisateur choisira celle qui convient le mieux. Actuellement, il existe 3 formats audio pris en charge pour l'élément `<audio>` : MP3, Wav et Ogg.

Sa syntaxe est la suivante :

```
<audio src="URL fichier" controls="controls" preload="none|metadata|auto" loop="loop" autoplay="autoplay" muted="muted"></audio>
```

E. Les liens

E.1. L'insertion d'un lien

Les liens constituent l'essence du langage HTML et des pages Web. La richesse de la fonction hypertexte permet de tisser cette toile gigantesque qu'est le Web.

La balise `<a> ... ` introduit un lien. Sa syntaxe de base est :

```
<a href="destination_du_lien">Texte de lien</a>
```

La destination du lien peut 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.

La balise `<a>` possède différents attributs :

Href : l'attribut `href` définit l'adresse (URL) de la cible du lien.

Hreflang : précise la langue du document cible si celle-ci est différente de celle du document de départ.

Rel : spécifie la relation entre le document de départ et le fichier cible du lien.

Les valeurs sont nombreuses : `alternate`, `author`, `bookmark`, `external`, `help`, `license`, `next`, `nofollow`, `noreferrer`, `noopener`, `prev`, `search`, `tag`.

Type : Indique au navigateur le type MIME de la cible si celui-ci n'est pas un document HTML5, par exemple un fichier son, une image ou un fichier PDF.

Target : indique au navigateur la façon d'afficher la cible du lien ; il prend les valeurs suivantes :

- **_blank** : cible dans une nouvelle fenêtre ;
- **_self** : cible dans la même fenêtre ;
- **_parent** : cible affichée dans la fenêtre parent immédiat de celle du lien ;
- **_top** : cible affichée dans la fenêtre parent de plus haut niveau de celle du lien.

E.2. Les liens externes

Les liens externes permettent de diriger le visiteur d'un site vers de nouveaux documents non contenus dans la page d'origine. Un lien externe peut faire référence soit à des documents du même site ou des documents d'autres sites, situés à une autre adresse sur le Web.

Pour créer un lien vers une page externe au site, on doit indiquer l'adresse URL complète de la page dans l'attribut href de la balise `<a>`. On parle alors d'adressage absolu.

Soit par exemple :

```
<a href=" https : //www.w3.org/standards/">Lien</a>
```

Pour créer un lien vers une page interne du même site, on utilise une adresse relative.

Soit par exemple :

```
<a href="page1.html">Lien 1</a>
```

La page page1.html doit se trouver sur le serveur, dans le même dossier que la page en cours qui contient le lien. Dans le cas contraire, le navigateur affichera au visiteur une page d'erreur nommée Erreur 404 qui indique que la page demandée n'existe pas.

Si on veut créer un lien vers une page située dans un autre dossier, on peut

- Utiliser un chemin d'accès relatif à partir de la racine du site.

```
<a href="/dossier1/page1.html">Lien 1</a>
```

- Utiliser un chemin d'accès relatif en indiquant le dossier si la page et le dossier se situent au même niveau de l'arborescence.

```
<a href="dossier1/page1.html">Lien 1</a>
```

- Utiliser un chemin d'accès relatif en remontant d'abord en utilisant la notation ../ pour être au même niveau puis descendre jusqu'à la page située dans l'autre dossier.

```
<a href="../dossier1/page1.html">Lien 1</a>
```

E.3. Les liens ciblés : les ancres

Quand le contenu d'une page est volumineux, l'utilisateur ne peut pas en avoir une vision globale. Il est alors souhaitable de lui proposer un menu composé de liens internes vers les différentes sections de la page. Il pourra

ainsi accéder directement au point de son choix sans faire défiler la page. De même, si le lien est externe, il est possible d'accéder en un point particulier de la page cible. Chaque point cible de la page doit être signalé par un lien particulier, appelé ancre ou signet, lequel est créé à l'aide de la balise `<a>`, muni simplement d'un attribut `id` dont la valeur doit être unique dans la page.

Soit par exemple la page suivante :

```
<p>
  <a id="para1">Paragraphe 1</a> HTML5 est la dernière révision majeure de l'HTML.
</p>
<p>
  <a id="para2">Paragraphe 2</a> les CSS forment un langage informatique qui décrit la présentation des
  documents HTML et XML....
</p>
```

Chaque paragraphe alors devient la cible d'un lien. Il suffit de créer un élément `<a>` avec l'attribut `href` contenant le nom de l'ancre (attribut `id` de la cible) directement précédé du signe dièse (`#`).

```
<p><a href="#para1">Vers le Paragraphe 1</a></p>
```

Si l'ancre est située dans une autre page HTML5, l'attribut `href` contient l'adresse relative ou absolue du document, suivie du caractère dièse, puis de l'identifiant de l'ancre.

Pour une adresse relative :

```
<p><a href="page2.html#para1">Vers le Paragraphe 1</a></p>
```

Pour une adresse absolue :

```
<p><a href="http://www.monsite.ma/page2.html#para1">Vers le Paragraphe 1</a></p>
```

Pour créer un menu sous forme de liste, situé en haut de la page, et qui permet d'accéder directement à chaque partie de la page, on peut écrire le code suivant :

```
<ul>
  <li><a href="#cours1">HTML5</a></li>
  <li><a href="#cours2">CSS</a></li>
  <li><a href="#cours3">JavaScript</a></li>
  <li><a href="#cours4">PHP</a></li>
</ul>
```

E.4. Les liens pour télécharger des fichiers

Pour permettre de télécharger un fichier, il suffit de spécifier le nom (avec l'extension) de celui-ci. Le fichier peut être sur le même site ou dans un autre site.

Soit un fichier "cours_html5.pdf" situé dans le même dossier. Le code est :

```
<a href="cours_html5.pdf">Télécharger le cours</a>
```

Si une application définie par défaut pour l'extension du fichier, le navigateur ouvre l'application et affiche le fichier. Dans le cas contraire, le navigateur propose la fenêtre d'invite de téléchargement afin de pouvoir télécharger le fichier en question.

E.5. Lien à partir d'une image ou un élément de type bloc

Le contenu de l'élément `<a>` est généralement composé de texte, mais il peut également inclure des éléments graphiques, tel que des images. Avec HTML5, le contenu peut même être des éléments de type bloc comme les divisions (`<div>`), les paragraphes (`<p>`), les listes (`` ou ``), etc...

Par exemple, pour une image, on peut écrire le code suivant :

```
<a href="cours_html5.html">
  
</a>
```

Pour une balise `<div>`, on peut écrire :

```
<a href="cours_html5.html">
  <div>du texte </div>
</a>
```

E.6. Le lien vers une adresse email

Pour ajouter un élément d'interactivité à un site web, il est intéressant de permettre aux visiteurs d'entrer en contact avec l'administrateur du site par courrier électronique afin qu'ils envoient leurs observations ou questions.

Pour réaliser cette opération, on doit créer un type de lien particulier provoquant, en un clic, l'ouverture automatique du logiciel de messagerie électronique par défaut du visiteur avec comme destinataire l'adresse que l'auteur du site aura choisi dans son code. Ainsi le contenu de l'attribut `href` du lien sera composé du nom du protocole `mailto` : suivi de l'adresse e-mail du contact.

```
<a href="mailto:nabil@monsite.ma">Contact</a>
```

On peut aussi ajouter d'autres options au lien comme des destinataires en copie simple (`cc`) ou caché (`bcc`), le sujet de l'e-mail, etc.

```
<a href="mailto:nabil@monsite.ma?subject=Observations&cc=admin@monsite.ma ">Contact</a>
```

E.7. Les liens déclenchant un script JavaScript

Un lien peut également être utilisé pour déclencher l'exécution d'un script JavaScript. Pour cela, il faut que l'attribut `href` de la balise `<a>` soit composé du nom JavaScript, suivi du code que l'on désire exécuter en cas de clic sur le lien.

Si le code est court, il peut être écrit directement comme dans l'exemple suivant :

```
<a href="javascript : alert('Bonjour JavaScript!');">Exécuter le script</a>
```

Sinon, il est conseillé de définir une fonction JavaScript directement dans la page web via la balise `<script>` ou dans un fichier JavaScript externe et l'inclure dans la page via la balise `<script>` et faire l'appel de la fonction dans l'attribut href :

```
<a href="javascript : mafonction();">Exécuter le script</a>
```

F. Les tableaux

Dans une page HTML, il est possible de réaliser une présentation d'information de type textuelle ou graphique sous la forme de tableau.

F.1. La structure générale d'un tableau

L'élément essentiel dans la création de tableau est `<table>` : il s'agit du conteneur renfermant tous les autres éléments de tableau.

La structure d'un tableau HTML est créée ligne par ligne. Chaque ligne est déclarée par un élément `<tr>`. La création de chaque cellule d'une ligne est déclarée par un élément `<td>` pour les cellules standards du tableau ou `<th>` pour les cellules qui jouent le rôle d'en-tête de colonne ou de ligne. Ces éléments sont les seuls enfants autorisés pour l'élément `<tr>`. Ainsi on peut illustrer la structure générale d'un tableau par le code suivant :

```
<table>
  <tr>
    <td>Bill</td>
    <td>Gates</td>
    <td>62</td>
  </tr>
  <tr>
    <td>Steve</td>
    <td>Jobs</td>
    <td>56</td>
  </tr>
</table>
```

Remarques :

- En HTML5, l'élément `<table>` n'a pas d'attribut, tous doivent être pris en charge par des feuilles de style CSS.
- Les éléments `<td>` sont les conteneurs de données de la table. Ils peuvent contenir toutes sortes d'éléments HTML ; Texte, images, listes, autres tableaux, etc.

La largeur du tableau

La largeur du tableau est déterminée par la propriété de style CSS width.

```

<table style="width : 100%">
  <tr>
    <td>Bill</td>
    <td>Gates</td>
    <td>62</td>
  </tr>
  <tr>
    <td>Steve</td>
    <td>Jobs</td>
    <td>56</td>
  </tr>
</table>

```

Les bordures du tableau

Par défaut, un tableau est affiché sans bordures.

Les bordures du tableau sont définies à l'aide de la propriété de style CSS border (pour bordure) par le code suivant :

```

table, th, td
{
  border : 1px solid black ;
}

```

| | | |
|-------|-------|----|
| Bill | Gates | 62 |
| Steve | Jobs | 56 |

Cela correspond à la bordure classique d'un tableau avec à la fois les bordures du tableau et les bordures des cellules. Pour réduire ces deux bordures en une seule, on utilise la propriété de style border-collapse.

```

table, th, td
{
  border : 1px solid black ;
  border-collapse : collapse ;
}

```

| | | |
|-------|-------|----|
| Bill | Gates | 62 |
| Steve | Jobs | 56 |

L'alignement du tableau

Le tableau est aligné à gauche. Pour un alignement centré, il faut mettre une marge (margin) automatique à gauche (left) et à droite (right) du tableau qui provoque le centrage.

```

table
{
  margin-left : auto ;
  margin-right : auto ;
}

```

L'alignement des cellules du tableau

Par défaut, le contenu des cellules est aligné à gauche. Pour les centrer dans les cellules, on peut utiliser la propriété de style text-align.

```
table
{
  text-align : center ;
}
```

L'espacement entre les bordures

L'espacement entre les bordures spécifie l'espace entre les cellules d'un tableau. Pour définir l'espacement des bordures pour une table, on utilise la propriété de style CSS border-spacing :

```
table
{
  border-spacing : 5px ;
}
```

F.2. Les cellules de tableaux

Les cellules peuvent contenir tous les éléments définis par le HTML, soit du texte, des images, des liens, des arrière plans et même des tableaux.

La largeur des cellules

Par défaut, le navigateur adapte la largeur des cellules selon leur contenu.

Pour uniformiser la largeur des cellules, il faut appliquer la propriété de style width à la balise <td>.

```
td
{
  width : 33% ;
}
```

L'alignement horizontal

L'alignement horizontal du contenu d'une cellule peut être à gauche, centré ou à droite. Celui-ci est obtenu par la propriété de style text-align appliquée à une cellule. La propriété text-align peut prendre la valeur left (gauche), center (centré) ou right (droite). La valeur par défaut est left.

```
td
{
  text-align : center ;
}
```

La marge intérieure des cellules

Par défaut, le contenu de la cellule est accolé à la bordure de celle-ci. La propriété de style padding permet d'ajouter de l'espace entre le contenu de la cellule et la bordure de celle-ci.

```
td
{
```

```
padding : 15px ;  
}
```

F.3. La fusion des cellules

Il est possible de fusionner horizontalement ou verticalement des cellules.

La fusion de colonnes

Pour fusionner des colonnes, le HTML dispose de l'attribut de cellule `colspan="x"` où `x` correspond au nombre de colonnes que l'on souhaite fusionner horizontalement.

```
<table >  
  <tr>  
    <td>Nom</td>  
    <td colspan="2">Téléphone</td>  
  </tr>  
  <tr>  
    <td>Bill Gates</td>  
    <td>55577854</td>  
    <td>55577855</td>  
  </tr>  
</table>
```

| | | |
|------------|-----------|----------|
| Nom | Téléphone | |
| Bill Gates | 55577854 | 55577855 |

La fusion de lignes

Pour fusionner des lignes, le HTML dispose de l'attribut de cellule `rowspan="x"` où `x` correspond au nombre de lignes que l'on souhaite fusionner verticalement.

```
<table >  
  <tr>  
    <td>Nom : </th>  
    <td>Bill Gates</td>  
  </tr>  
  <tr>  
    <td rowspan="2">Téléphone : </td>  
    <td>55577854</td>  
  </tr>  
  <tr>  
    <td>55577855</td>  
  </tr>  
</table>
```

| | |
|-------------|------------|
| Nom : | Bill Gates |
| Téléphone : | 55577854 |
| | 55577855 |

F.4. Les cellules d'entête

Les cellules d'entête de colonnes ou de lignes sont définies par la balise `<th> ... </th>`. Elles fournissent un titre aux données de la colonne ou de la ligne. Dans les navigateurs visuels, le contenu de la balise `<th>` apparaît en gras et en centré. La balise `<th>` a été gardée dans la spécification du HTML5 car elle est d'une grande utilité avec les lecteurs d'écran utilisé par les personnes non voyantes. Elle contribue ainsi à l'accessibilité des sites web et particulièrement des tableaux qui, s'ils sont d'une certaine ampleur, posent vite problème lorsqu'ils sont découverts par l'audition.

Un exemple d'un tableau avec entête est :

```
<table>
  <tr>
    <th>Nom</th>
    <th>Téléphone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>555 77 854</td>
  </tr>
</table>
```

| Nom | Téléphone |
|------------|------------|
| Bill Gates | 555 77 854 |

Un exemple d'un tableau avec entête vertical est :

```
<table>
  <tr>
    <th>Nom : </th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th>Téléphone : </th>
    <td>555 77 854</td>
  </tr>
</table>
```

| | |
|--------------------|------------|
| Nom : | Bill Gates |
| Téléphone : | 555 77 854 |

F.5. La légende d'un tableau

La balise `<caption>` permet d'associer un titre ou une légende au tableau.

La balise `<caption>` doit être placée juste après la balise ouvrante `<table>` et ne peut apparaître qu'une fois dans le tableau.

Par défaut, la plupart des navigateurs affichent le contenu de la balise `<caption>` de façon centrée au-dessus du tableau. Il est possible de mettre cette légende en dessous du tableau à l'aide de la propriété de style CSS `caption {caption-side : bottom}` ou l'aligner vers la droite ou la gauche via la propriété `text-align`.

```

<table>
  <caption>Répertoire</caption>
  <tr>
    <th>Nom : </th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th>Téléphone : </th>
    <td>555 77 854</td>
  </tr>
</table>

```

| Répertoire | |
|--------------------|------------|
| Nom : | Bill Gates |
| Téléphone : | 555 77 854 |

F.6. Le groupement de colonnes

La balise `<colgroup>` est utilisée pour grouper des colonnes en vue de leur appliquer une mise en forme à l'ensemble des balises du groupe par l'entremise d'une feuille de style CSS.

L'attribut `span="x"` détermine le nombre de colonnes ainsi groupées. Par défaut, la valeur de `x` est 1. Cet attribut `span` est le seul attribut de la balise `<colgroup>`. En outre, la balise `<colgroup>` se positionne juste après la balise `<table>` (ou juste après la balise `<caption>` si celle-ci est utilisée) et avant toute balise `<thead>`, `<tbody>`, `<tfoot>` ou `<tr>`.

La balise `<col>` spécifie les propriétés des colonnes pour chaque colonne dans un élément `<colgroup>`.

Pour définir différentes propriétés d'une colonne dans un élément `<colgroup>`, on utilisera donc la balise `<col>` à l'intérieur d'une balise `<colgroup>`.

Par définition, la balise `<colgroup>` ne s'applique que pour les tableaux.

```

<table style="width : 500px">
  <colgroup>
    <col span="2" style="background-color : red">
    <col style="background-color : yellow">
    <col style="background-color : green">
  </colgroup>
  <tr>
    <th>Nom</th>
    <th>Prénom</th>
    <th>Age</th>
    <th>Adresse</th>
  </tr>
  <tr>
    <td>Bill</td>
    <td>Gates</td>
    <td>62</td>
    <td>500 Fifth Avenue North Seattle, WA 98102, États-Unis</td>
  </tr>
  <tr>
    <td>Steve</td>
    <td>Jobs</td>
    <td>56</td>
    <td>2101 Waverley Street Palo Alto, CA 94301</td>
  </tr>

```

```
</tr>
</table>
```

| Nom | Prénom | Age | Adresse |
|-------|--------|-----|--|
| Bill | Gates | 62 | 500 Fifth Avenue North Seattle, WA 98102, États-Unis |
| Steve | Jobs | 56 | 2101 Waverley Street Palo Alto, CA 94301 |

F.7. La structuration d'un tableau

Le HTML prévoit des balises pour structurer de façon logique le contenu d'un tableau.

Ces balises sont :

- `<thead>` pour regrouper les informations concernant l'entête du tableau comme par exemple le titre et l'intitulé des colonnes.
- `<tbody>` pour le corps du tableau, soit l'ensemble des données de celui-ci.
- `<tfoot>` pour le contenu de bas de page comme par exemple des remarques ou une légende.

Ces divisions logiques n'affectent en rien la présentation par défaut du tableau mais elles peuvent être reprises par des propriétés de style pour agrémenter la présentation du tableau.

Les navigateurs peuvent utiliser ces éléments pour activer le défilement du corps de table indépendamment de l'en-tête et du pied de page. En outre, lors de l'impression d'une table de grande taille qui s'étend sur plusieurs pages, ces éléments permettent d'afficher l'en-tête de la table et le pied de page à imprimer en haut et en bas de chaque page.

On notera que la balise `<tfoot>` doit se situer avant la balise `<tbody>` de sorte que le navigateur puisse prévoir le pied de page avant la réception des lignes de données.

```
<table style="width : 500px">
  <thead>
    <tr>
      <th>Nom</th>
      <th>Prénom</th>
      <th>Age</th>
      <th>Adresse</th>
    </tr>
  </thead>
  <tfoot>
    <td colspan="4" style="text-align : center ; ">Pied de page du tableau</td>
  </tfoot>
  <tbody>
    <tr>
      <td>Bill</td>
      <td>Gates</td>
      <td>62</td>
      <td>500 Fifth Avenue North Seattle, WA 98102, États-Unis</td>
    </tr>
    <tr>
      <td>Steve</td>
      <td>Jobs</td>
      <td>56</td>
      <td>2101 Waverley Street Palo Alto, CA 94301</td>
    </tr>
  </tbody>
```

```
</table>
```

| Nom | Prénom | Age | Adresse |
|-------------------------|--------|-----|--|
| Bill | Gates | 62 | 500 Fifth Avenue North Seattle, WA 98102, États-Unis |
| Steve | Jobs | 56 | 2101 Waverley Street Palo Alto, CA 94301 |
| Pied de page du tableau | | | |

G. Les formulaires

Les formulaires occupent une place prépondérante dans la conception et l'exploitation d'une application ou un site Web. C'est en effet la seule façon de recevoir en retour des informations provenant directement de l'utilisateur final, et en outre, structurées selon les besoins du concepteur.

Grâce au formulaire, l'utilisateur peut entrer des textes ou des mots de passe, opérer des choix avec des boutons radio, des cases à cocher ou des listes de sélection. Il peut également effectuer le transfert de ses propres fichiers vers le serveur.

G.1. La déclaration de formulaire

La balise `<form> ... </form>` a comme unique fonction de déclarer au navigateur qu'il doit mettre en place un formulaire. Elle englobera tous les éléments constitutifs d'un formulaire comme une ligne de texte, des cases à cocher, des listes déroulantes, etc.

```
<form>  
  Éléments du formulaire  
</form>
```

Les attributs usuels de la balise `<form>` sont :

Name : Le nom du formulaire.

Action : Lorsque vous donnez l'ordre au navigateur de transmettre les données du formulaire, il a besoin de connaître l'action qu'il doit effectuer.

Cette action sera :

- Soit l'adresse d'un programme de traitement des données, situé sur le serveur, en CGI, Perl, PHP, ASP...
Par exemple, `action = "http : //www.monsite.ma/traitement.php"`.
- Soit une adresse de courrier électronique pour récupérer simplement les données. Le protocole mailto est alors utilisé.
Par exemple, `action="mailto:mon_email@monsite.ma"`.
- Soit, lorsque les données d'un formulaire sont traitées en interne (côté-client) par du JavaScript, l'attribut action reste vide.
Par exemple `action=""`.

Enctype : Lorsque la valeur de l'attribut method est post, cet attribut définit le type MIME qui sera utilisé pour encoder les données envoyées au serveur. C'est attribut peut prendre les valeurs suivantes :

- **application/x-www-form-urlencoded** : la valeur par défaut si l'attribut n'est pas défini
- **multipart/form-data** : la valeur utilisée par un élément <input> dont l'attribut type vaut "file".
- **text/plain** : la valeur utilisée pour l'envoi vers une adresse électronique avec le protocole mailto.

Method : Cet attribut définit la méthode HTTP qui sera utilisée pour envoyer les données au serveur. C'est un attribut peut prendre deux valeurs :

- **get** : correspondant à la méthode GET du protocole HTTP. Les données du formulaire sont ajoutées à l'URI de l'attribut action avec '?' comme séparateur. L'URI ainsi composée est ensuite envoyée vers le serveur. Cette méthode est mieux adaptée pour des quantités courtes, non-sensibles, de données, car il a des limites de taille trop.
- **post** : correspondant à la méthode POST du protocole HTTP, les données du formulaire sont incluses dans le corps du formulaire et envoyées vers le server. Cette méthode est recommandée pour l'envoi de données sensibles ou personnelles. Elle n'a aucune limite de taille, et peut être utilisé pour envoyer de grandes quantités de données.

accept-charset : Spécifie les codages de caractères qui doivent être utilisés pour la soumission du formulaire.

autocomplete : indique si un formulaire doit avoir l'autocomplétions des champs activée (on) ou désactivée (off). Sa valeur par défaut est «on».

Novalidate : Cet attribut booléen indique si le formulaire doit être validé au moment de sa soumission.

G.2. Les composants communs

Quel que soit le rôle qu'on assigne à un formulaire, il contient généralement deux éléments communs à tous les formulaires : un bouton d'envoi et un bouton de réinitialisation.

Le bouton d'envoi

C'est un bouton qui envoie le formulaire. Sa syntaxe est la suivante :

```
<input type="submit" value="Envoi">
```

Il est possible de remplacer le bouton d'envoi par une image grâce à la balise <input type="image">.

```
<input type="image" src="submit.jpg" alt="Soumettre" width="150" height="150">
```

Le bouton réinitialisation

Le bouton réinitialisation ou d'annulation permet de remettre le formulaire dans son état initial, y compris les valeurs par défaut qui ont pu y être définies. Sa syntaxe est la suivante :

```
<input type="reset" value="Annuler">
```

Le bouton de commande

En plus de l'élément HTML <input> pour créer les boutons, il y a aussi la balise <button> ... </button>. Elle permet de déclencher, au clic de celui-ci, une action spécifique définie par le concepteur du site, généralement à l'aide de JavaScript. Elle offre également la possibilité de réaliser la soumission et l'annulation du formulaire (submit et reset).

En outre, cette balise `<button>` a l'avantage de posséder une balise d'ouverture et de fermeture. Il est ainsi possible d'y inclure du texte, des images ou du contenu Html. Notons également que cette balise `<button>` n'a pas besoin d'être imbriquée dans un formulaire `<form>` et peut ainsi être utilisée dans de multiples contextes. Tout ceci en fait une balise polyvalente et justifie son succès auprès des développeurs.

Par exemple, pour créer un bouton d'envoi avec du texte, on peut écrire :

```
<button type="submit" value="Envoi2">Envoi2</button>
```

Et pour créer un bouton d'envoi avec une image, on peut écrire :

```
<button type="submit" value="Envoi2">  
    
</button>
```

G.3. Les composants de saisie d'informations

La saisie d'une ligne de texte

La saisie d'une ligne de texte se fait avec la balise `<input>` en utilisant l'attribut `type="text"`. Les données saisies sont de types textuels tant alphabétiques que numériques sur une seule ligne. Les sauts de ligne sont automatiquement retirés de la valeur saisie.

```
<input type="text" name="Nom">
```

La saisie de mot de passe

Les champs de saisie de mot de passe sont quasi identiques aux champs de saisie de texte. Ils ne comportent qu'une seule ligne et sont créés avec le même élément `<input />`. La différenciation entre ces deux champs réside dans la valeur de l'attribut `type` qui prend la valeur « password » au lieu de « text ». Pour l'utilisateur le champ a le même aspect visuel, mais quand il frappe son mot de passe, les caractères qu'il utilise ne sont pas affichés dans la zone et sont remplacés par un astérisque (*) ou un puce, ce qui le protège des regards indiscrets mais ne protège en aucun cas les données car elles seront transmises en clair.

```
<input type="password" name="code">
```

La saisie de texte long

Dans certaines situations, il faut prévoir plus d'espace à l'utilisateur pour s'exprimer. C'est le cas, par exemple, pour des commentaires, remarques ou suggestions. On utilise alors la balise `<textarea> ... </textarea>` qui introduit une zone de texte de plusieurs lignes.

```
<textarea rows="4" cols="30"></textarea>
```

La balise `<textarea>` est une balise non vide, Une valeur par défaut peut être prévue dans la zone de texte, de la façon suivante :

```
<textarea rows="4" cols="30">Vos suggestions ici! </textarea>
```

La saisie d'adresse e-mail

L'élément spécifique de saisie d'adresse e-mail est `<input />` dont l'attribut type prend la valeur email selon la syntaxe :

```
<input type="email" name="mail" >
```

Cette nouvelle balise HTML5 permet la validation directe de l'adresse mail par le navigateur, sans l'ajout de script JavaScript de validation.

La saisie de numéro de téléphone

Un autre élément `<input />` qui permet d'effectuer la saisie de numéro de téléphone uniquement en donnant à l'attribut type la valeur tel selon la syntaxe :

```
<input type="tel" name="telperso">
```

La saisie d'URL

La balise `<input type="url">` est destinée à recueillir des adresses Web (URL).

En fonction du support du navigateur, le champ url peut être validé automatiquement via le navigateur lorsqu'il est soumis sans avoir besoin d'une validation pas JavaScript.

```
<input type="url" name="homepage">
```

La saisie de l'heure

Pour créer un composant de saisie de l'heure, il faut donner la valeur time à l'attribut type avec la syntaxe :

```
<input type="time" name="heure">
```

Le format à respecter pour la saisie est hh : mm : ss.

La saisie de la date

Pour créer un composant de saisie de la date, il faut que l'attribut type ait la valeur date. Le format à respecter pour la saisie est AAAA-MM-JJ. La syntaxe est :

```
<input type="date" name="date">
```

La saisie de la date et l'heure

Quand l'attribut type de l'élément `<input />` prend la valeur datetime-local, il impose la saisie d'une date et d'une heure dans le même champ. Le format de saisie est AAAA-MM-JJThh : mm : ss, le caractère T servant de séparateur entre la date et l'heure. La syntaxe est :

```
<input type="datetime-local" name="dateheure">
```

Saisie du mois

Pour saisir le numéro du mois, il faut que l'attribut type de l'élément `<input />` ait la valeur month ; le format de saisie est AAAA-MM (comprend aussi l'année). La syntaxe est :

```
<input type="month" name="mois">
```

Saisie de la semaine

Le dernier composant de saisie du temps permet d'obtenir le numéro de la semaine en donnant la valeur week à l'attribut type. Le format de saisie AAAA-WNN est composé de l'année suivie d'un tiret, de la lettre W obligatoire et du numéro sur deux chiffres. La syntaxe est donc :

```
<input type="week" name="semaine">
```

La saisie des nombres

L'HTML5 introduit aussi des composants destinés à la saisie de nombres. Il s'agit des compteurs numériques et ceux qui ont pour vocation d'afficher un résultat.

La saisie des nombres s'effectue encore avec l'élément `<input />`, mais en donnant la valeur number à l'attribut type. La syntaxe est donc :

```
<input type="number" name="nombre" />
```

Un autre moyen de saisir un nombre est fourni par le composant qui permet de choisir une valeur dans un intervalle. Il s'agit d'un segment gradué muni d'un curseur que l'on peut faire glisser pour choisir une valeur. C'est encore l'élément `<input />` qui permet de créer en affectant à l'attribut type la valeur range selon le modèle :

```
<input type="range" name="intervalle" value="v" min="m" max="M" step="S" >
```

Où v, m, M et S sont la valeur initiale, le minimum, le maximum et l'incrément respectivement.

Pour afficher le résultat d'un calcul, HTML5 propose le composant `<output>`. Comme son nom l'indique, cet élément ne contient pas la saisie d'un utilisateur mais un résultat fourni par le navigateur ou le serveur distant. Sa syntaxe est :

```
<output name="resultat" ></output>
```

Choisir une couleur

HTML5 prévoit également un composant destiné à recevoir l'encodage d'une couleur (en hexadécimal) avec la balise `<input type="color">`.

En cliquant sur ce composant, selon le support du navigateur, une palette de couleur apparaîtra automatiquement pour choisir une couleur et la valeur de celle-ci serait reportée.

```
<input type="color" name="couleurfav" value="#ff0000">
```

Le composant de recherche

Il est courant de trouver sur les sites un outil de recherche. Celui-ci prend souvent la forme d'une ligne de texte. Avec HTML5, un composant spécialement dédié à cette possibilité de recherche est proposé avec la balise `<input type="search">`. Sa syntaxe est :

```
<input type="search" name="recherche" >
```

La saisie d'une ligne de texte avec liste de suggestions

Afin de fournir une fonction "autocomplete" pour les éléments de saisie de texte `<input>`, HTML5 introduit la balise `<datalist>`. Il spécifie une liste d'options prédéfinies pour un élément `<input>`.

Les utilisateurs voient une liste déroulante des options prédéfinies au fur et à mesure qu'ils entrent des données. Ceci est proche de ce que "Google Suggest" présente lors de l'entrée d'un mot-clé dans sa barre de recherche.

La syntaxe de cette balise est :

```
<input list="navigateurs" />
<datalist id="navigateurs">
  <option value="Chrome">
  <option value="Firefox">
  <option value="Internet Explorer">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Les boutons radio et les cases à cocher

En plus des composants de saisie de texte, HTML5 propose des éléments de formulaires spéciaux nommés boutons radio et cases à cocher. Ils permettent d'encoder des données prévisibles et de nombre limité.

Les boutons radio ont comme particularité qu'une seule option à la fois peut être activée.

Pour créer un bouton radio, on utilise l'élément `<input />` dont l'attribut `type` prend cette fois la valeur `radio`. L'ensemble des boutons radio avec lesquels on peut opérer un choix donné constitue un groupe. Il faut que tous les attributs `name` de ses éléments aient la même valeur. Pour cocher par défaut un des boutons radio un groupe, on peut utiliser l'attribut `checked`.

```
<input type="radio" name="taille" value="S" > S
<input type="radio" name="taille" value="M" checked > M
<input type="radio" name="taille" value="L"> L
<input type="radio" name="taille" value="XL"> XL
```

Pour des données où c'est possible de sélectionner plusieurs choix simultanément, on peut ne pas utiliser les boutons radio. Les cases à cocher représentent la meilleure solution. Leur fonctionnement semble identique au boutons radio, à la différence que les cases à cocher ne font pas partie d'un groupe, et qu'il est possible d'opérer plusieurs choix dans des cases différentes sans que le choix précédent se trouve décoché. Une case à cocher est encore créée à l'aide de l'élément `<input />` dont l'attribut `type` prend cette fois la valeur `checkbox`.

```
<input type="checkbox" name="n1" value="Foot" checked >Foot
<input type="checkbox" name="n2" value="Basketball">Basketball
```

```
<input type="checkbox" name="n3" value="Tennis">Tennis  
<input type="checkbox" name="n4" value="Ping Pong">Ping Pong  
<input type="checkbox" name="n5" value="Natation">Natation
```

Les listes de sélection

Un autre moyen de faciliter la saisie de données sur un site web consiste à proposer au visiteur d'effectuer un ou plusieurs choix parmi une liste de sélection déroulante pouvant contenir un grand nombre d'éléments. Ces listes peuvent être utilisées quand l'éventail des réponses est prévisible, comme une liste de pays ou de types de cartes bancaires. Cette méthode a l'avantage d'empêcher les fautes d'orthographe que pourraient commettre les différents utilisateurs.

```
<select name="navigateurs">  
  <option value="2">Firefox</option>  
  <option value="5">Google Chrome</option>  
  <option value="1">Internet Explorer</option>  
  <option value="4">Opera</option>  
  <option value="3">Safari</option>  
</select>
```

Pour structurer les options d'une liste déroulantes et leur donner plus de visibilité particulièrement si la liste est longue, il est possible de grouper les éléments liés via des groupes d'options `<optgroup>` et de leur attribuer un titre. La syntaxe de la balise

`<optgroup>` est le suivant :

```
<select name="pays">  
  <optgroup label="Europe">  
    <option value="2">France</option>  
    <option value="5">Belgique</option>  
    <option value="1">Espagne</option>  
  </optgroup>  
  <optgroup label="Afrique">  
    <option value="2">Maroc</option>  
    <option value="5">Algérie</option>  
    <option value="1">Tunisie</option>  
  </optgroup>  
</select>
```

Les champs cachés

Un formulaire peut aussi contenir des champs cachés. Ils permettent de transmettre au serveur des informations particulières et ainsi de recueillir des données sur le poste client tel que le système d'exploitation, le type de navigateur, sa version. Ces champs servent également à communiquer tout type d'informations utiles au serveur, comme la taille maximale d'un fichier téléchargeable du poste client vers le serveur.

Pour créer un champ caché, on utilise encore l'élément `<input />`. L'attribut `type` reçoit la valeur `hidden`, et les attributs `id` et `name` servent à identifier le champ et à recueillir l'information dans une variable. L'attribut `value` contient la valeur qui sera transmise après l'envoi du formulaire. Pour créer un champ caché, on peut écrire le code suivant :

```
<input type="hidden" id="c" name="c" value="cache">
```

Le transfert de fichiers

La balise `<input type="file">` prend en charge le transfert de fichiers (file) du poste de l'utilisateur vers un ordinateur de type serveur.

Pour effectuer le transfert de fichier, l'élément `<form>` doit impérativement utiliser la méthode poste et posséder un attribut `enctype` dont la valeur est `multipart/form-data` :

```
<form action="traitement.php" method="post" enctype="multipart/form-data">
```

La syntaxe de cette balise est la suivante :

```
<input type="file" name="fichier" >
```

En plus des attributs communs, il y a aussi l'attribut `accept` qui permet de limiter le transfert à certains types de fichiers. On peut citer des fichiers texte (txt), Word (doc), Excel (xls), pdf, des fichiers image (jpeg, gif ou png), etc. Dans la désignation des fichiers, le joker `*` est accepté. Par exemple, pour accepter juste le transfert des fichiers images, on peut écrire le code suivant :

```
<input type="file" name="fichier" accept="image/*" >
```

Il faut noter que cette balise HTML ne sert qu'à sélectionner le fichier à transférer. Le transfert lui-même doit être pris en charge par des applications côté serveur comme par exemple du code PHP.

L'organisation des champs de formulaires

Dans le cas de formulaires longs et complexes, il est parfois utile de regrouper graphiquement certains éléments pour organiser la page de façon logique. Les balises `<fieldset>` et `<legend>` permettent d'améliorer sensiblement l'ergonomie et l'usabilité des formulaires.

La balise `<fieldset> ... </fieldset>` englobe les champs de formulaires que vous déterminez. Ces champs sont alors visualisés à l'écran par une bordure.

La balise `<legend> ... </legend>`, qui se place directement derrière la balise `<fieldset>`, ajoute une légende qui vient s'insérer dans la bordure dessinée par le `<fieldset>`.

Voici un exemple d'utilisation des balises `<fieldset>` et `<legend>` :

```
<form action="process.php">
  <fieldset>
    <legend>Nos entrées</legend>
    <input type="checkbox" name="n1"> Salade niçoise<br>
    <input type="checkbox" name="n2"> Salade du chef<br>
    <input type="checkbox" name="n3"> Légumes grillés<br>
  </fieldset>
  <fieldset>
    <legend>Nos pizzas</legend>
    <input type="checkbox" name="n4"> Margarita<br>
    <input type="checkbox" name="n5"> 4 saisons<br>
    <input type="checkbox" name="n6"> Fruits de mer<br>
  </fieldset>
  <fieldset>
    <legend>Nos desserts</legend>
    <input type="checkbox" name="n7"> Glaces variées<br>
    <input type="checkbox" name="n8"> Tiramisu<br>
    <input type="checkbox" name="n9"> Moelleux au chocolat<br>
  </fieldset>
</form>
```

```
<input type="submit">
<input type="reset">
</form>
```

L'étiquetage des champs de formulaires

La balise `<label>` associe explicitement l'intitulé à un champ de formulaire particulier. Un peu comme si on collait une étiquette (label) en face d'un élément de formulaire.

Dans un premier temps, la balise `<label>` améliore grandement l'ergonomie des formulaires en permettant d'activer un élément de celui-ci.

Par exemple un bouton radio, en cliquant sur le bouton radio lui-même ou sur l'intitulé de celui-ci.

Mais les balises `<label>` sont aussi particulièrement utiles dans le domaine de l'accessibilité des sites Web aux personnes non voyantes. Ces "étiquettes" sont prises en charge par les aides techniques (barrettes braille ou synthèses vocales) et facilitent grandement l'utilisation des formulaires par les personnes visuellement déficientes.

Dans un premier temps, le texte assigné à un élément de formulaire doit être placé entre les balises `<label>...</label>`.

```
<label>Nom</label> :
<input type="text"><br>
```

Il faut ensuite relier cette étiquette label au contrôle de formulaire. Pour y faire, l'élément de formulaire sera défini par un identifiant de type id.

```
<label>Nom</label> :
<input type="text" id="f1"><br>
```

L'attribut `for="..."` ajouté à la balise `<label>` permet de raccorder directement l'étiquette au champ de formulaire en faisant référence à cet identifiant.

```
<label for="f1">Nom</label> :
<input type="text" id="f1"><br>
```

Voici un exemple d'utilisation de la balise `<label>` :

```
<form action="process.asp">
  <p> Paiement par : <br>
    <label for="cash1"> Visa</label><input type="radio" name="cash" id="cash1"><br>
    <label for="cash2"> American Express</label><input type="radio" name="cash" id="cash2"><br>
    <label for="cash3"> Mastercard</label><input type="radio" name="cash" id="cash3">
  </p>
</form>
```