

Atelier N°1**Initiation au langage HTML**

- Architecture Web
- Structure d'une page HTML
- Les formats du texte et des paragraphes
- Les listes, les images, les liens
- Les tableaux
- Les formulaires
- Application

I. Architecture Web

Le *World-Wide Web* (ou WWW, ou Web) est un très grand système d'information réparti sur un ensemble de *machines* connectées par le réseau Internet. Le web fonctionne selon le mode *client-serveur* pur. Les pages stockées sur le serveur HTTP (Hyper Text Transfert Protocol: *protocole de transfert permettant de demander un fichier au serveur ainsi qu'une description de types des données associées*).

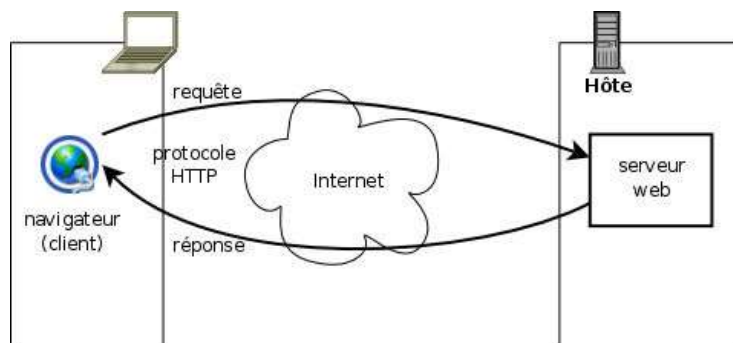
Les documents, et plus généralement les ressources sur le Web, sont identifiés par une URL (Uniform Resource Locator) qui encode toute l'information nécessaire pour trouver la ressource et aller la chercher. Cet encodage prend la forme d'une chaîne de caractères formée selon des règles précises illustrées par l'URL fictive suivante:

```
https://www.example.com:443/chemin/vers/doc?nom=orm&type=latex#fragment
```

Un client se connecte au serveur à travers un navigateur web tel que google chrome, Internet Explorer...

1. Le client formule une requête HTTP au serveur (*en précisant dans la barre d'url du navigateur un chemin fixe d'une page web*).
2. Le serveur répond à la requête : soit par un document (page web, image etc.) par tout moyen, soit en indiquant qu'il y a erreur (formulation incorrecte de la requête ou données non disponibles).
3. L'échange reprend à l'étape 2 ou se termine (et peut ensuite reprendre à l'étape 1).

Figure 2.1. Communication client-serveur web en HTTP

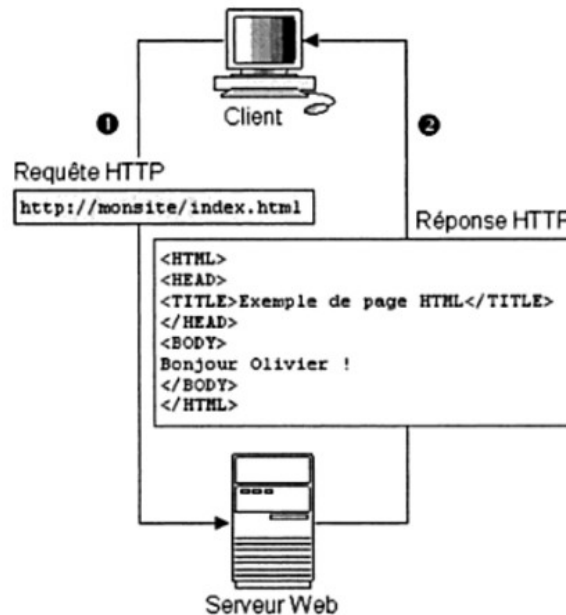


Ce type d'échange, bien construit, permet une totale transparence au réseau : chaque document peut se trouver n'importe où sur Internet, aussi bien sur la même machine que le client qu'à l'autre bout du monde.

1.1.1ère génération des architectures web : le web statique

Le web est essentiellement constitué de *documents hypertextes* décrites à l'aide du langage HTML (Hyper Text Markup Language). Ces pages sont stockées sur un serveur http (Hyper Text Transfert Protocol) précis accessible depuis n'importe quel poste connecté à l'Internet. Le poste client ne nécessite qu'un navigateur web. Les pages sont statiques. Elles seront toujours affichées à l'identique.

HTML (**H**yper **T**ext **M**arkup **L**anguage) est un langage de marquage de pages. Les marqueurs qui entourent un texte ou une image déterminent son apparence lorsqu'il sera visionné à l'aide d'un programme de navigation : grand ou petit, gras ou italique, aligné à gauche ou à droite, clignotant ou normal, voire même sa couleur. Les marqueurs déterminent également l'emplacement d'un objet sur une page.



2.2.2ème génération de l'architecture web : le web dynamique

Le contenu d'une page web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. À l'inverse, le contenu d'une page web statique est a priori identique à chaque consultation. Pour rendre les pages dynamiques, il n'y a pas d'autres solutions que d'insérer des modules programmés dans les pages côté clients et côté serveurs.

a. Côté client

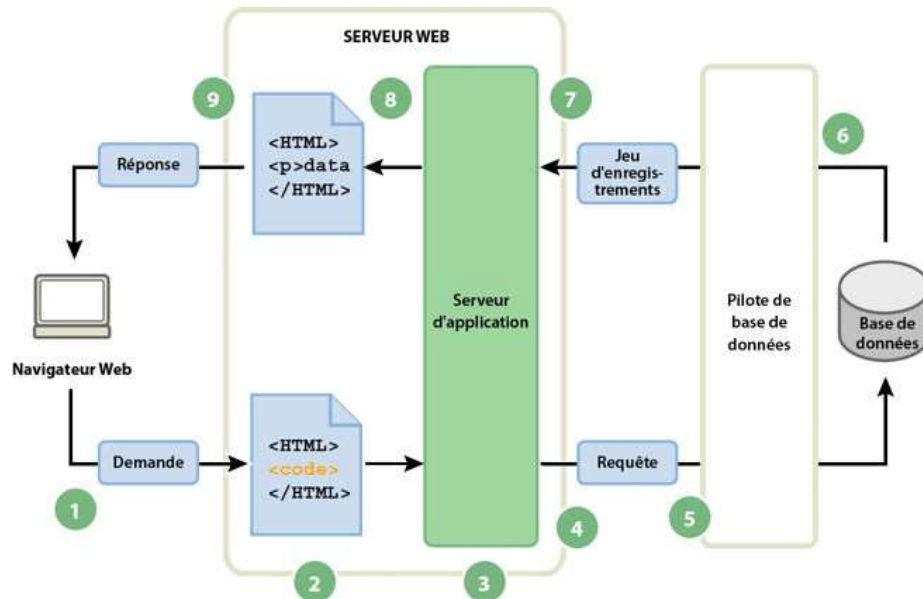
Les scripts de type **javascript** sont des petits programmes insérés au sein de la page qui s'exécutent au chargement ou suite à une action de l'utilisateur. Le javascript commençait alors à être utilisé (avec parcimonie) pour améliorer l'ergonomie. Puis l'Ajax est arrivé et à tout bouleversé. Il est petit à petit devenu évident qu'il n'était pas utile de changer de page à chaque clic (et ainsi tout recalculer côté serveur), alors qu'on pouvait ne mettre à jour que des morceaux de la page HTML. Avec Ajax, Javascript a pris tout son sens.



b. Côté Serveur

Une autre approche est d'augmenter les fonctionnalités du serveur. Au lieu de servir des documents stockés une fois pour toute (des pages dites *statiques*), le serveur va exécuter des instructions pour répondre à chaque requête et générer la réponse (code HTML) appropriée; on parle alors de pages *dynamique*.

L'exemple suivant illustre le processus d'interrogation de la base de données via des requêtes et de la transmission des données au navigateur :



La création d'une page dynamique consiste à écrire d'abord le code HTML, puis à ajouter les scripts ou les balises côté serveur au code HTML pour rendre la page dynamique. Lorsque vous visualisez le code obtenu, le langage apparaît incorporé dans le code HTML de la page. Par conséquent, ces langages sont appelés langages de programmation HTML intégrés. Les pages web dynamiques sont fréquemment écrits avec les langages PHP (.php), JavaServer Pages (.JSP) ou Active Server Pages (.ASP).

3.3.3ème génération de l'architecture web : les web Services

L'architecture applicative web moderne est orientée services, soumission partielle de formulaires, mise à jour que des parties de HTML concernées, voire seulement des données modifiées (avec prise en charge en Javascript côté client). Il ne s'agit donc plus d'appeler telle ou telle page mais d'invoquer tel ou tel service ultra spécialisé.

II. Structure et principes de base d'une page HTML

Un document écrit en HTML (document ou page source) contient des commandes appelées **balises** (tag en anglais). Ces dernières agissent de façon générale sur le groupe de mots qu'elles encadrent dans le document source.

A la différence du texte, l'image et la vidéo ne sont pas directement incorporées dans le document source, mais sous forme de références à des fichiers extérieurs qui sont alors chargés et exécutés.

La création d'une page web nécessite l'utilisation :

- D'un éditeur de texte simple comme par exemple le Bloc-notes ou Notepad de Windows.
- D'un interpréteur de commandes (browser ou navigateur web) tel que Netscape Navigator de Netscape ou Microsoft Explorer.

1. Etapes de construction d'une page web

1. Ouvrir l'éditeur de texte (par exemple Bloc-notes de Windows : Démarrer\ Tous les programmes\ Accessoires\ Bloc-notes ou bien NotePad ++ téléchargeable sur le net).
2. Ecrire le code HTML suivant :

```
<!doctype html>
<html>
  <head>
    <title>Ma première page</title>
    <meta charset=UTF-8></meta>
    <meta name="Description"
      content="Introduction au langage HTML">
    </meta>
  </head>
  <body>
    <p>Bonjour tout le monde!</p>
  </body>
</html>
```

3. Enregistrer le document avec l'extension **.htm** ou **.html** en format Unicode. (veuillez que le nom de la page ne comporte que des minuscules sans accents, espaces ou caractères spéciaux, afin d'éviter tout problème avec certains systèmes informatiques).
4. Ouvrir le navigateur (par exemple Internet Explorer).
5. Ouvrir le fichier par Fichier\ Ouvrir...

Pour vos éventuelles modifications, il n'est pas nécessaire de rouvrir à chaque fois le navigateur.

1. Retourner dans l'éditeur de texte (sans fermer le navigateur).
2. Modifier les codes HTML.
3. Enregistrer le fichier.
4. Utiliser la commande **Actualiser** dans le navigateur.

Pour retourner au code HTML à partir de la page web : Affichage\ source.

2. Les parties d'un document HTML

1.1. Le doctype

Le doctype est un code spécifique placé au tout début d'un document HTML. Il précise au navigateur le langage retenu pour l'écriture de la page : HTML ou XHTML, en version stricte ou transitionnelle, etc.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> pour xhtml
ou
<!DOCTYPE html> pour le html5

2.2. La balise principale <html>

3.3. L'entête <head>

a. Balise <title>

Le contenu de la balise **<title>** est le titre du document HTML : ce titre est affiché dans la barre de titre du navigateur.

b. Balise `<meta>`

La balise `<meta>` permet de donner des "méta-informations" sur le document. On écrira ainsi, par exemple :

```
<meta name="Mots clés" content="HTML, initiation">
```

```
<meta name="Description" content="Cette page fournit une introduction au langage HTML">
```

Ces meta sont utilisés principalement par les moteurs de recherche (*google*) avec des name spécifique tel que *word-key*

Les spécifications de langue et d'encodage est imposé par les normes du web.

La langue du document est précisée dans la balise `<meta>` dans l'en-tête `<head>` de tout document HTML comme suit :

```
<meta charset=UTF-8>
```

- iso-8859-1 qui est l'encodage classique, presque complet pour les langues de l'Europe occidentale ;
- utf-8 qui permet d'utiliser la plupart des caractères de la majorité des langues du monde

1.2. Le corps `<body>`

Il contient tout ce qui apparaîtra dans la fenêtre du navigateur, en utilisant les balises et les attributs nécessaires.

3. Les briques: balises et attributs, commentaires

La syntaxe utilisée est la suivante :

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

Balises

Les briques élémentaires d'un document HTML sont les balises, dotées ou non d'attributs. Ces balises permettent de structurer l'information. Une balise, sauf exceptions, s'ouvre et se ferme. Par exemple, pour déclarer un paragraphe, on écrira :

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

Il est possible également de déclarer un texte souligné :

```
<p><u>Ceci est un paragraphe souligné.</u></p>
```

Le navigateur ne fera pas apparaître les balises (ou *tags* ou marqueurs) `<p>`, `</p>`, `<u>` ou `</u>` mais interprétera leurs effets.

La balise est le moyen de communiquer avec le navigateur. Elle apparaît entre les caractères réservés `<` et `>`.

Attributs

Par exemple,

```
<p align="right">Ceci est un paragraphe aligné à droite.</p>
```

L'attribut **align** a pour effet d'aligner le texte à droite.

Les attributs permettent de modifier le comportement par défaut d'une balise, ou bien de spécifier des informations indispensables (comme l'adresse d'une image intégrée à la page, ou bien celle d'un lien externe).

Commentaires

Indispensables dans certains cas, ils allongent cependant le temps de chargement d'une page. La syntaxe est la suivante :

```
<!-- Commentaire -->
```

III. Les paragraphes

Le marqueur **<P>** désigne le saut d'une ligne vierge et le début d'un paragraphe et la balise **</P>** la fin du paragraphe.

Exemple 1

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple 1</title>
</head>
<body>

  <h2>The title attribute</h2>

  <p title="I'm a tooltip">
    Mouse over this paragraph, to display the title attribute as a tooltip.
  </p>

</body>
</html>
```

D'une façon générale, ce marqueur permet de séparer d'une manière appropriée deux paragraphes.

Le marqueur **
** permet de passer à la ligne suivante à l'intérieur d'un paragraphe (pas de **</BR>**).

IV. Les formats des textes

1. Les attributs :

Les attributs apportent une spécification supplémentaire au contenu des marqueurs. Ainsi, un texte entouré de **<H1 ALIGN=CENTER>** **</H1>** se présentera comme un titre centré. L'attribut **ALIGN= "CENTER"** indique la position du texte. D'autres propriétés sont utilisables afin de fixer la position du texte : **ALIGN = "RIGHT"** pour positionner le texte à droite. **ALIGN = "LEFT"** pour positionner le texte à gauche. **ALIGN = "JUSTIFY"** pour justifier le texte.

Exemple 2

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple 2</title>
</head>
<body>
  <h1>This is heading 1</h1>
  <h2>This is heading 2</h2>
  <h3>This is heading 3</h3>
  <h4>This is heading 4</h4>

</body>
</html>
```

La taille des titres est spécifiée par l'utilisation des marqueurs **<H1>**, **<H2>**,...**<H6>**, c'est le chiffre qui détermine la taille du texte, la plus grande taille est donnée par H1, la taille normale (celle du texte de type « normal ») par H3, et la plus petite par H6.

1. Autres balises :

Le marqueur gras
Le marqueur italique
Le marqueur souligné

** Mon texte est mis en gras.**
<I> Mon texte est passé en italique </I>.
<U> Mon texte est souligné </U>.

V. Les séparateurs

La balise **<HR>** permet d'insérer une ligne dans la page. Les valeurs par défaut de la balise **<HR>** sont :

- Une épaisseur de trait de 2 pixels.
- Un alignement centré.
- Une largeur de 100% de la fenêtre.

Il est possible de modifier les valeurs de ces attributs.

Epaisseur :

<HR size= > en pixels

Largeur :

<HR width = > en pixels

<HR width ="%"> en % de la fenêtre

Alignement :

<HR Align=left> : pour positionner la ligne à gauche

<HR align=right> : pour positionner la ligne à droite

<HR align=center> : pour centrer la ligne

VI. Les listes

1. Listes à puces balise

C'est le type de liste le plus simple. Elle s'introduit en utilisant la balise ****, qui signifie *Unordered List*. Elle possède le même attribut que la balise ****, qui s'utilise dans les mêmes conditions, et est également *obsolète en HTML4.01 strict* :

Attribut	Effet	Valeur(s)
type	Choix du type de puce	<ul style="list-style-type: none"> • disc (défaut) ○ circle ■ square

Exemple 3

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple 3</title>
</head>
<body>
  <ul>
    <li>Premier élément</li>
    <li type="circle">Deuxième élément</li>
    <li>Troisième élément</li>
  </ul>
</body>
</html>
```

2. Listes numérotées balise

Ce type de listes est ordonné. Elle s'introduit en utilisant la balise , qui signifie *Ordered List*. Les éléments de la liste sont également déclarés par la balise . La "numérotation" peut se faire de cinq manières différentes, cela étant spécifié par la valeur de l'attribut **type**. Elle peut débiter à n'importe quelle position (par exemple à 2, etc.), grâce à la valeur de l'attribut obsolète **start**.

Attribut	Effet	Valeur(s)
type	choix du type de numérotation	1 (par défaut) a A i I
start	numéro de début de liste	selon le type, un numéro ou une lettre

Exemple 4:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple 4</title>
</head>
<body>
  <ol>
    <li>Premier élément</li>
    <li>Deuxième élément</li>
    <ol type=a>
      <li>premier sous élément </li>
      <li>deuxième sous élément</li>
    </ol>
    <li>Troisième élément</li>
  </ol>
</body>
</html>
```

VII. Les images

On peut insérer des images. Le code Html est :

Exemple 5

1. Créer un dossier « img » dans votre dossier courant.
2. Copier l'image « collines.jpg » à partir du dossier "mes documents\mes images" dans le dossier « img ».
3. Créer le fichier exemple5.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple 5</title>
</head>
<body>
  <IMG align= «botton » SRC= « .\img\collines.jpg »>
</body>
</html>
```

RQ : « \ » désigne le dossier courant du fichier appelant.

La balise image possède de nombreux attributs.

Texte alternatif	alt="*****"	Pour les browsers n'ayant pas l'option "image" activée
Dimensions	width=? height=?	Hauteur et largeur (en pixels)
Bordure	border=?	(en pixels)
Alignement	align=top align=middle align=bottom align=left align=right	

VIII. Les liens

Pour définir un lien, il faut désigner son départ et son arrivée : c'est la même balise `<a>` (pour *anchor* ou ancre) à l'aide de deux attributs différents **href** et **name** qui les prend en charge.

Attribut	Effet	Valeur(s)
href	Hypertexte <i>RE</i> ference, désigne l'adresse à atteindre par le navigateur	URL valeur de l'attribut name (nom) de l'ancre à atteindre dans la même page Internet précédée du caractère #.
name	étiquette de l'ancre	chaîne de caractères
accesskey	définit une touche du clavier. La combinaison "ALT + touche" a le même effet que le clic sur le lien.	touche du clavier
tabindex	fixe la position de l'élément dans l'ordre séquentiel des tabulations.	valeur numérique entière
target	permet de diriger un document vers un cadre (frame) autre que celui qui contient le lien activé	
title	Une rapide description du lien	Une chaîne de caractères.

Exemple 6:

Dans l'éditeur de texte, nous allons créer deux fichiers Html.

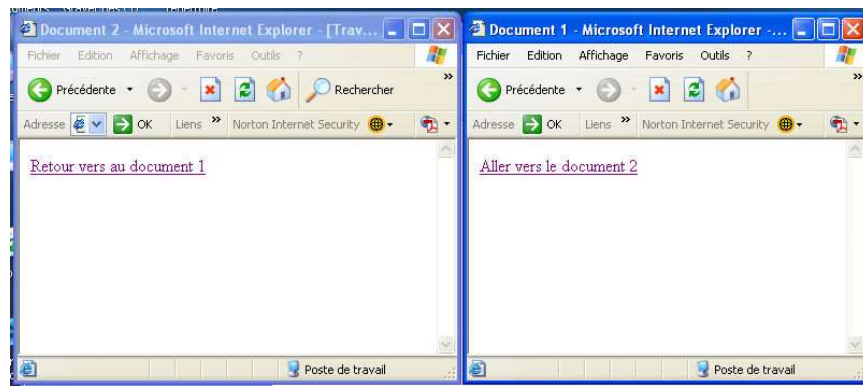
Le fichier Exemple6 doc1.html:

```
<html>
<head><title>Document 1 </title></head>
<body>
<A HREF="exemple6_doc2.html">Aller vers le document 2</A>
</body>
</html>
```

Le fichier Exemple6 doc2.html:

```
<html>
<head><title>Document 2 </title></head>
<body>
<A HREF="exemple6_doc1.html">Retour vers au document 1</A>
</body>
</html>
```

On sauve ces deux fichiers dans un même répertoire. Le browser affichera :



IX. Les tableaux

Usuellement un tableau est constitué de lignes et de colonnes. Ce n'est pas tout à fait le cas en HTML :

1. Un tableau est un conteneur **<table>**
2. Un tableau est constitué de lignes **<tr>** (pour *table row*)
3. Un tableau est constitué de cellules **<td>** (pour *table data*)

Il est également possible de fournir une légende avec la balise **<caption>** du tableau. Tous ces éléments se structurent globalement dans un entête, un pied de tableau, et un corps de tableau.

Exemples 7:

Pour insérer un tableau de bordure 1 avec une légende, ayant deux lignes, trois colonnes égales. Essayer ceci :

```

<!DOCTYPE html>
<html>
<title>Exemple 7</title>
<body>

<TABLE border=1>
  <caption>Légende du tableau.</caption>
  <TR>
    <TD>Cellule 11</TD>
    <TD>Cellule 12</TD>
    <TD>Cellule 13</TD>
  </TR>
  <TR>
    <TD>Cellule 21</TD>
    <TD>Cellule 22</TD>
    <TD>Cellule 23</TD>
  </TR>
</TABLE>

</body>
</html>

```

Légende du tableau.		
Cellule11	Cellule 12	Cellule13
Cellule 21	Cellule 22	Cellule 23

X. Les Formulaires

Les formulaires HTML (FORMS en anglais) sont des ensembles de composants, appelés aussi champs qui permettent à l'utilisateur d'entrer des informations, d'exprimer ses choix, de saisir du texte ... Toute la partie formulaire de la page doit se trouver entre les marqueurs `<FORM> ...</FORM>`.

Exemples 8:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple 8</title>
  </head>
  <body>
    <form method="post" action="">
      <p>
        <label for="pseudo">Votre pseudo :</label>
        <input type="text" name="pseudo" id="pseudo" />
        <br />
        <label for="pass">Votre mot de passe :</label>
        <input type="password" name="pass" id="pass" />
      </p>
    </form>
  </body>
</html>
```

a. Champs de texte

Ligne de texte

```
<INPUT TYPE="text" NAME="Nom du champ" VALUE="texte initial" SIZE=longueur>
```

Les divers paramètres

- `TYPE="text"` est facultatif, car il s'agit du type de champ par défaut (si on ne précise pas).
- `NAME` pour préciser le nom du champ (utile pour la programmation).
- `VALUE` pour donner un texte visible au champ.
- `SIZE` fixe la longueur visible du champ (éventuellement son contenu défile vers la gauche).
- `MAXLENGTH` pour limiter le nombre de caractères pouvant être entrés dans le champ.

On pourra énumérer les types utilisés en HTML5 tel que :

- `email` : le champ requiert un contenu au format d'adresse électronique.
- `url` : le champ accueille des URL absolues.
- `tel` : le champ est destiné aux numéros de téléphone.
- `number` : le champ accepte uniquement les caractères numériques. Accepte les attributs `min`, `max`, `step` et `Value`
- `range` se présentent sous la forme d'un curseur défilant, ou *slider* associé aux attributs `min`, `max`, `step` et `Value`

- **date** affiche un contrôle de date (année, mois, jour).
- **datetime** affiche un contrôle de date associé à une heure.
- **month** affiche un contrôle de mois.
- **week** affiche un contrôle de numéro de semaine.
- **time** affiche un contrôle d'heure.

• **color** : le champ est prévu pour les chaînes représentant une valeur de couleur.

Chacun des types de champ obéit à une norme qui définit les valeurs acceptées ou rejetées.

Ces différents types viennent s'ajouter aux valeurs classiques de HTML 4.01, à savoir

submit, **image**, **text**, **radio**, **button**, **checkbox**, **hidden**, **file** et **password**.

b. Boutons de commande

Les boutons servent à recevoir des "clics". Leur principale fonction est de permettre à l'utilisateur de déclencher des événements auxquels seront rattachées des fonctions.

Boutons communs

```
<INPUT TYPE="Button" NAME= ".." VALUE="Cliquez">
```

La valeur de l'attribut VALUE est le texte gravé sur le bouton. Un bouton sert essentiellement à déclencher des traitements locaux, par appel d'une fonction écrite en langage JavaScript.

Boutons de validation

Ce bouton sert à valider les informations saisies dans les champs du formulaire. Il provoque l'expédition de ces informations soit sur le serveur pour y être traitée, soit en message électronique ou bien

```
<INPUT TYPE="SUBMIT" NAME= ".." VALUE="Envoyer">
```

Envoyer

Boutons de réinitialisation

Ce bouton remet le formulaire dans son état initial, en réinitialisant les valeurs par défaut.

```
<INPUT TYPE="RESET" VALUE="Effacer">
```

Effacer

c. Liste de sélection

Pour permettre un choix dans une liste de plusieurs options présentées sous forme de liste déroulante, on utilise la balise **<SELECT>**

```
<SELECT NAME="..." SIZE=...>
  <OPTION SELECTED >option 1
  <OPTION>option 2
  <OPTION>option 3
</SELECT>
```

Les divers paramètres

- **<OPTION>** pour introduire chaque option de la liste.
- **SIZE= . . .** pour préciser le nombre de lignes visibles.
S'il y a plus d'options, la liste pourra être parcourue à l'aide d'une barre de défilement.
- **<OPTION SELECTED>** pour sélectionner cette option par défaut .
- **<SELECT MULTIPLE>** l'attribut *facultatif* **MULTIPLE** autorise à sélectionner plusieurs options dans la liste.

XI. Application

Créez un formulaire permettant à un membre d'une de ces associations de s'inscrire au repas, de donner ses disponibilités dans un ensemble de dates proposées et de spécifier quel type de plat il préparera. Ce formulaire doit contenir au moins un élément de chaque type d'entrée de formulaire possible (champ texte, menu déroulant, cases à cocher, etc.).

Piquenique Multiassociations



Inscriptions

Prenom :

Mot de passe :

Association :

Disponibilites pour la semaine du:

☐ Lundi ☐ Mardi ☐ Mercredi ☐ Jeudi ☐ Vendredi

Contribution : ☐ Entree ☐ Plat ☐ Dessert

1. Donnez pour titre à ce fichier « Fiche Multiassociations ».
2. Enregistrez le fichier sous le nom « votrenom.html ».

Rq : L'image « repas.jpg » est disponible dans le dossier Image fournie en classe.