

## 1.Qu'est-ce que l'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 un langage interprété. Ce langage est pourvu d'un système de balisage qui va permettre de structurer notre document.

## 2. Une balise HTML

Une **balise** (**tag** ou **markup** en anglais) est un mot clé qui va donner des indications au navigateur sur ce qu'il doit afficher. Elle n'est pas affichée telle quelle dans le navigateur, elle est interprétée par celui-ci. Les caractères qui ne sont pas compris entre les signes « < » et « > » sont donc considérés comme du texte et seront affichés tel quel par le navigateur. La balise 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">
```

## 3. Les types des balises :

On distingue deux types de balises :

- Les balises simples.
- Les balises doubles.

### 3.1 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*

```

```

### 3.2 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>
```

### 3.3 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>
```

## 4. Structure d'un document HTML

Avant de commencer tout développement, il est important de connaître la structure d'une page HTML.

*Document HTML de base minimal*

```
<!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 code source -->
</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 tout l'en-tête de la page et `<body>` qui va contenir tout le corps du document.

### 4.1 Le DOCTYPE

Il sert d'une part à déclarer quel "langage", "grammaire", 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. 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.

### 4.2 Les commentaires :

Les commentaires ne seront pas affichés dans le navigateur mais ils resteront présents dans le code source. Ils sont délimités par `<!--` et `-->` et peuvent être multilignes.

Par souci de compatibilité entre les navigateurs et de mal interprétation de ceux-ci, il est fortement déconseillé d'écrire deux tirets d'union (ou plus) consécutifs.

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

```
<!-- Voici un deuxième commentaire.
```

```
  Celui-ci est multilignes. -->
```

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

### 4.3 Le corps du document - body

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

#### 4.3.1 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 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.

#### 4.3.2 Les paragraphes de texte

Il faut préciser en html 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>
```

### 4.3.3 La mise en forme de texte

Les différentes techniques pour mettre en forme un texte sont les suivantes :

#### Le texte en gras

`<b>Ce texte s'affichera en gras.</b>`

#### Le texte souligné

`<u>Ce texte sera souligné.</u>`

#### Le texte en exposant

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

#### Le texte en italique

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

#### Le texte barré

`<s>Ce texte sera barré.</s>`

#### Le texte en indice

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

#### Modifier la couleur du texte

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

##### 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,
- Textuelles en anglais US ; Exemples : red, yellow, pink.

#### Modifier la police du texte

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

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

#### Modifier la taille du texte

Pour modifier la taille du texte, on utilise l'attribut `size` de la balise `<font>` :

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

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

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

### 4.3.4 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.

#### Les différents types de listes

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

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

#### Les listes non ordonnées

Elles sont démarquées par la balise `<ul>` et chacun des items est entouré de la balise `<li>` :

*Exemple de liste non ordonnée*

`<p>Ma liste de matières :</p>`

`<ul>`

`<li>Mathématiques</li>`

`<li>Informatique</li>`

`<li>Chimie</li>`

`<li>Physique</li>`

`</ul>`

## Les différents types de puces

On renseigne le type des puces avec l'attribut **type** de la balise `<ul>`.

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.

*Exemple de liste non ordonnée*

```
<p>Ma liste de matières :</p>
```

```
<ul type="square">  
  <li>Mathématiques</li>  
  <li>Informatique</li>  
  <li>Chimie</li>  
  <li>Physique</li>
```

```
</ul>
```

## Les listes ordonnées

Elles sont introduites par la balise `<ol>` et chacun des items est encadré par la balise `<li>` :

*Exemple de liste ordonnée*

```
<p>Top 3 du tiercé :</p>
```

```
<ol>  
  <li>Jolly</li>  
  <li>Tonnerre</li>  
  <li>Tornado</li>
```

```
</ol>
```

## Les différents types de numérotation

Il existe cinq manières différentes de numérotter les listes ordonnées. On renseigne le type de numérotation avec l'attribut **type** de la balise `<ol>`.

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.

## 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 jeunesse</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>
```

### 4.3.5 Les tableaux :

Un tableau en html est délimité par la balise <table>. Le contenu d'un tableau HTML est généré horizontalement. C'est-à-dire qu'il n'est pas créé colonne par colonne mais ligne par ligne. 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. Et pour un 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>
```



Cellule 1	Cellule 2
Cellule 3	Cellule 4

### 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>
```



Cellule qui se répand sur 5 colonnes				
Colonne 1	Colonne 2	Colonne 3	Colonne 4	Colonne 5

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>
```

### 4.3.6 Les images :

Pour afficher une image en HTML, on utilise la balise <img>. La balise image admet entre autre les propriétés suivantes :

src = le chemin et le nom du fichier image.

border = en pixels

width = la largeur en pixels ou pourcentage (par rapport à la taille de la page web)

height = la longueur en pixels ou pourcentage (par rapport à la taille de la page web)

align= Top, Bottom, Middle, Left, Right

Exemple : ``

#### 4.3.7 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 et autres balises .

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 d'un lien texte

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

Exemple d'un lien image

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

#### 4.3.8 Les cadres :

Les cadres servent à séparer la fenêtre en différents morceaux indépendants qui peuvent chacun contenir une page HTML différente, ces morceaux ayant la capacité de communiquer entre eux.

On utilise au minimum deux cadres dans une fenêtre, en général un cadre pour le menu à gauche et un autre pour le contenu du site. Parfois on rajoute un cadre en haut pour un bandeau fixe.

Il y a les balises `<frameset>` et `<frame>` qui vont être utilisées pour la mise en œuvre du jeu de cadres. La balise `<frameset>` remplace d'ordinaire la balise `<body>`.

La balise `<frameset>` va désigner le nombre de cadres et leur taille.

Si on renseigne l'attribut *cols*, la page sera coupée verticalement, alors que si on renseigne l'attribut *rows*, la page sera coupée horizontalement. Ces attributs prennent pour valeur les tailles des différents cadres séparées par une virgule. Si une des valeurs est une étoile, ça signifie que le cadre en question prendra toute la place disponible restante dans la fenêtre. On ne peut évidemment pas utiliser ces deux attributs conjointement sur une seule et même balise `<frameset>`.

Deux cadres verticaux

```
<frameset cols="20%, *">
```

Trois cadres verticaux

```
<frameset cols="20%, *, 20%">
```

Deux cadres horizontaux

```
<frameset rows="100, *">
```

Exemple de cadres

```
<frameset cols="200, *">
```

```
  <frame name="menu" src="menu.html">
```

```
  <frame name="contenu" src="page1.html">
```

```
</frameset>
```

#### 4.3.9 Les formulaires

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.

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*, qui peut prendre pour valeurs une URL en absolu ou en relatif, voire même une adresse e-mail.

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" et "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".

**Exemple de balise form**

```
<form action="envoi-formulaire.php" method="post">
```

```
  <!-- Contenu du formulaire -->
```

```
</form>
```

- Chaque composant interactif d'un formulaire (zone de texte, case à cocher, case à option, liste, bouton ...) est défini par un nom (Name) et sera considéré ensuite comme une variable contenant la valeur saisie (ou choisie) dans ce composant. Lors du clic sur le bouton SUBMIT (valider ou envoyer), ces variables (avec leurs valeurs) seront envoyées à un micro serveur pour les traiter.
- Ces données sont traitées par un programme :
  - Ecrit en un langage de programmation spécifique et pas le HTML (exemple : PHP, ASP ...)
  - Ce programme sera exécuté dans le micro serveur, et les résultats seront par la suite envoyés au navigateur du micro client pour les afficher.

```

<H2><u><CENTER>Fiche de renseignements</CENTER></u></H2>
<form methode=post action=RECHERCHE_auteur.php>
<!-- les données du formulaire seront traitées dans le micro serveur par un
fichier nommé : RECHERCHE_auteur.php -->

Nom : <INPUT TYPE ="text" NAME="Nom" VALUE="" >
Prénom : <INPUT TYPE ="text" NAME="Prenom" VALUE="" >
Sexe : <SELECT NAME="Sexe" VALUE="" >
<OPTION>Masculin</OPTION>
<OPTION>Féminin</OPTION>
</SELECT>
Mot de passe : <INPUT TYPE="password" NAME="Pass" SIZE=5><P>

Type de formation :
<INPUT TYPE="radio" NAME="option" VALUE="LMD" checked> Système LMD
<INPUT TYPE="radio" NAME="option" VALUE="Classique"> Système Classique
<P>

Quelle est votre spécialité préférée ?
<SELECT NAME="spacialites" MULTIPLE SIZE=3>
<OPTION>Mathématiques
<OPTION SELECTED>Informatique
<OPTION>Chimie
<OPTION>Physique
<OPTION>Langues
<OPTION>Gestion
<OPTION>Autres ...
</SELECT>
<P>

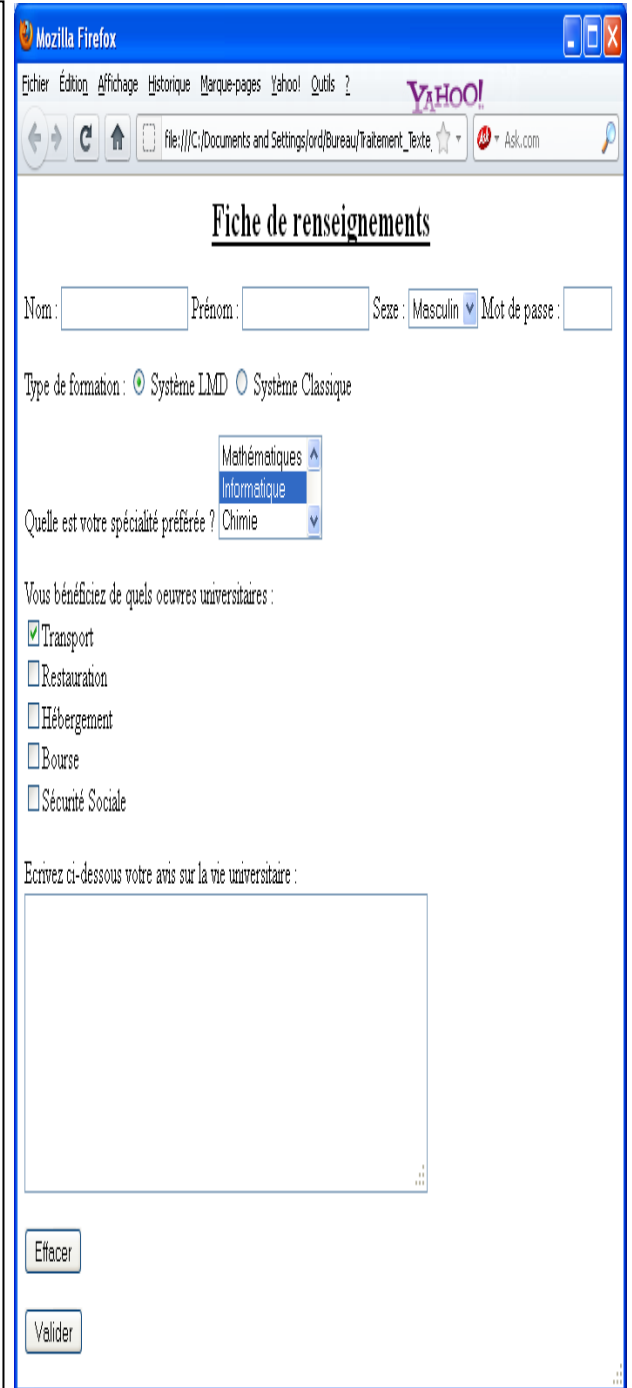
Vous bénéficiez de quels oeuvres universitaires : <BR>
<INPUT TYPE="checkbox" CHECKED NAME="Transport" >Transport<BR>
<INPUT TYPE="checkbox" NAME="Restauration">Restauration<BR>
<INPUT TYPE="checkbox" NAME="Hebergement">Hébergement<BR>
<INPUT TYPE="checkbox" NAME="Bourse">Bourse<BR>
<INPUT TYPE="checkbox" NAME="Securite">Sécurité Sociale
<P>

Ecrivez ci-dessous votre avis sur la vie universitaire :<BR>
<TEXTAREA NAME="Projet" ROWS=8 COLS=55>
</TEXTAREA>
<P>
<INPUT TYPE="RESET" VALUE="Effacer">
<!-- Pour recommencer --> <P>

<INPUT TYPE="SUBMIT" VALUE="Valider">
<!-- Pour envoyer ces informations --> <P>

</FORM >

```



#### 4.4 Entête de document html :

La balise <head> contient de nombreux renseignements sur la page, sur l'auteur, ainsi que la plupart des scripts (JavaScript, VBscript,...) .

##### 4.4.1 La balise titre :

Indique le titre de la page html.

Exemple de la balise titre :

```
<head>
  <title>Tutoriel HTML - Apprenez les bases du HTML</title>
</head>
```

Notez bien que le **titre ne s'affiche pas dans la page** (seul ce qui est dans le **body** peut s'y afficher) mais dans la barre située en haut du navigateur.

#### 4.4.2 La balise meta :

La balise **meta** va nous permettre de donner au navigateur des informations sur la page web à afficher. On dit que ce sont des **métadonnées** (du grec meta "après" et du latin data "informations", soit une donnée servant à définir ou décrire une autre donnée).

On distingue **2 types** de balises méta :

- Les **balises méta "name"** qui permette de décrire la page.
- Les **balises méta "http-equiv"** qui envoie des informations supplémentaires via le protocole HTTP.

##### Les balises méta "name"

Elles s'utilisent toujours comme ceci : `<meta name="mot clé ici" content="valeur ici" />`

Elles nous permettent notamment de préciser :

- L'auteur de la page grâce au mot clé **author** :

```
<meta name="author" content="Nom de la personne qui a réalisé la page" />
```

- Une description de la page grâce au mot clé **description** :

```
<meta name="description" content="Brève description de la page ici" />
```

- Une liste de mots clés pour qualifier la page grâce au mot clé **keywords**. Ces mots clés seront séparés par des virgules :

```
<meta name="keywords" content="mot clé 1, autre mot clé, blabla" />
```

- Le logiciel utilisé pour réaliser la page grâce au mot clé **generator** :

```
<meta name="generator" content="Adobe Dreamweaver" />
```

##### Les balises méta "http-equiv"

Elles s'utilisent comme ceci :

```
<meta http-equiv="mot clé ici" content="valeur ici" />
```

Elles nous permettent :

- de préciser l'encodage utilisé pour la page grâce au mot clé **Content-Type**. Exemple pour un encodage au format UTF-8 :

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

- de préciser si la page ne doit pas être sauvegardée dans la cache du navigateur (c'est à dire que la page ne sera pas mémorisée dans les fichiers temporaires du visiteur) grâce au mot clé **pragma** :

```
<meta http-equiv="pragma" content="no-cache" />
```

- de préciser une date d'expiration pour la page grâce au mot clé **expires** :

```
<meta http-equiv="expires" content="Tue, 20 Aug 1996 14:25:27 GMT" />
```

- de demander à la page de se rafraichir (= se réactualiser) toutes les N secondes grâce au mot clé **refresh**. Par exemple, pour que la page se réactualise toutes les 30 secondes :

```
<meta http-equiv="refresh" content="30" />
```