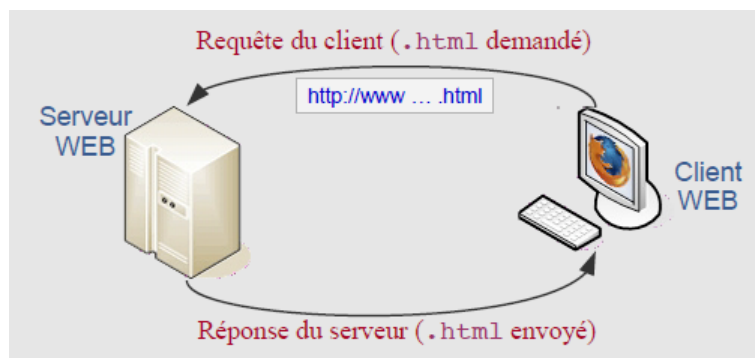


Annexe 3 : HTML, Formulaires et CSS

1 Généralités

- Côté serveur Web : Fichiers `.html` (ou `.htm`), enregistrés dans un répertoire spécial (`/www/ext`)
- Côté client : Navigateurs lisent le HTML/CSS
- **Sites statiques** : uniquement HTML et CSS (même code chargé par tous les clients).



Il est possible d'insérer ici du code **Javascript**, qui sera exécuté par le client (si activé dans le navigateur) et qui permet d'ajouter de l'interactivité dans les pages (validation de formulaire).

Pour écrire des pages en HTML, il y a des règles à respecter et une syntaxe à utiliser (balises) (`<head>`, `<body>`, ...)

Il faut séparer le code du "fond" et de la "forme" : le texte de la page (langage HTML) et la mise en forme de la page (langage CSS)

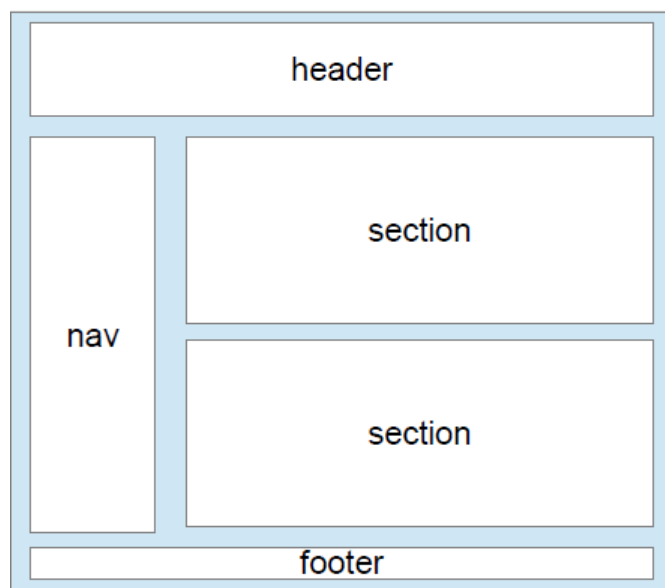
Un site contient ainsi des fichiers `.html` et des fichiers `.css`

Attention, tous les navigateurs n'interprètent pas le HTML et le CSS de la même façon.

2 Structure d'une Page

Par exemple, une page peut être constituée comme ceci :

```
<!DOCTYPE HTML>
<html lang='fr'>
  <head>
    <meta charset='utf-8'/>
    <link rel='stylesheet' href='fic.css'/>
    <title> Ma page </title>
  </head>
  <body>
    <header>
      ...
    </header>
    <nav>
      ...
    </nav>
    <section>
      ...
    </section>
    <footer>
      ...
    </footer>
    ...
  </body>
</html>
```



3 Quelques balises

3.1 Balises de type "bloc"

- Niveaux de titre : `<h1> ... </h1>`
- Listes numérotées et non numérotées :

```
<ol>
  <li> ... </li>
  <li> ... </li>
</ol>
```

```
<ul>
  <li> ... </li>
  <li> ... </li>
</ul>
```

- Tableaux : (attributs `colspan`, `rowspan`)

```
<table>
  <tr> <!-- Lig -->
    <td> ... </td> <!-- Col -->
    ...
  </tr>
  ...
</table>
```

- Paragraphes : `<p> ... </p>`

3.2 Balises de type "inline"

- Mise en évidence simple : ` ... `
- Mise en évidence forte : ` ... `
- Passage à la ligne : `
`
- Insertion d'une image : ``
- Liens vers une autre page, une ancre, mail : ` Clic Clic `

4 Formulaires

4.1 Généralités

Les informations à saisir dans le formulaire seront "envoyées" à un script **PHP**.

Il y aura donc 2 fichiers :

- ⇒ Un fichier **.html** (ou **.php**) contenant au moins le code HTML du formulaire.
- ⇒ Un fichier **.php** pour le traitement des informations en provenance du formulaire.

En code HTML, cela donne :

```
<form method='post' action='script.php'>
  <p>
    <!-- Ici les champs du formulaire -->
  </p>
</form>
```

Il faut obligatoirement un champ **button** de type **submit**.

Le script **script.php** récupérera les données dans un tableau (`$_POST`).

4.2 Les Champs

- Champ input : `<input type='typeChamp' name='nomchamp' />`
 - nomChamp : nom du champ passé au script PHP
 - typeChamp : nature du champ
 - 'text' : petite zone de texte
 - 'password' : mot de passe
 - 'checkbox' : case à cocher
 - 'radio' : bouton d'option (1 parmi N de même nom)
 - 'hidden' : champ caché (valeur invisible dans le formulaire, passée au script PHP)
 - 'email', 'tel', 'url', ... (attention aux navigateurs)
 - Attributs (options) : autofocus, checked, required, value, ...
 - Champ "label" (libellé d'un autre champ) : `<label for='idChamp'> Libellé </label>`
 Quand on clique sur "Libellé" le champ dont l'id vaut 'idChamp' est sélectionné.
 - Champ "select" (liste déroulante) :


```
<select name='nomListe'>
  <option value='val1'>choix 1</option>
  <option value='val2'>choix 2</option>
  ...
</select>
```
- Par exemple, si l'utilisateur sélectionne "choix 2" la valeur passée au script PHP sera : val2
- Champ "textarea" (grande zone de texte) : `<textarea name='nomZone'> Texte </textarea>`
 - Champ "button" (bouton) : `<button type='submit'> Libellé </button>`
 - Doit être toujours présent dans les formulaires
 - Quand on clique sur le bouton submit :
 - ⇒ Envoi au script PHP du tableau associatif : \$_POST
 - ⇒ \$_POST contient les valeurs de tous les champs du formulaire
 - Autre type de bouton : 'reset'
 - Attributs indispensables pour tous les champs :
 - name : utilisé par le script **PHP** pour récupérer la valeur du champ
 - id : utilisé par le **CSS** et par **JavaScript**.
 Il est possible d'avoir évidemment la même valeur pour le name et l'id.

Par exemple,

```
<form method='post' action='monScript.php'>
  <p>
    <label for='prenom'>Ton prénom</label>
    <input type='text' name='prenom' id='prenom' autofocus required/><br/>
    <input type='radio' name='sexe' value='h' /> Homme
    <input type='radio' name='sexe' value='f' checked/> Femme<br/>
    <label for='lstSport'>Sport</label>
    <select name='lstSport' id='lstSport'>
      <option value='foot'>Foot</option>
      <option value='tennis'>Tennis</option>
      <option value='hand'>Handball</option>
      <option value='danse' selected>Danse</option>
    </select><br/>
    <button type='submit' name='btn'>Go</button>
  </p>
</form>
```

5 CSS

- Possibilité de mettre les CSS dans :
 - Directement dans les balises (attribut `style`)
 - Dans la partie `<head>`
 - Un fichier `.css`

- Appeler le `.css` dans le `<head> ... </head>` :

```
<link rel='stylesheet' href='fic.ss' />
```

- Dans le fichier `.css`, indiquer les balises avec les propriétés de style :

```
balise{  
    propriété : valeur;  
    ... }
```

- Possibilité de désigner les `class` (.) et les `id` (#)