LE HTML :

Les points importants à retenir :

Pour faciliter les choses, il est important de créer un dossier et de tout intégrer à l’intérieur.

Chaque document de code doit être enregistré en tant que « document.html »

L’un des meilleurs éditeurs de textes pour coder est visual studio code. Il est important de télécharger le plug in « open in browser » pour pouvoir directement visualiser ce que l’on code sur un navigateur grâce à la commande « alt+b ». « alt+shift+b » permet d’avoir une liste de navigateurs pour vérifier la compatibilité sur tous les formats.

Pour commencer, il faut noter « html » et sélectionner « HTML5 » pour avoir accès à la dernière version reconnue par W3C, avec des lignes de codes pré enregistrées pour faciliter les choses.

Pour mettre du texte pour tester les pages, si l’on ne sait pas quoi écrire, il peut être utile de taper « lorem ipsum » sur google pour avoir un texte généré, afin de se faire une idée de la présentation de la page.

Les différentes balises :

Il existe deux types de balises : les balises de lignes et les balises de blocs.

Une balise de ligne ne sert à baliser qu’une ligne, comme par exemple le <i> </i> qui sert à mettre une ligne en italique.

Une balise de bloc sert à baliser tout un bloc, ou un paragraphe. Ainsi, quand on ferme la balise, l’éditeur reviendra seul à la ligne.

<head> </head> est une balise qui est utile pour le navigateur. Celui-ci va interpréter ce qu’il y a intégré dans la balise.

<body> </body> sert à désigner le corps du texte.

<P> </p> désigne un paragraphe.

<br> désigne un passage de ligne dans le paragraphe.

<hr> désigne une ligne (tracée) entre deux paragraphes. On peut désigner une taille à cette ligne avec <hr width= « 200 »> et la largeur de cette ligne avec <hr size= « x »>

<pr> </pr> désigne une zone de texte où les espaces sont respectés.

<i> </i> désigne une zone en italique

<u> </u> désigne une zone surlignée

<b> </b> désigne une zone en gras

<s> </s> désigne une zone barrée

<center> </center> permet de centrer un texte. Cette commande est obsolète.

<div> </div> permet de créer un bloc style un paragraphe mais qu’on pourra bouger et exploiter, par comme une simple zone de texte.

Il en existe plein d’autres.

Les balises contenant un lien :

***Pour insérer une image :***

Depuis le dossier : <img src= « img.jpg »/>

Depuis un dossier externe :<img src= « nomdudossier/img.jpg »/>

Depuis un lien internet : <img src= [http://lien.sélectionné.com/exemple.jpg »/](http://lien.sélectionné.com/exemple.jpg )>

Pour définir la largeur et la longueur de l’image on doit faire les commandes :

<img src= « image.jpg » width= « 200p » height= « 200px » />

Si on ne précise pas la taille, l’image sera probablement trop grande ou trop petite.

On peut donner la taille en pourcentages pour dimensionner par rapport à la taille du navigateur.

Il faut ajouter à cette balise alt= « description de l’image » pour que le moteur de recherche décrive l’image aux malvoyants.

R0 : imgU.gif

R1 : index.html  
 img.gif

R2 : img2.gif

***Les chemins relatifs :***

Pour insérer une image issue du même dossier (ici r1) :

<img src= « img.gif » />

Pour insérer une image issue d’un dossier dans le dossier où est présent idex.html (ici R2) :

<img src= « R2/img2.gif » />

Pour insérer une image issue d’un dossier parent où est stocké R1 (ici R0) :

<img src= « ../imgU.gif » />

Ici, le symbole ../ signifie que le dossier est parent au dossier de départ. Ce signe sert à remonter. S’il y avait un dossier R-1, pour y accéder depuis le dossier 1 il faudrait mettre :

<img src= « ../../image.jpeg » />

***Les chemins absolus :***

Si au lieu d’avoir l’image dans le dossier, elle est stockée ailleurs, il est possible d’aller la chercher :

<img src= « c:/users/nom du dossier/img.gif » />

Pour Linux :

<img src= « c:/usr/home/nom du dossier/img.gif » />

Dans VS code, la commande ***ctrl+spacebar*** permet de faciliter les liens en allant chercher directement l’image.

Pour insérer un lien :

Depuis le dossier : <a href= « document.html »> texte à cliquer </a>

Depuis internet : <a href= <http://www.google.com>> texte à cliquer </a>

Depuis un dossier <a href= « ../../apropos.html »> texte à cliquer </a>

Les ../ signifient que l’on part du dossier où est le fichier pour remonter sur un dossier au-dessus.

Dans la balise du lien on peut faire que si on clique sur le lien soit ça ouvre sur l’onglet existant, soit un nouvel onglet grâce à la balise target.

<a href= «  <http://www.google.com> » target= « \_blank »> google </a>

Si on met « **\_ blank** » dans target, on ouvrira un nouvel onglet. Si on met « **\_ self**», ça ouvrira le lien sur le même onglet.

***Créer un lien vers une adresse email :***

<a href= « mailto :antoine.afpa@gmail.com »> mail </a>

***Il est possible de mettre une image à la place d’un mot pour renvoyer vers un lien.***

<a href= « <http://www.google.com> »> <img src= « img.jpeg » width= « xxpx » height= « xxpx » > </a>

***Pour un lien interne à la page :***

Par exemple, un lien pour revenir en haut de la page :

### <a href= « #top »> texte à écrire </a>

Il faut cibler une partie pour qu’elle devienne un lien. Par exemple, dans ce cas, on va mettre en haut de la page **<a name= « top »> le mot </a>.** Si on veut qu’un mot fasse office de lien, il suffit de l’écrire entre les balises.

***Pour un lien externe directement mais directement sur un lien interne de la nouvelle page :***

<a href= « index.html#paragraphe1 »> nom du lien </a>

Il faut évidemment créer un endroit <a name= « #paragraphe1 » paragraphe1 </a> pour que le lien de l’autre page renvoie vers lui.

Les listes :

Les listes se présentent avec la balise <ul> </ul>. UL signifie *unordered list*

Une liste se présente de la sorte :

<ul>

<li> un </li>

<li> deux </li>

</ul>

On peut aussi utiliser la commande <ol> </ol> qui signifie *ordered list.*

Elle se présente de la sorte :

<ol>

<li> un </li>

<li> deux </li>

</ol>

La liste désordonnée mettra des points par défaut, tandis que la liste ordonnée mettra des chiffres.

Les tableaux :

Pour faire des tableaux, il faut utiliser : <table> </table>

Chaque ligne se définit par <tr> </tr>

Chaque donnée ou cellule se définit <td> exemple </td>

Concrètement, cela donnerait :

<table>

<tr> <td>histoire </td> <td> géo </td> </tr>

<tr> <td>maths </td> <td> français</td> </tr>

Pour une en-tête de tableau : on utilise <th> </th>

<tr> <th> NOM </th> <th>PRENOM </th> </tr>

Exemple concret :

<table>

</tr>

<tr>

|  |  |  |
| --- | --- | --- |
| <th> **donnée** </th> | <th> **donnée** </th> | <th> **donnée** </th> |
| <td> donnée </td> | <td> donnée </td> | <td> donnée </td> |
| <td> donnée </td> | <td> donnée </td> | <td> donnée </td> |

</table>

Pour dessiner les lignes du tableau, il suffit juste de taper <table **border= « 1 »**> dans la balise d’ouverture du tableau.

Pour faire qu’une ligne du tableau ne fasse qu’une seule grande case, par exemple sur un tableau de trois cases :

<td colspan= « 3 »> Donnée </td>

Pour avoir deux cases ça serait donc : <td colspan= « 2 »> table</td> <td>**&nbsp ;** </td>

|  |  |  |
| --- | --- | --- |
|  | <td colspan= « 3 »> table </td> |  |
|  |  |  |
|  |  |  |

&nbsp ; signifie que l’on affiche une donnée invisible pour remplir un espace vide afin de donner à une case vide une taille normale.

|  |  |  |
| --- | --- | --- |
|  | <td colspan= « 2 » x </td> | <td > &nbsp ; </td> |
|  |  |  |

Pour faire une colonne, on utilisera la formule :

<td rowspan= « 3 »> donnée </td>

|  |  |  |
| --- | --- | --- |
|  |  |  |
| <td rowspan= « 3 »> x </td> |  |  |
|  |  |  |

***Pour avoir la taille des cellules du tableau en pourcentages, peu importe la taille de l’écran :***

<table width= « 80% »> signifie que le tableau fait 80% de la page en largeur.

<td width= »33%> </td> <td width= »33%> </td> <td width= »34%> </td> signifie que chacune des trois case fait 33% du tableau. La dernière à 34% est là pour arriver à 100%.

***Pour qu’une information soit affichée en haut de tableau ou de page peu importe la taille :***

<tr> **<td valign= « top »>** Information à placer en haut</td> </tr>

Les commentaires :

Ils sont très importants en développement pour comprendre la logique du développeur ou simplement se souvenir de ce que l’on faisait.

Pour faire un commentaire, dans le <body> </body> :

**< !- -**commentaire, uniquement visible en faisant afficher le code source de la page **- ->**

*(Il n’y a pas d’espace entre les deux tirets, c’est uniquement pour plus de visibilité.)*

L’hexadécimale :

L’hexadécimale représente la couleur. Chaque code représente une couleur différente

Rouge Vert Bleu

#FF 22 16

Par exemple, pour colorer un tableau, on mettra <table bgcolor= «red »> On peut remplacer la couleur par le code hexadécimal.

Les informations meta :

Ce sont des informations propres au travail du développeur. Elles servent à identifier certaines informations par exemple ou participent au référencement d’une page. Elles se situent dans le <head> </head>.

<meta charset= « uft-8 »> signifie que l’écriture occidentale est reconnue en intégralité.***(primordial)***

<meta author= « Leinhart »> permet d’identifier l’auteur du code.

<meta generator= « le générateur qu’on a utilisé pour coder, VScode par exemple »>

<meta keyword= « … », « … », « … »> Ils servent à participer au référencement de la page (10% de l’importance du référencement.)

<link rel= « icon » type= « image/x-icon » href= « image.ico »> Ce meta permet de mettre une icône sur l’onglet. (ce meta ne change jamais)

Il existe d’autres meta, une trentaine en tout. Certains sont plus utiles que d’autres.

Les styles :

Il existe trois types de styles :

Les styles Inline

Les styles Internal

Les styles external

***Les styles Inline :***

Permet de mettre la couleur d’écriture de tout un paragraphe en une couleur choisie. Il est de moins en moins utilisé.

<p style= « color=red »>

***Les styles internal :***

Il permet d’adapter une couleur à tous les paragraphes d’une page.

Dans le head on va créer un <style>

P{

color :red

}

</style>

***Le style external***:

On va passer par une feuille CSS pour styliser la page. C’est le plus utilisé et le meilleur. Dans le <head> On rentre le meta :

<link rel= « stylesheet » href= « style.css » />

Ce meta ne change jamais.

Les div :

Les <div> </div> sont comme des paragraphes <p>, saut qu’à la fin il n’y a pas de passage de ligne mais un retour chariot.

Un paragraphe <p> sert à délimiter un texte. Le div lui sert à délimiter un objet pour le déplacer où on veut, mais n’a aucune position fixe comme un paragraphe. On peut mettre une image, vidéo ou n’importe quoi dans un <div>.

**La div sert à définir un bloc que l’on peut manipuler.**

Les span :

Un <span> </span> peut intervenir au milieu d’une ligne. Contrairement au <div>, il n’y a pas de passage de ligne après un span.

**Un span est un bloc comme une div, mais un « bloc ligne ».**

Les formulaires :

Les formulaires servent à saisir des données et les envoyer à une autre page qui va les enregistrer.

<form mehod="get">

Prénom:<input type="text" name="prenom" > <br>

Nom:<input type="text" name="nom"> <br>

Sexe:Homme <input type="radio" name="gender">

Femme<input type="radio" name="gender">

Autre <input type="radio" name="gender"> <br>

Password: <input type="password" name="password"> <br>

email: <input type="email" required name="email"> <br>

<input type="submit" value="envoyer">

<input type="submit" value="effacer">

</form>

Dans la balise <body> pour créer un formulaire il faut créer une balise <form> </form>.

Il existe deux méthodes de formulaires : la méthode= « get » et la méthode= « post ».

La méthode get est la plus courante, mais les données sont faciles à récupérer. La méthode post permet de traduire les données en binaire pour éviter de les cracker trop facilement.

***Les balises :***

**<input type= « text » name= « xx »>** permet d’avoir une zone de texte pour taper ses informations.

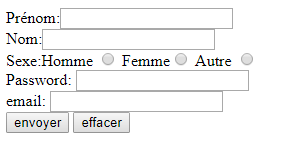
**<input type= « radio » name= « xx »>** permet d’avoir une zone à cocher. Il faut mettre le même name= « xx » à chaque donnée radio pour n’avoir qu’une seule possibilité de cocher.

**<input type= « password » name= « xx »>** permet d’avoir une zone ou on écrit en étoiles pour pas que les gens derrière voient ce que l’on écrit. Utile pour les mots de passe.

**<input type= « email » name= « email » required>** ajouté par exemple dans la balise de l’email ici signifie que l’information est obligatoire à remplir pour que le formulaire puisse s’envoyer. La balise « email » vérifie que la donnée envoyée est une adresse mail.

**<input type= « submit » value= « envoyer»>** permet de créer des boutons pour envoyer ses informations.

**<input type= « submit » value= « effacer »>** permet d’effacer les infos saisies.



**Il existe d’autres types de formulaires possibles, comme par exemple un numéro de tel ou une date d’anniversaire. Les formulaires se trouvent facilement online.**

Pour faire un un formulaire défilant (par exemple pour sélectionner un jour de la semaine) :

Jour de la semaine<select name="nom" size="1" >

<option>lundi</option> <option>mardi</option><option>mercredi</option> <option>jeudi

<option>vendredi</option> <option>samedi</option><option>dimanche</option>

</select> <br>

Ici, ce n’est pas un **<select type= « »** mais un **<select name= « nom ».**

Le size sert à montrer combien de choix seront affichés en même temps., et chaque option doit être entre les balises <option> </option>. Il faut bien évidemment refermer la balise </select>.

Mettre <option selected> vendredi </option> fera que le menu déroulant n’affichera pas la première option en premier (ici lundi) mais commencera par l’option choisie, donc par exemple vendredi.

On obtient :



Pour créer une zone de texte pour par exemple laisser un commentaire :

<textarea name="commentaire" rows="10" cols="40"></textarea>

Rows équivaut au nombre de lignes visibles et cols équivaut au nombre de caractères par lignes. Il ne faut rien écrire entre les balises <textarea>.

