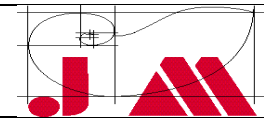


Numérique et Sciences Informatiques		
4h	Web : HTML + CSS	
Objectif : savoir écrire un site Web en HTML + CSS		
Matériel : éditeur Notepad++ et navigateur web		

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

Un site internet est composé de pages pouvant être écrites dans différents langages notamment HTML et CSS.

Les deux principaux langages utilisés dans le web sont le HTML et le CSS.

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.

L'alliance de ces deux langages permet de représenter la majorité des sites internet présents sur le web.

Les HTML et le CSS sont donc des **langages descriptifs** contrairement au Python qui est un **langage algorithmique**.

Structure du site

Dans le répertoire introduction au 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 pour que celui-ci soit facilement accessible à partir d'internet, c'est la page qui s'affichera par défaut.

Nous allons donc ouvrir cette page dans notre navigateur chrome. On constate qu'elle comporte un contenu mais qu'il n'est pas très lisible... Nous allons donc ouvrir une seconde fois ce fichier avec cette fois-ci un éditeur de texte. Pour cela nous faisons : clic droit ! ouvrir avec ! Notepad++. Nous obtenons un texte formaté avec des balises (tags) que l'on reconnaît facilement car encadrées par des chevrons "<" et ">".

Cette page écrite en html 5 est déjà en partie structurée pour avoir un site "classique". Voici une description rapide des différentes parties de notre site.

Balise html

Pour commencer à écrire notre code il faut commencer par informer le navigateur du langage utilisé avec la balise <!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.

```
<!DOCTYPE html>
<html>
    ...
</html>
```

Balise head

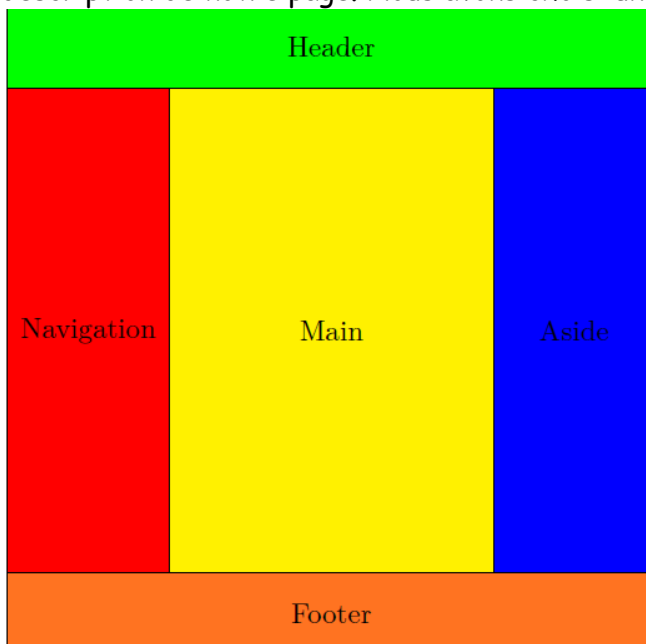
La première partie de notre page appelée head concerne les métadonnées (des informations concernant la page mais qui n'apparaissent pas dans son contenu) :

```
<head>
  <meta charset="utf-8">
  <title>NSI à Jean Moulin</title>
  <!-- Ajouter le lien vers la feuille de style -->
</head>
```

Ici nous donnons simplement l'encodage du texte (que vous connaissez maintenant utf-8) ainsi que le titre de la page qui sera affiché dans les onglets du navigateur. Ainsi qu'un commentaire nous demandant d'ajouter un lien vers la feuille de style que nous ferons dans la partie suivante.

Balise body

La balise body définit le contenu de la page (ce qui va s'afficher dans le navigateur). On va y placer la description de notre page. Nous avons choisi une organisation classique du site sous cette forme :



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.

```
<!-- En-tête -->
<header class="page-header">
  <!-- mettre à gauche -->
  
  <!-- mettre au centre -->
  NSI à Jean Moulin
  Numérique et Sciences Informatiques
  <!-- mettre à droite -->
  Recherche
  
</header>
```

Balise nav

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

```
<!-- Navigation -->
<nav>
    Présentation
    Cours
        1ère NSI
        Tale NSI
    Activités
    Contact
</nav>
```

Balise main

La balise main contient le contenu principal du site et changera en fonction des pages que l'on souhaite afficher (c'est le seul contenu qui changera, les autres balises contiendront le même code sur toutes les pages). On y mettra le texte concernant la page qui nous intéresse, pour celle-ci on y mettra la page d'accueil dont le texte présente le lycée Jean Moulin, car c'est la page par défaut.

```
<!-- Contenu principal -->
<main>
    Accueil
    Bienvenue sur le site de la spé NSI du lycée Jean Moulin (Angers)
    "Bannière Jean Moulin"
    Sur ce site on vous propose des ressources de cours pour la NSI.
    Le lycée
    Le Lycée Jean Moulin a ouvert ses portes en 1993.
    Le lycée est un lycée polyvalent qui accueille des formations professionnelles, technologiques et générales.
    Le lycée dispose également d'un internat.
    Les effectifs
    Effectif du lycée Jean Moulin à Angers
    Niveau Classe Effectif Effectif Total
    Seconde 2nde1 32 élèves 120
    2nde2 30 élèves 2nde3 28 élèves
</main>
```

Balise aside

La balise aside permet de mettre des informations complémentaires sur la page, nous l'utiliserons pour mettre des liens vers les principaux réseaux sociaux.

```
<!-- Informations annexes (réseaux sociaux) -->
<aside>
    Facebook
    Instagram
    Twitter
</aside>
```

Balise footer

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

```
<!-- Bas de page -->
<footer>
    Copyright (&copy;)
    Politique de confidentialité
    Politique d'utilisation des cookies
    Mentions légales
    Conditions générales d'utilisation
    Classe de NSI
</footer>
```

Balise script

La balise script permet d'intégrer par la suite du code Javascript (elle reste vide pour le moment).

```
<script>
    /*Ajouter un script*/
</script>
```

Pour commencer à travailler, nous allons mettre en forme notre page afin d'y voir plus clair. Nous allons pour cela créer des blocs de couleurs correspondant à chacune des parties de notre page et les placer à l'endroit souhaité.

Pour cela nous allons modifier la feuille de style qui s'appelle style.css en commençant par l'ajouter à notre fichier index.html.

Etape 1 : ajoutez la ligne suivante dans la partie head :

```
<link rel="stylesheet" href="style.css">
```

Cette ligne indique à notre page d'aller chercher les informations sur le style de notre page dans ce fichier. Nous allons ensuite attribuer une couleur en rgb à chacune des balises précédentes. En CSS, les couleurs RGB sont définies par leur code hexadécimal précédé d'un #. Par exemple :

```
header{
    background-color: #00ff00;
}
```

Pour définir la couleur de fond du header en vert.

Etape 2 : ajoutez une couleur différente pour toutes les parties de la page (header, nav, main, aside, footer)

Nous allons ensuite définir la taille de chacune de ces parties avec les informations suivantes :

- Header : hauteur de 100 pixels, largeur de 100%
- Nav : largeur de 200 pixels
- Main : la hauteur minimale est de 1000 pixels, la largeur minimale de 500 pixels et la largeur maximale de 1000 pixels. (Cela dépendra du contenu).
- Aside : largeur de 100 pixels
- Footer : largeur de 100%

On peut utiliser pour cela les propriétés height et width que l'on peut précéder par max- ou min-.

Etape 3 : modifiez le style pour ajuster les éléments à la bonne taille On constate que les éléments sont les uns au dessus des autres. On souhaite aligner les trois éléments du centre (nav, main et aside). Pour cela nous allons créer un conteneur qui contient les trois éléments et définir comment afficher les éléments au sein de ce conteneur.

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

On donnera un nom de classe à cette balise, par exemple "container-main".

```
<div class="container-main">
  <nav>
    ...
  </nav>
  <main>
    ....
  </main>
  <aside>
    ...
  </aside>
</div>
```

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 la propriété display: flex; et flex-direction: row;. Pour changer les propriété d'une classe (et non d'une balise) il faut précéder le nom d'un point.

Ce qui nous donne dans le CSS :

```
.container-main{
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}
```

On peut ajouter la propriété justify-content: space-between; pour espacer les blocs.

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

On pourrait les remplacer par nos propres div. C'est d'ailleurs ce que nous allons faire avec notre header en le séparant en trois éléments header-gauche, header-centre et header droite.

Etape 5 : modifiez la page pour avoir un header en trois parties. A gauche le logo du lycée, au milieu le titre du site et à droite le bouton rechercher.


Nous en avons terminé avec la V1 de notre site. Pour la suite, nous allons travailler sur les fichiers présents dans le dossier V2.

Contenu du main

Pour cela, nous commençons par structurer le texte principal de notre page, celui présent entre les balises main. Nous souhaitons obtenir la structure suivante :

Accueil

Bienvenue sur le site de la NSI de Jean Moulin à Angers



Sur ce site on vous propose des ressources de cours pour la NSI.

Le lycée

Le Lycée Jean Moulin a ouvert ses portes en 1993. Le lycée est un lycée polyvalent qui accueille des formations professionnelles, technologiques et générales. Le lycée dispose également d'un internat.

Les effectifs

Effectif du lycée Jean Moulin à Angers

Niveau	Classe	Effectif	Effectif Total
Seconde	2nde1	32 élèves	120
	2nde2	30 élèves	
	2nde3	28 élèves	

On y voit plusieurs choses :

- Des gros titres : Accueil, Le lycée
- Des plus petits titres : Les effectifs
- Une image : Panorama du lycée
- Un tableau : Tableau représentant les effectifs du lycée
- Des paragraphes

Pour retrouver la même structure nous allons utiliser les balises suivantes :

- Niveaux de titres : <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- Paragraphes : <p>
- Retour à la ligne :

- Image :
- Tableau : <table>
- Nouvelle ligne : <tr>
- Nouvelle cellule : <td>
- Nouvelle cellule avec par exemple 3 lignes : <td rowspan="3">
- Cellule d'entête : <th>
- Titre du tableau : <caption>

Il est possible d'ajouter des options à certaines balises. Par exemple, choisir la source de l'image et sa largeur avec :

```

```

Etape 6 : modifiez le document index.html pour obtenir le même résultat.

Notre tableau n'est toujours pas très lisible. Il est possible de le rendre plus beau en ajoutant des propriétés au fichier

style.css. Il faut pour cela ajouter des bordures aux éléments table,

```
table,td,th{  
    border: solid black 1px;  
    border-collapse: collapse;  
}
```

ou

```
table,td,th{  
    border-style: solid  
    border-color: black;  
    border-width: 1px;  
    border-collapse: collapse;  
}
```

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

Etape 7 : effectuez les modifications sur le tableau

Menu de navigation

Nous avons maintenant plusieurs pages dans notre répertoire V2. Nous allons donc mettre en place un menu permettant de se déplacer entre ces différentes pages à l'aide de liens hypertexte. Pour créer un lien hypertexte il faut utiliser la balise <a> dans laquelle on précisera le lien vers lequel on veut naviguer et le texte que l'on souhaite afficher. Par exemple dans nav on peut créer le lien suivant :

```
<li><a href="contact.html">Contact</a></li>
```

./contact.html (= contact.html) étant la page que l'on souhaite atteindre et Contact le texte que l'on souhaite afficher. Vous remarquerez que nous avons précédé le nom de la page par ./ le point signifie que l'on cherche dans le répertoire courant et le slash permet de désigner la suite du chemin à suivre (notamment lorsque l'on navigue dans des répertoires imbriqués). 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 8 : effectuez le même traitement avec toutes les pages.

Si vous testez la navigation vers une autre page vous remarquerez qu'il n'est pas possible d'utiliser les liens définis auparavant. La raison est que vous n'avez modifié le menu que pour une page. Il faudrait alors faire le même traitement sur toutes les pages. En soit toutes les pages seront identiques dans notre site sauf pour la partie main qui dépendra de la page. (Vous pourrez faire les modifications plus tard).

Afin d'avoir une présentation ressemblant un peu plus à un menu nous allons utiliser des listes. Pour cela nous allons utiliser les balises et . La première balise permet de définir où commence et où se termine la liste et le second permet de dire où commence et se termine un élément de la liste.

Exemple :

```
<ul>
  <li>Premier élément</li>
  <li>Deuxième élément</li>
  <li>Troisième élément</li>
</ul>
```

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

```
<ul>
  <li>Premier élément</li>
  <li>Deuxième élément
    <ul>
      <li>Deuxième élément 1</li>
      <li>Deuxième élément 2</li>
      <li>Deuxième élément 3</li>
    </ul>
  </li>
  <li>Troisième élément</li>
</ul>
```

Etape 9 : utilisez les listes pour structurer votre menu de navigation comme suit :



Pour rendre le menu plus agréable il est possible d'enlever les puces dans la feuille de style en ajoutant une propriété pour les listes de <nav> uniquement. Pour cela on le spécifie ainsi :

```
nav ul ul{
  list-style-type: none;
  margin-left: 20px;
}
```

Marges et écarts

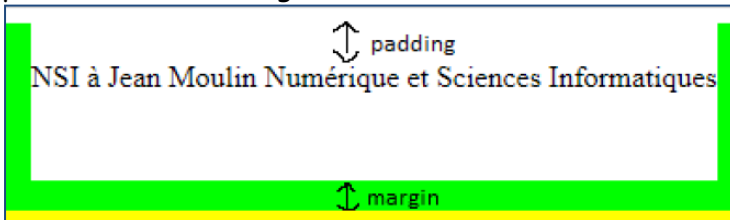
Pour aérer un peu notre page nous allons ajouter des marges et des écarts. Mais avant ça, quelle est la différence ?

Les marges (margin) correspondent à l'espacement à l'extérieur d'un bloc tandis que l'écart (padding) correspond à l'espacement à l'intérieur d'un bloc. Voici un exemple simple illustrant la différence.

Ajouter les propriétés suivantes à votre feuille de style :


```
.container-main{  
  background-color: #000000;  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
}
```

Nous avons ainsi changé le fond du bloc header-centre en blanc en ajoutant un écart en haut de 20 pixels et une marge de 20 pixels en bas. Les 20 pixels appartiennent donc au bloc header-centre et restent blancs tandis que les 20 pixels du bas gardent la couleur du bloc container-header. Comme on peut le voir sur l'image :



Par défaut, il existe souvent une marge et un écart sur les balises réservées comme `<html>`, `<body>`, ... Il peut de temps en temps être intéressant de mettre le padding et le margin à 0 au début d'un projet pour éviter que le placement des éléments ne soit pas parfait (à quelques pixels près).

Etape 10 : à vous d'améliorer à votre guise la mise en page de cette V2 !