

Partie 1 : HTML

Ce cours complet sur le HTML va vous permettre d'en apprendre les bases conformément aux recommandations et normes en vigueur établies par le W3C. Il répondra à quasiment toutes les questions que vous pouvez vous poser sur ce langage.

Des remarques ? Des questions ? N'hésitez pas.

I - Introduction

I-A - Introduction au HTML

I-A-1 - Qu'est-ce que le HTML ?

Le **HTML** (HyperText Markup Language) est un format de présentation de données permettant de créer des

pages web pouvant être lues dans des **navigateurs**.

C'est un langage de description de données, et non un **langage de programmation**. Il est figé c'est-à-dire qu'une fois le document chargé dans le navigateur, il ne répond à aucune action de l'utilisateur sur le contenu de la page..

I-A-2 - Qu'est-ce qu'une balise HTML ?

Une balise HTML est un élément que l'on va ajouter au texte de départ pour dire au navigateur de quelle manière l'afficher. Elle n'est pas affichée telle quelle dans le navigateur, elle est interprétée par celui-ci.

Elle est toujours délimitée par les signes < et >.

Exemple de balise

```
<html>
```

Une balise peut en plus comporter de zéro à plusieurs attributs. Les attributs sont des informations complémentaires qui la caractérisent. Ils se présentent sous la forme **nomattribut="valeur"**.

Exemple de balise et son attribut

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

Chaque balise ouverte doit être fermée, cependant il existe des exceptions. On distingue deux types de balises :

- Les balises simples.
- Les balises doubles.

I-A-2-a - Les balises simples

Ce sont des balises qui sont dites "vides", c'est-à-dire qu'elles ne vont contenir aucune autre balise HTML. Ces balises n'ont pas besoin d'être fermées.

Exemple de balise simple

```

```

I-A-2-b - Les balises doubles

Les balises doubles sont dites ouvrantes/fermantes, c'est-à-dire qu'elles nécessitent deux balises, une ouvrante et une fermante dans lesquelles on va pouvoir mettre d'autres balises ou du texte.

La balise fermante est identique à la balise ouvrante, à la différence qu'elle contient un "/" pour indiquer à quel endroit on la ferme.

Exemple de balise double

```
<p>Ici du texte ou tout autre balise.</p>
```

I-A-2-c - Imbrication des balises

Lorsque l'on cumule l'écriture de plusieurs balises ouvrantes (forcément inévitable), il est impératif de respecter la hiérarchie des balises, c'est-à-dire que la première ouverte sera la dernière à être fermée.

Le chevauchement des balises est une grave erreur.

Exemple de balises imbriquées

```
<b><u>Ici les balises sont correctement imbriquées.</u></b>
```

Exemple de balises chevauchées

```
<b><u>Ici les balises ne sont</b> pas correctement imbriquées, c'est une erreur.</u>
```

I-B - Structure d'un document HTML

Avant de commencer tout développement, il est important de connaître la structure d'une page HTML. Commençons donc par voir celle-ci avec la structure minimale obligatoire d'une page qui sera notre base de travail tout au long de ce tutoriel :


Document HTML de base

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Titre de la page</title>
  <meta http-equiv="content-type" content="text/html;
  charset=utf-8">
</head>
<body>

<!-- Ici votre site -->

</body>
</html>
```

La première ligne du document s'appelle le DOCTYPE. C'est une balise indispensable pour conserver la compatibilité du rendu de votre page sur les différents navigateurs modernes. Après le DOCTYPE, vient la balise `<html>`. Elle encadre l'ensemble des autres balises. La balise `<html>` contient deux grandes balises : `<head>`, qui va contenir toute l'en-tête de la page et `<body>` qui va contenir tout le corps du document.

 Si vous incluez des pages à l'aide d'un langage serveur (PHP, ASP, JSP...), ayez toujours en tête que le DOCTYPE, la balise `<html>` ainsi que les balises `<head>` et `<body>` sont uniques dans une page. Si tel n'est pas le cas dans votre code source, vous aurez de mauvaises surprises à l'affichage de votre site.

I-C - Le DOCTYPE

À quoi sert ce fameux DOCTYPE ? Est-il vraiment utile ? Oui, il est utile, même très utile.

Il sert d'une part à déclarer quel "langage", ou norme on applique à notre page. Cette déclaration nous permet de soumettre nos pages au **validateur** du **W3C** (World Wide Web Consortium) qui vérifie que notre code HTML ne comporte aucune erreur de syntaxe (balises mal fermées, utilisation d'attributs inexistants...) en fonction du DOCTYPE, donc de la **DTD** (Document Type Definition) choisie. D'autre part, il indique au navigateur de quelle manière interpréter telle balise, telle propriété CSS. L'absence de cette balise est appelée le mode Quirks, ou non conforme, alors qu'au contraire, la présence de cette balise est appelée le mode Standard, ou conforme.

I-D - Les commentaires

Les commentaires permettent de laisser des informations ou des descriptions du code et sont utilisables à n'importe quel endroit de votre document. Cependant, ce ne sont pas ces commentaires-là qu'il faudra utiliser pour les langages JavaScript et CSS qui possèdent leur propre système de commentaires.

Ces commentaires ne seront pas affichés dans le navigateur mais ils resteront présents dans le code source, donc il faut à tout prix éviter de mettre des informations confidentielles dedans.

Ils sont délimités par `<!--` et `-->` et peuvent être multilignes.

Commentaires HTML

```
<!-- Voici un premier commentaire sur une ligne. -->
```

```
<!-- Voici un deuxième commentaire.  
      Celui-ci est multilignes. -->
```

```
<!------- Cette méthode est déconseillée ! -----  
>
```

Commentez l'article :

II - Le corps du document - body

C'est dans la balise `<body>` que nous allons mettre tout le texte et toutes les images qui apparaîtront lorsqu'un visiteur viendra sur notre page.

II-A - Les niveaux de titres

Il existe différents niveaux de titres, ils vont de 1 à 6 et ont chacun leur importance. La balise représentant ces niveaux de titre est `<hX>` où X représente le niveau.

Exemple :

Exemple de titre

```
<h1>Titre de ma page</h1>
```

Il est important de respecter la hiérarchie des différents niveaux de titres, sachant que l'élément le plus haut est le

`<h1>`. En effet, un titre `<h2>` ne se retrouvera jamais au-dessus d'un `<h1>` dans un document. Le titre `<h1>` étant le plus important, il convient également de ne le faire apparaître qu'une seule fois dans un document.

II-B - Les paragraphes de texte

Comme vous tapez du texte dans un traitement de texte, vous segmentez votre texte en paragraphes. En HTML c'est la même chose, sauf qu'il faut préciser nous-même où commence un paragraphe et où il se termine. C'est ce à quoi sert la balise `<p>`.

Les paragraphes de texte en HTML

`<p>Voici un premier paragraphe.</p>`

`<p>Voici un deuxième paragraphe.</p>`

II-C - La mise en forme de texte

II-C-1 - Le texte en gras

Pour mettre le texte en gras on l'encadre de la balise `` :

Texte en gras

`Ce texte s'affichera en gras.`

II-C-2 - Le texte en italique

Pour mettre le texte en italique on l'encadre de la balise `<i>` :

Texte en italique

`<i>Ce texte s'affichera en italique.</i>`

II-C-3 - Le texte souligné

Pour souligner le texte on l'encadre de la balise `<u>` :

Texte souligné

`<u>Ce texte sera souligné.</u>`

II-C-4 - Le texte barré

Pour barrer le texte on l'encadre de la balise `<s>` :

Texte barré

`<s>Ce texte sera barré.</s>`

II-C-5 - Le texte en exposant

Pour placer le texte en exposant on l'encadre de la balise `<sup>` :

Texte en exposant

```
<sup>Ce texte sera en exposant.</sup>
```

II-C-6 - Le texte en indice

Pour placer le texte en indice on l'encadre de la balise `<sub>` :

Texte en indice

```
<sub>Ce texte sera en indice.</sub>
```

II-C-7 - Modifier la couleur du texte

Pour modifier la couleur du texte on utilise l'attribut `color` de la balise `` :

Texte en rouge

```
<font color="#ff0000">Ce texte sera  
en rouge.</font>
```

Les couleurs peuvent être écrites de deux manières :

- En hexadécimal de type RVB et précédées d'un dièse (#) ; Exemples : #ff0000 => rouge, #00ff00 => vert, #0000ff => bleu.
- Textuelles en anglais US ; Exemples : red, yellow, pink.

II-C-8 - Modifier la police du texte

Pour modifier la police du texte on utilise l'attribut `face` de la balise `` :

Texte en verdana

```
<font face="verdana">Ce texte sera en verdana.</font>
```

On a tendance à écrire des familles de polices plutôt qu'une seule police. En effet, il se peut que sur un système différent du vôtre la police choisie ne soit pas installée et il faut qu'il puisse alors afficher le texte dans une police proche de celle que vous vouliez avoir.

On sépare alors les différentes polices par une virgule dans l'ordre de sélection le cas où la précédente ne peut pas être affichée.

Texte en verdana ou sans-serif

```
<font face="verdana,sans-serif">Ce texte sera en verdana ou en  
sans-serif si verdana n'est pas installée.</font>
```


II-C-9 - Modifier la taille du texte

Pour modifier la taille du texte on utilise l'attribut **size** de la balise `` :

Texte en taille 5

```
<font size="5">Ce texte sera en taille 5.</font>
```

Par défaut, la valeur de l'attribut **size** vaut "3".

Il y a deux manières de modifier la valeur de cet attribut : et les valeurs possibles sont les entiers de "1" à "7".

Et il est bien évidemment possible de renseigner ces trois attributs (**color**, **face**, **size**) à la fois dans la même balise ``.

II-D-1 - Les images

Pour afficher une image en HTML, on utilise la balise ``. Elle admet différents attributs, dont les plus importants sont **src** et **alt**.

L'attribut **src** permet de renseigner le chemin de l'image, qui peut être **absolu** HTTP ou **relatif**, très utile pour pouvoir l'afficher.

L'attribut **alt** affiche un texte alternatif à la place de l'image dans le cas où, pour une raison ou autre, celle-ci ne peut pas être chargée. Il aide de plus les moteurs de recherche à référencer l'image.

Afficher une image

```

```

⚠ L'attribut **alt** n'est pas censé créer une infobulle lors du survol de l'image, qui est le rôle de l'attribut **title**, cependant sur Internet Explorer le **alt** affiche cette infobulle.

Les différents formats d'image supportés par cette balise sont :

- GIF
- JPG
- PNG
- BMP (à éviter)

II-E - Les liens hypertextes

Un lien hypertexte est un élément HTML permettant d'envoyer le visiteur vers une nouvelle page. On peut insérer dans cet élément toute sorte de textes, images... Un lien sera représenté dans le code par la balise `<a>`. Tout ce qui sera dans cette balise fera office de lien. L'adresse de destination doit se trouver dans l'attribut **href**.

Exemple texte :

Exemple d'un lien texte

```
<a href="adresse/de/destination.html">Texte du lien</a>
```

Exemple image :

Exemple d'un lien image

```
<a href="adresse/de/destination.html"></a>
```

On distingue deux types de liens, les liens externes et les liens internes.

II-E-1 - Les liens externes

Un lien externe est un lien hypertexte qui renvoie vers une autre page. Cette autre page peut être une page de votre site ou celle d'un site tiers.

II-E-2 - Les liens internes

Un lien interne est un lien qui renvoie vers la même page, mais pas forcément au même endroit de la page.

Exemple d'un lien vers un autre cadre

```
<a href="adresse/de/destination.html" target="valeur">Lien de mon  
menu</a>
```

Il existe plusieurs valeurs pour l'attribut target :


Valeur	Description
_self	Affichage dans le cadre courant (par défaut).
_blank	Ouvre le lien dans une nouvelle fenêtre.

II-E-3 - Les liens e-mails

Vous pouvez permettre à vos visiteurs d'écrire sur votre adresse e-mail avec la syntaxe "**mailto:**" :

Exemple d'un lien vers une adresse e-mail

```
<a href="mailto:jwillette@monsite.com">Ecrivez-moi</a>
```

 Ce type de liens ouvre le logiciel de messagerie par défaut de vos visiteurs.

Vous pouvez enfin mettre plusieurs destinataires :

Exemple d'un lien vers plusieurs adresses e-mail

```
<a href="mailto:destinataire1 @monsite.com;destinataire2 @monsite.com">Ecrivez-nous</a>
```

Exemple d'un lien vers une adresse e-mail et une autre en copie

```
<a href="mailto:destinataire1 @monsite.com?cc=destinataire2 @monsite.com">Ecrivez-nous</a>
```

II-E-5 - Informations complémentaires

- Pour éviter d'avoir le cadre bleu autour d'une image lorsqu'elle est dans un lien, il vous suffit de renseigner l'attribut **border** à "0" :
- Si vous voulez avoir une infobulle qui s'affiche sur votre lien lors du passage de la souris, vous pouvez renseigner l'attribut **title** :

Infobulle sur le lien

```
<a href="adresse/de/destination.html" title="Description de la page cible">Texte du lien</a>
```

Exemple de cadre local

```
<iframe frameborder="0" > Désolé, votre navigateur ne supporte  
pas les cadres locaux  
</iframe>
```

II-G - Les listes

Elles servent, comme leur nom l'indique, à créer une liste d'items, plus précisément à les énumérer, comme dans le cas d'un sommaire par exemple.

II-G-1 - Les différents types de listes

On distingue plusieurs types de listes différents, chacun prévu pour un usage bien précis :

- Les listes non ordonnées, ou à puces.
- Les listes ordonnées.
- Les listes de menus.

II-G-1-a - Les listes non ordonnées

Elles sont démarquées par la balise `` et chacun des items est entouré de la balise `` :

Exemple de liste non ordonnée

```
<p>Ma liste de courses :</p>  
<ul>  
  <li>Champignons</li>  
  <li>Avocat</li>  
  <li>Dentifrice</li>  
  <li>Céréales</li>  
</ul>
```

II-G-1-a-i - Les différents types de puces

On va pouvoir jouer sur le design des puces à l'aide de trois types différents. On renseigne le type des puces avec l'attribut `type` de la balise ``.

Les types sont :

- **disc** (par défaut au niveau 1 d'imbrication)
- **circle** (par défaut au niveau 2 d'imbrication)
- **square** (par défaut au niveau 3 et suivants d'imbrication)

Dans l'exemple précédent, nous avons donc vu une liste non ordonnée avec une puce de type disc. Voici à quoi ressemblent les deux autres :

Exemple de liste non ordonnée

```

<p>Ma liste de
courses :</p>
<ul type="circle">
  <li>Champignons</li>
  <li>Avocat</li>
  <li>Dentifric
    e</li>
  <li>Céréales</li>
</ul>

```

Exemple de liste non
ordonnée

```

<p>Ma liste de courses :</p>
<ul type="square">
  <li>Champignons</li>
  <li>Avocat</li>
  <li>Dentifrice</li>
  <li>Céréales</li>
</ul>

```

Résultat

:

Ma liste de courses :

- Champignons
- Avocat
- Dentifrice
- Céréales

II-G-1-b - Les listes ordonnées

Elles sont introduites par la balise `` et chacun des items est encadré par la balise `` :

Exemple de liste ordonnée

```
<p>Top 3 du tiercé :</p>
<ol>
  <li>Jolly</li>
  <li>Tonnerre</li>
  <li>Tornade</li>
</ol>
```

II-G-1-b-i - Les différents types de numérotation

Il existe cinq manières différentes de numéroter les liste ordonnées. On renseigne le type de numérotation avec l'attribut **type** de la balise ``.

Les types sont :

- **1** : Numérotation par les chiffres arabes (par défaut).
- **I** : Numérotation par les chiffres romains.
- **i** : Numérotation par les chiffres romains minuscules.
- **A** : Numérotation par les lettres de l'alphabet.
- **a** : Numérotation par les lettres de l'alphabet minuscules.

Exemple de liste ordonnée

```
<p>Top 3 du
tiercé :</p>
<ol type="I">
  <li>Jolly
  </li>
  <li>Tonnerre</li>
  <li>Tornade
  </li>
</ol>
```

Exemple de liste ordonnée

```
<p>Top 3 du
tiercé :</p>
<ol type="i">
  <li>Jolly</li>
  <li>Tonnerre</li>
  <li>Tornade
  </li>
</ol>
```

Exemple de liste ordonnée

```
<p>Top 3 du
tiercé :</p>
<ol type="A">
  <li>Jolly
  </li>
  <li>Tonnerre</li>
  <li>Tornade
  </li>
</ol>
```

II-G-1-e - Les listes de menus

Elles sont représentées par la balise `<menu>` et chacun des items est délimité par la balise ``.

Exemple de liste de menus

```
<menu>
  <li>Accueil</li>
  <li>Page 1</li>
  <li>Page 2</li>
</menu>
```

Résultat

:

- Accueil
- Page 1
- Page 2

II-G-2 - Les listes imbriquées

L'imbrication des listes permet une présentation claire et détaillée d'un menu ou d'un sommaire par exemple. On peut imbriquer les listes non ordonnées et ordonnées sans distinction et même les mélanger :

Exemple de listes imbriquées

```
<ol type="I">
  <li>La genèse</li>
  <li>La vie
    <ol type="A">
      <li>L'enfance</li>
      <li>L'adolescence</li>
      <li>Le passage à l'âge adulte
        <ul>
          <li>Le jeune adulte</li>
          <li>La crise de la quarantaine</li>
        </ul>
      </li>
      <li>La vieillesse</li>
    </ol>
  </li>
  <li>La mort</li>
</ol>
```

Résultat :

- I. La genèse
- II. La vie
 - A. L'enfance
 - B. L'adolescence
 - C. Le passage à l'âge adulte
 - Le jeune adulte
 - La crise de la quarantaine
 - D. La vieillesse
- III. La mort

II-G-3 - L'utilité des listes

La principale utilisation des listes de nos jours se fait par les listes non ordonnées pour la création de divers menus. L'imbrication de ces listes non ordonnées couplée avec un soupçon de CSS permet de faire des menus à multiples niveaux d'affichage, comme les menus déroulants.

II-H - Les tableaux

II-H-1 - À quoi servent-ils ?

Un tableau qu'est-ce que c'est ? Tout simplement une suite de lignes et de colonnes qui forment un ensemble de cellules.

Les tableaux servent avant tout à présenter des données dont voici un exemple :

Elèves	Français	Math	Histoire	Sport
Laure	14	10	11	11
Philippe	8.5	12	9	15
Jean	15	14.	10.	12
Maurice	7.5	18	12	14

II-H-2 - Du tableau simple au tableau complexe

II-H-2-a - Les bases d'un tableau

Un tableau est délimité par la balise `<table>`. Pour créer un tableau, nous commençons par créer des lignes grâce aux balises `<tr>`. Puis nous insérons dans ces lignes les cellules du tableau grâce aux balises `<td>`. Le contenu (textes, images, autres balises) se trouve uniquement dans les balises `<td>` et ne doit se trouver **en aucun cas** en dehors.

Donc pour avoir un tableau d'une cellule, soit une ligne et une colonne, on obtient ce code :

Tableau d'une cellule

```
<table border="1">
  <tr>
    <td>Contenu de la cellule</td>
  </tr>
</table>
```

Et pour un tableau de deux lignes et deux colonnes :

Tableau de deux lignes et deux colonnes

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

Par défaut, la bordure n'est pas visible, dans les codes précédents on a rajouté l'attribut **border** à la balise `<table>`

pour la faire apparaître. On peut évidemment épaissir la bordure en augmentant la valeur.

On peut voir sur les tableaux affichés précédemment que le texte n'est pas directement collé à la bordure. Il s'agit là d'un espace interne par défaut des navigateurs. Pour le supprimer, on doit renseigner l'attribut **cellpadding** et lui mettre la valeur "20".

Si on reprend tout ce dont on vient de parler, on aurait un code comme :

Récapitulatif des attributs de la balise table

```
<table border="1" cellpadding="20" >
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
</table>
```

II-H-2-b - Le titre d'un tableau

Pour décrire le contenu d'un tableau, par exemple expliquer dans notre tout premier exemple de tableau qu'il s'agit des notes de la classe de 3^{ème}B, il faut utiliser la balise `<caption>`, à placer directement après l'ouverture du tableau. Cette balise prend l'attribut **align** qui accepte comme valeurs "**top**" (par défaut), "**bottom**", "**left**" et "**right**", en fonction de l'endroit où l'on veut afficher cette légende.

Exemple d'un titre de tableau

```
<table border="1">
  <caption align="left">Notes de la classe de 3ème B</caption>
  <tr>
    <td>Elèves</td>
    <td>Français</td>
    <td>Mathématiques</td>
    <td>Histoire</td>
    <td>Sport</td>
  </tr>
  <tr>
    <td>Laure</td>
    <td>14</td>
    <td>10</td>
    <td>11</td>
    <td>11</td>
  </tr>
</table>
```

II-H-2-c - La ligne d'en-tête d'un tableau

Pour reprendre l'exemple des notes de la classe de 3èmeB, on remarque que la première ligne peut faire office d'en- tête. En effet, elle renseigne le contenu de chacune des colonnes et n'a donc pas le même genre de contenu que tout le reste du tableau.

Pour préciser au navigateur qu'il s'agit donc de cellules spéciales, on va les modifier en remplaçant pour cette ligne seulement les balises `<td>` par des balises `<th>` :

Utilisation de la balise th

```
<table border="1">
  <caption align="bottom">Notes de la classe de 3èmeB</caption>
  <tr>
    <th>Elèves</th>
    <th>Français</th>
    <th>Mathématiques</th>
    <th>Histoire</th>
    <th>Sport</th>
  </tr>
  <tr>
    <td>Laure</td>
    <td>14</td>
    <td>10</td>
    <td>11</td>
    <td>11</td>
  </tr>
</table>
```

II-H-2-e - La fusion de lignes et de colonnes d'un tableau

On distingue deux types de fusion : la fusion horizontale où sont en jeu deux (ou plus) colonnes, la fusion verticale où sont en jeu deux (ou plus) lignes. Dans les deux cas, elle s'opère sur les cellules, donc sur la balise `<td>`. Elle se traduit par l'utilisation de l'attribut **colspan** dans le cas d'une fusion horizontale et par l'attribut **rowspan** dans le cas d'une fusion verticale. Ces attributs prennent pour valeurs le nombre de cellules à fusionner.

Exemple d'utilisation de colspan

```
<table border="1">
  <tr>
    <td colspan="5">Cellule qui se répand sur 5 colonnes</td>
  </tr>
  <tr>
    <td>Colonne 1</td>
    <td>Colonne 2</td>
    <td>Colonne 3</td>
    <td>Colonne 4</td>
    <td>Colonne 5</td>
  </tr>
</table>
```

En ce qui concerne l'attribut **rowspan**, on aurait un code tel que :

Exemple d'utilisation de rowspan

```
<table border="1">
  <tr>
    <td rowspan="2">Cellule sur deux lignes</td>
    <td>Ligne 1 - Colonne 2</td>
    <td>Ligne 1 - Colonne 3</td>
    <td>Ligne 1 - Colonne 4</td>
  </tr>
  <tr>
    <td>Ligne 2 - Colonne 2</td>
    <td>Ligne 2 - Colonne 3</td>
    <td>Ligne 2 - Colonne 4</td>
  </tr>
</table>
```

Il est tout de suite beaucoup plus difficile de se rendre compte de son erreur, le **rowspan** est déjà moins intuitif que le **colspan**.

On pourrait aller plus loin en combinant les deux techniques sur un seul et même tableau :

Exemple d'utilisation de colspan et rowspan

```
<table border="1">
  <tr>
    <td colspan="3">Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td rowspan="3">Cellule 3</td>
    <td>Cellule 4</td>
    <td>Cellule 5</td>
    <td>Cellule 6</td>
  </tr>
  <tr>
    <td>Cellule 7</td>
    <td colspan="2" rowspan="2">Cellule 8</td>
  </tr>
  <tr>
    <td>Cellule 9</td>
  </tr>
</table>
```

On remarque que le code est devenu extrêmement compliqué et que la moindre erreur de fusion de notre part entraînerait une interprétation du tableau complètement bancal par les navigateurs.

II-I - Les formulaires

II-I-1 - À quoi servent-ils ?

Les formulaires servent à envoyer des données au serveur, données remplies en général par un visiteur, que ce soit pour l'inscription sur un site ou un formulaire de contact par exemple. Les formulaires sont dotés de divers contrôles comme des champs de saisie, des boutons, des listes, etc, qui permettent au visiteur d'interagir avec la page qu'il est en train de consulter.

II-I-2 - La balise form

Tous les champs de formulaires quels qu'ils soient doivent se trouver dans cette balise `<form>`.

C'est cette balise qui va permettre de renseigner la page de destination du formulaire, à l'aide de l'attribut `action`.

Elle est aussi utilisée pour préciser le mode d'envoi des données. C'est avec l'attribut `method` que l'on va le préciser. Cet attribut prend deux valeurs différentes qui sont `"get"` ou `"post"`. Les valeurs envoyées en "GET" passent par l'URL alors qu'en "POST" elles sont envoyées de manière transparente, aussi, on préférera l'utilisation de la méthode d'envoi "POST".

Voyons maintenant à quoi pourrait ressembler notre balise `<form>` :

Exemple de balise form

```
<form action="envoi-formulaire.html" method="post">
  <!-- Contenu du formulaire -->
</form>
```

II-I-3 - Les champs texte monolignes

On distingue deux types de champs texte monolignes : les champs de type "texte" et les champs de type "mot de passe".

Le champ de type "texte" est l'élément de base d'un formulaire, il permet d'entrer du texte court sur une ligne, comme un login, un e-mail, ou autres alors que le champ de type "mot de passe" permet de renseigner un mot de passe, dont l'affichage sera camouflé aux yeux des autres personnes.

Le champ de texte monoligne est caractérisé par la balise `<input>` et l'attribut `type` dont la valeur sera `"text"` ou

`"password"`.

Voici un exemple de formulaire


```
<form action="gcheicky2005(@yahoo.fr" method="post">
<div>
<label >Nom :</label>
<input type="text" id="nom" />
</div>
<div>
<label>Courriel :</label>
<input type="email" id="courriel" />
</div>
<div>
<label >Message :</label>
<textarea id="message"></textarea>
</div>
<div>
<button type="submit">Envoyer message</button>
<button type="reset" >Effacer</button>
</div>
</form>
```

On emploie la méthode "get" lorsque le formulaire ne produit aucun effet secondaires. Nombre de recherches dans les bases de données n'ont pas d'effets secondaires visibles.

Si le service associé au traitement d'un formulaire entraîne des effets secondaires (par exemple, si le formulaire modifie une base de données ou l'abonnement à un service), on devrait alors employer la méthode "post".

II-I-4 - Les champs texte multilignes

Si vous souhaitez que vos visiteurs puissent saisir de longs textes, comme des champs de commentaires pour des articles d'un blog par exemple, il faudra se tourner vers l'utilisation de la balise `<textarea>`.

Contrairement au champ de texte monoligne, le `textarea` est une balise double et son contenu est écrit entre les balises ouvrante et fermante.

II-I-6 - Les champs de fichiers

Pour permettre aux visiteurs d'envoyer des fichiers par l'intermédiaire du formulaire, que ce soit en pièce jointe d'un e-mail ou le chargement d'une image sur le serveur, on va utiliser la balise `<input>` dont l'attribut **type** est renseigné à **"file"**.

Ceci fait, un champ avec un bouton "Parcourir" sera disponible.

```
<FORM action=http://server.com/cgi/gestion
method="post">
  <label >Nom :</label>
  <INPUT type="text" ><BR>
  <label >envoi de fichier :</label>
  <INPUT type="file" > <BR>
  <INPUT type="submit" value="Envoyer">
  <INPUT type="reset">
</FORM>
```

Mais pour que le champ soit opérationnel, il faut impérativement renseigner l'attribut **enctype** de la balise `<form>` à **"multipart/form-data"**.

Il est aussi plus qu'utile de préciser l'attribut **name** du champ "file".

II-I-7 - Les boutons radio

Les boutons radio sont utilisés lorsque vous voulez laisser aux visiteurs un choix et un seul parmi une liste de propositions. La balise est alors `<input>` dont l'attribut **type** est renseigné à **"radio"**.

La valeur de l'attribut **value** va être transmise au serveur en fonction du bouton choisi.

Exemples de groupes de boutons radio

```
<p>
  <input type="radio" name="civilité" value="mlle"> Mademoiselle
  <input type="radio" name="civilité" value="mme"> Madame
</p> <input type="radio" name="civilité" value="mr"> Monsieur
```

II-I-8 - Les cases à cocher

Les cases à cocher sont sensiblement identiques aux boutons radio, mais permettent un choix multiple pour un groupe de propositions. Les attributs sont identiques, à la différence que l'attribut **type** prend pour valeur **"checkbox"**.

Exemple de cases à cocher

```
<input type="checkbox" name="qualite" value="intelligent"> Intelligent  
<input type="checkbox" name="qualite" value="beau"> Beau  
<input type="checkbox" name="qualite" value="serviable"> Serviable  
<input type="checkbox" name="qualite" value="fort"> Fort  
<input type="checkbox" name="qualite" value="généreux"> Généreux
```

II-I-9-b - Le bouton d'envoi

Le bouton d'envoi sert, comme son nom l'indique, à envoyer les données du formulaire au serveur. C'est celui qui va permettre de valider le formulaire.

Il est caractérisé par la valeur "**submit**" de l'attribut **type** des balises `<input>` et `<button>`. Exemples de boutons d'envoi :

Exemples de boutons d'envoi

```
<input type="submit" name="envoyer" value="Valider">  
<button type="reset" name="effacer">effacer</button>
```

II-I-10 - Les listes

Les listes permettent de laisser un choix aux visiteurs, au même titre que les boutons radio ou les cases à cocher, mais sont plus pratiques lorsqu'il s'agit de proposer un grand nombre de choix.

On distingue deux types de listes, les listes normales et les listes déroulantes.

Les deux sont introduites par la balise `<select>` qui comprend les différents choix sous forme de balise `<option>` dont on renseigne la valeur qui sera envoyée au serveur grâce à son attribut `value`.

On peut même suggérer une proposition par défaut en précisant l'attribut `selected` de la balise `<option>` de notre choix. Par défaut, une liste est déroulante.

Exemple de liste

```
<select name="couleur_yeux">
  <option value="bleus">Bleus</option>
  <option value="bruns" selected="selected">Bruns</option>
  <option value="verts">Verts</option>
  <option value="noisettes">Noisettes</option>
  <option value="gris">Gris</option>
  <option value="vairons">Vairons</option>
</select>
```

Cette liste est donc une liste déroulante avec un seul choix possible parmi les six proposés. On peut rendre cette liste non déroulante en rajoutant l'attribut `size` qui prend pour valeur le nombre de choix à afficher, par exemple "3".

Ce qui donne :

Exemple de liste

```
<select name="couleur_yeux" size="3">
  <option value="bleus">Bleus</option>
  <option value="bruns">Bruns</option>
  <option value="verts" selected="selected">Verts</option>
  <option value="noisettes">Noisettes</option>
  <option value="gris">Gris</option>
  <option value="vairons">Vairons</option>
</select>
```

Ici encore, un seul choix est possible. Pour pouvoir proposer plusieurs choix, comme dans le cas de cases à cocher, il suffit de rajouter l'attribut **multiple** qui prend pour seule valeur "**multiple**". Voyons pour l'exemple :

Exemple de liste à choix multiple

```
<select name="matieres_preferees" size="5" multiple="multiple">
  <option value="maths">Maths</option>
  <option value="français">Français</option>
  <option value="anglais">Anglais</option>
  <option value="histoire">Histoire</option>
  <option value="sport">Sport</option>
  <option value="dessin">Dessin</option>
  <option value="musique">Musique</option>
  <option value="physique">Physique</option>
  <option value="sciences naturelles">Sciences naturelles</option>
</select>
```