|  |  |  |
| --- | --- | --- |
| Numérique et Sciences Informatiques | | |
| 3-4h | **Web : HTML + CSS** |  |
| **Objectif** : savoir écrire un site Web en HTML + CSS | | |
| **Matériel**: éditeur Notepad++ et navigateur web Chrome (quelques problèmes dans Firefox !) | | |

Le **web** est une application d’internet caractérisée par des sites comportant des **liens hypertexte** permettant de passer d'une page à une autre ou d’un site à un autre.

Le W3C (World Wide Web Consortium) a standardisé le **HTML** (fond), le **CSS** (forme) et le **JavaScript** (interactions) pour écrire des sites web.

Les navigateurs web (Firefox, Chrome, Safari ...) décodent le HTML, CSS et JavaScript.

Le **HTML** (HyperText Markup Language) est un langage balisé permettant de décrire le contenu d’une page tandis que le **CSS** (Cascading Style Sheets) est un langage composé de feuilles de style permettant de décrire la présentation d’une page.

Les **HTML** et le **CSS** sont des **langages descriptifs** contrairement à **Python** qui est un **langage algorithmique** (if, for, while).

**Structure du site que nous allons créer**

Dans le répertoire **Activités WEB** vous trouverez un répertoire **V1** représentant une première version de notre site internet.

Notre site est pour le moment composé d’une seule page appelée **index.html**.

Il est conseillé pour tous les sites d’avoir une page appelée **index.html** car c’est la page qui s’affichera par défaut : https://meteofrance.com/ = https://meteofrance.com/index.html

Vous allez ouvrir cette page dans le navigateur chrome (ou autres) et constater qu’elle comporte un contenu pas très lisible.

Vous allez maintenant ouvrir le fichier dans un éditeur de texte (clic droit → edit with **Notepad++**) et constatez qu'on a un texte formaté avec des balises que l’on reconnaît facilement car elles sont encadrées par des chevrons, exemple : **<body>**

**Notepad++** sert uniquement d'éditeur, pour vérifier si le code est valide il y a 2 solutions :

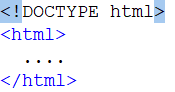
* Ouvrir le fichier dans un navigateur.
* Ouvrir le code dans un validateur (≈ décodeur) : https://validator.w3.org/

Cette page écrite en **html 5** est déjà en partie structurée pour avoir un site "classique" donc voici les différentes parties :

**Balise <html>**

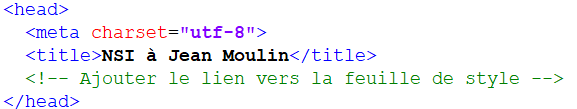
Il faut commencer par informer le navigateur que le langage utilisé est le html : **<!DOCTYPE html>**.

L’ensemble du code sera ensuite encadré par les balises **<html>** et **</html>** permettant de savoir où commence notre code et où il se termine.



**Balise <head>**

La première partie de notre page appelée **<head>** concerne les métadonnées (= informations concernant la page) :



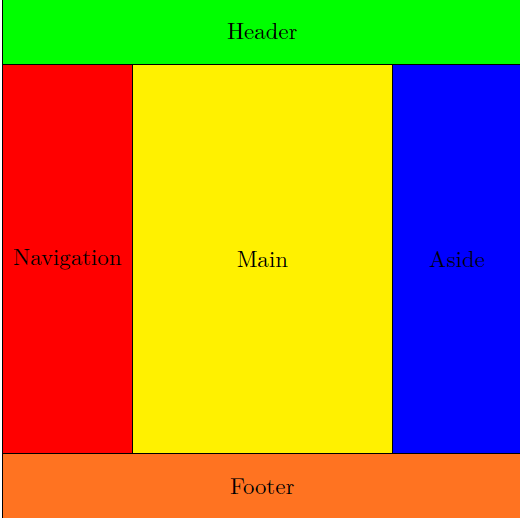
Ici nous donnons l’encodage du texte (utf-8) ainsi que le titre de la page qui sera affiché dans l'onglet du navigateur.

Ainsi qu’un commentaire nous demandant d’ajouter un lien vers la feuille de style que nous ferons plus tard.

**Balise <body>**

La balise **<body>** définit le contenu de la page (ce qui va s’afficher dans le navigateur).

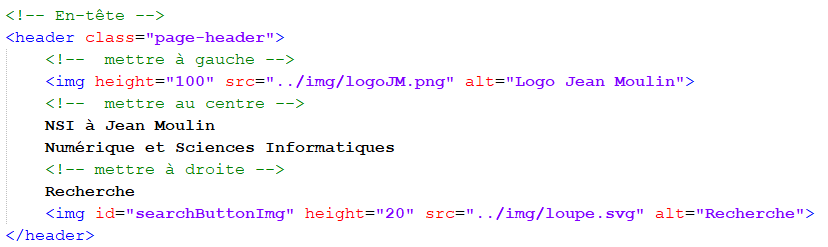
Nous avons choisi une organisation classique du site sous la forme suivante (5 zones) :



**Balise <header>**

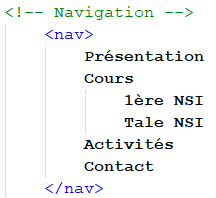
La balise **<header>** contient l’en-tête de notre page.

Dans notre cas, il y a le logo du lycée, le titre du site et un logo de recherche.



**Balise <nav>**

La balise **<nav>** contient le menu de navigation permettant d’avoir un lien direct vers les pages principales du site.

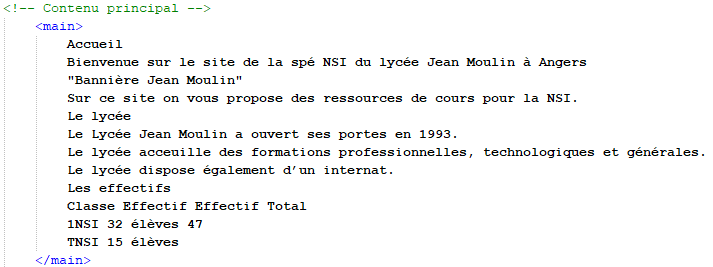


PS : on voit que le texte ne s'affiche pas vraiment à gauche dans le navigateur, il faudra un fichier **CSS** pour cela !

**Balise <main>**

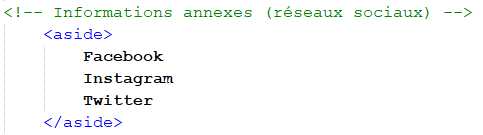
La balise **<main>** contient le contenu principal du site et changera en fonction des pages que l’on souhaitera afficher dans le site (c’est le seul contenu qui changera sur le site, les autres balises contiendront le même code sur toutes les pages).

Sur **index.html** on trouvera le texte présentant le lycée Jean Moulin.



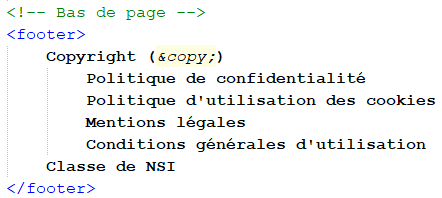
**Balise <aside>**

La balise **<aside>** permet de mettre des informations complémentaires sur la page, nous l’utiliserons pour mettre des liens vers certains réseaux sociaux.



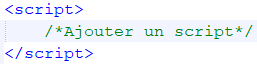
**Balise <footer>**

La balise **<footer>** désigne le pied de page et contient la plupart du temps les informations concernant les copyrights, les conditions générales de ventes ...



**Balise <script>**

La balise **<script>** permettra d’intégrer par la suite du code JavaScript afin de réaliser des interactions.



**Pour mettre en forme votre page, suivez les étapes suivantes :**

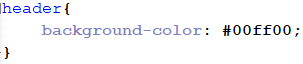
**Etape 1** : pour faire appel à la feuille de style (fichier CSS), ajoutez la ligne suivante dans la partie head :



Vous allez ensuite attribuer une couleur de fond en RGB à chacune des balises précédentes.

En CSS, les couleurs RGB sont définies par leur code hexadécimal précédé d’un #.

Par exemple pour la couleur verte :



**Etape 2** : ouvrez le fichier **style.css** dans un éditeur de texte (clic droit → edit with Notepad++) puis

définissez les couleurs (voir aspect du site présenté plus haut) pour toutes les zones de la page (header, nav, main, aside, footer).

Taille de chacune de ces zones :

* **Header** : hauteur de 100 pixels et largeur de 100% (100% de la taille de l'écran).
* **Nav** : largeur de 200 pixels.
* **Main** : hauteur minimale de 1000 pixels, largeur minimale de 500 pixels et largeur maximale de 1000 pixels (les dimensions dépendront du contenu à afficher dans cette partie).
* **Aside** : largeur de 100 pixels.
* **Footer** : largeur de 100% (par défaut la hauteur sera le reste de l'écran).

On utilisera les propriétés **height** et **width** que l’on peut précéder pas **max-** ou **min-**.

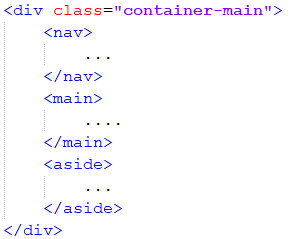
**Etape 3** : ajustez la taille des différentes zones.

On constate que les zones sont les unes après les autres, or on souhaite aligner les trois zones du centre (nav, main et aside).

Pour cela vous allez créer un **conteneur** qui contient les trois zones.

**Etape 4** : dans le fichier **index.html**, ajoutez une balise **<div>** qui commence avant la balise **<nav>** et termine après la balise **<aside>**.

Dans cette balise **<div>** créez le conteneur ayant le nom de class : **"container-main".**



On peut alors indiquer comment seront positionnés les éléments à l’intérieur de ce **conteneur**.

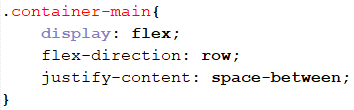
Ici nous choisissons de les afficher en ligne en mettant les propriétés :

**display: flex;** et **flex-direction: row;**.

Pour changer les propriétés d’une classe (et non d’une balise) il faut précéder le nom de la classe d’un point : .**container-main**

On ajoutera la propriété **justify-content: space-between**; pour que les 3 zones prennent la largeur de l'écran avec des espaces entre chaque zone.

Ce qui nous donne dans le CSS :

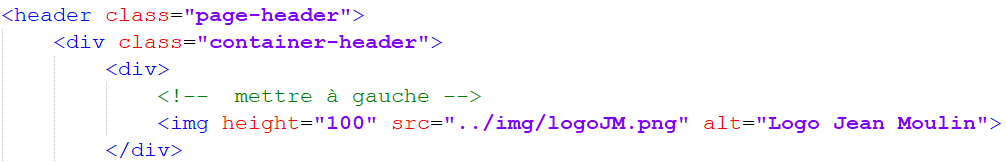


**Etape 5** : modifiez la page pour avoir les zones nav, main et aside en ligne (suivant la taille de l'écran il y aura des espaces entre chaque zone) :

|  |  |  |
| --- | --- | --- |
| **Partie nav** | **Partie main** | **Partie aside** |

Les balises <header>, <main>, <nav>, <aside> et <footer> sont équivalentes à des **<div>** mais avec des propriétés par défaut qui sont différentes.

On ajoutera des balises **<div>** dans le **<header>** pour le séparer en trois zones.



**Etape 6** : modifiez la page pour avoir un header en trois zones.

|  |  |  |
| --- | --- | --- |
| **logo du lycée** | **titre du site** | **bouton rechercher** |

**Nous en avons terminé avec la V1 de notre site.**

**Pour la suite, vous allez utiliser les fichiers présents dans le dossier V2 (= V1 corrigée).**

**Contenu du main**

Nous souhaitons obtenir le visuel suivant :



Ce qu'on observe :

* Deux gros titres : Accueil, Le lycée.
* Un plus petit titre : Les effectifs.
* Une image : panorama du lycée.
* Un tableau : tableau représentant les effectifs du lycée.
* Des paragraphes.

Pour avoir cette structure nous allons utiliser les balises suivantes dans le HTML :

* Niveaux de titres : <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
* Paragraphes : <p>
* Saut de ligne : <br>
* Image : <img>

Pour avoir la structure du tableau nous allons utiliser les balises suivantes dans le HTML :

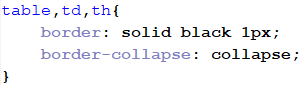
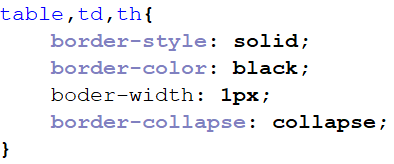
* Tableau : <table>
* Ligne du tableau (une ligne est constituée de cases) : <tr>
* Cellule d’entête (texte en gras) : <th>
* Cellule normale (texte pas en gras) : <td>
* Cellule normale constituée de plusieurs lignes (ex : 3 lignes) : <td rowspan="3">
* Titre (pas utile pour notre site) : <caption>

Il est possible d’ajouter des options à certaines balises comme par exemple, choisir la source de l’image et sa largeur :



**Etape 7** : modifiez le document **index.html** pour obtenir le visuel souhaité.

Pour ajouter les traits de lignes et de colonnes, il faudra ajouter des bordures aux éléments **table** dans le fichier CSS :

 ou 

On ajoutera la propriété border-collapse: collapse afin que les bordures de chaque élément soient fusionnées.

**Etape 8** : effectuez les modifications sur le fichier **style.css**.

**Menu de navigation**

Vous allez mettre en place un menu permettant d'accéder aux différentes pages HTML du répertoire **V2,** à l’aide de liens hypertexte.

Pour créer un lien hypertexte il faut utiliser la balise **<a>** dans laquelle on indiquera le lien de renvoie et le texte que l’on souhaite afficher sur l'hypertexte.

Par exemple dans **nav** on peut créer le lien suivant :



presentation.html (= ./presentation.html) étant la page où l’on souhaite aller et **Présentation** le texte que l’on souhaite afficher sur l'hypertexte.

Remarque sur ./ :

* le point signifie que l’on cherche dans le répertoire courant
* le slash permet de désigner la suite du chemin à suivre.

Pour revenir dans le répertoire précédent il faut utiliser ../ c’est ce que l’on fera pour récupérer des images dans le répertoire **img**.

**Etape 9** : créez le menu de navigation vers les autres pages du site.

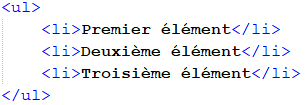
Plus tard il faudra créez le même menu sur toutes les pages pour pouvoir naviguer à l'intérieur de notre site.

Toutes les pages HTML seront identiques dans notre site, sauf pour la partie main.

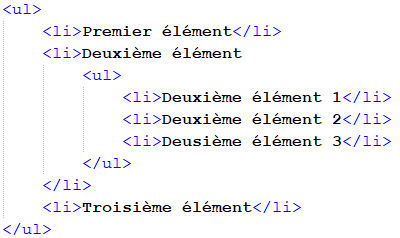
Afin d’avoir une présentation qui ressemble un peu plus à un menu nous allons utiliser des listes.

* Balise <ul> : définit où commence et où se termine la liste .
* Balise <li> : définit où commence et se termine un élément de la liste.

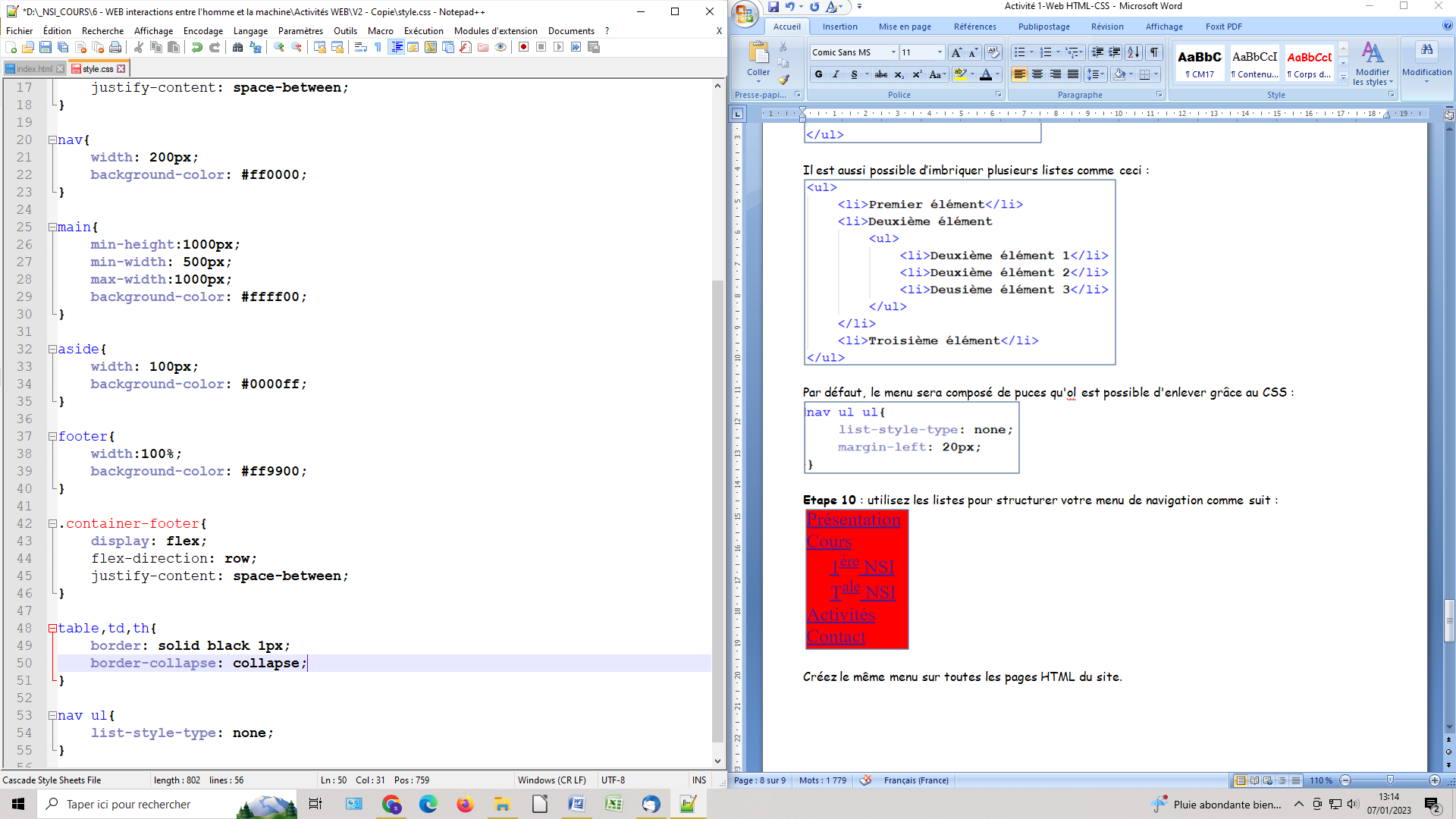
Exemple :



Il est aussi possible d’imbriquer plusieurs listes comme ceci :



Par défaut, le menu sera composé de puces qu'ol est possible d'enlever grâce au CSS :



**Etape 10** : utilisez les listes pour structurer votre menu de navigation comme suit :



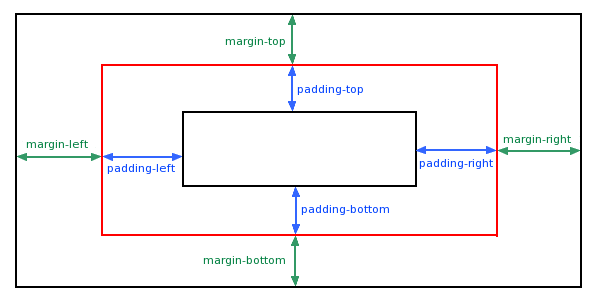
Créez le même menu sur toutes les pages HTML du site.

**Marges et écarts**

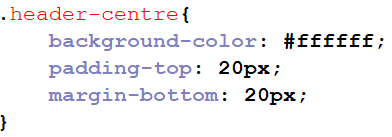
Pour aérer un peu notre page nous allons ajouter des marges et des écarts.

* Les marges (margin) correspondent à l’espacement à l’extérieur d’un bloc.
* L’écart (padding) correspond à l’espacement à l’intérieur d’un bloc.

Exemple de margin et padding sur le bloc rouge :



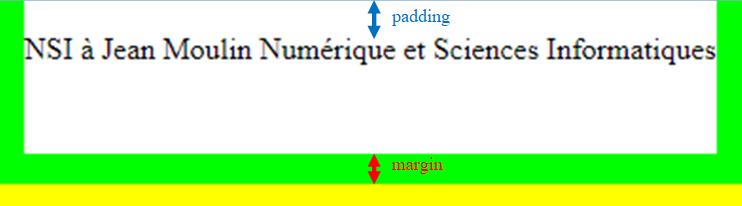
**Etape 11** : ajoutez les propriétés suivantes à votre feuille de style :



→ bloc **header-centre** devient blanc.

→ écart (padding) en haut de 20 pixels → les 20 pixels appartiennent au bloc header-centre et restent blanc.

→ marge (margin) de 20 pixels en bas → les 20 pixels du bas gardent la couleur du bloc container-header donc vert.



Par défaut, il existe souvent une marge et un écart sur les balises réservées comme <html>, <body> ...

**Etape 12** : améliorez la mise en page du site avec les marges et les écarts de votre choix.