# La mise en forme avec CSS

## Lier le CSS au html

Pour lier le fichier HTML au fichier CSS, il suffit de rajouter un lien spécifique dans le *head* du fichier html :

```
<head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" />
    <title>SIN - LPO ARAGON PICASSO</title>
</head>
```

En prenant au chemin relatif qui mène au fichier.

## Syntaxe

Le fichier CSS est composé de plusieurs règles qui permettent de modifier l'affichage de une ou plusieurs balises HTML.

```
Sélecteur {
     propriete_1 : valeur_1 ;
     propriete_n : valeur_n ;
}
```

## Sélecteurs

La nature du sélecteur permet de trier les éléments sur lesquels vous allez appliquer les propriétés. Ils peuvent être :

- Universel : ***

```
* {
  color : blue;
}
```

- Balises de premier niveau et d'architecture : ***body, header, footer,…***

```
body
{
  background-color: grey;
  font-family: "Arial", sans-serif;
  color: #181818;
}
```

- Balises générales : ***p, a, nav, article,…*** (connues du langage html)

- Classes : ***objets que vous définissez ayant certaines propriétés,***

```
.logo
{
    display: flex;
    align-items: center;
}
```

Rque : La syntaxe **p.logo** affecte uniquement les <p> de classe « center » : <p class=''center''>..

- Id : ***ressemblant aux classes mais faisant référence à un objet unique du code HTML.***

```
#banniere_image {
  display: flex;
  justify-content: center;
  width : auto;
  height: 250px;
}
```

## Regroupement de sélecteurs

On peut appliquer le même paramètre a plusieurs sélecteurs en les regroupant dans une même instruction, simplement séparés par une virgule.

```
article,  aside
{
    text-align: justify;
}
```

## Sélecteur de descendants

On cherchera souvent à toucher des éléments qui sont inclus dans un autre.

Exemple : tous les liens (balise ***a***) qui sont dans le menu (balise ***nav***) mais uniquement ceux-là.

Dans ce cas, on imbrique les sélecteurs :

```
nav a
{
    font-size: 1.3em;
    color: #2f2d29;
    padding-bottom: 3px;
    text-decoration: none;
}
```

## Sélecteurs d'enfants

Cela ressemble aux sélecteurs de descendants mais il s'agit ici de descendants directs (de premier niveau).

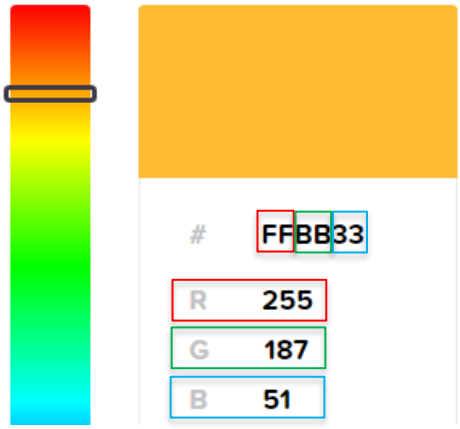
```
nav>a
{
    font-size: 1.3em;
    color: #2f2d29;
    padding-bottom: 3px;
    text-decoration: none;
}
```

## Codage des couleurs

En html, les couleurs sont codées sur 24 bits (3 octets) et utilise les composantes de base RVB (rouge, vert, bleu). Ceci nous permet d'obtenir $2^{24}$ possibilités, ce qui correspond à plus de 16,7 millions de nuances….

La valeur de chaque couleur peut donc varier de :

| | | | |
|---|---|---|---|
| En binaire : 0000 0000 | | à | 1111 1111 |
| En décimal : | 0 | à | 255 |
| En hexadécimal : | 0 | à | FF |




De nombreux sites permettent d'obtenir le code d'une couleur :
https://www.w3schools.com/colors/colors_picker.asp

```css
body
{
  background-color: grey;
  color: #181818;
}
```

Si on a un peu du mal avec l'hexadécimal, il est toujours possible de note les couleurs en décimal avec la commande :

***color : rgb(255 ,0 ,255) ;*** à la place de ***color : #FF00FF ;***

***ou*** ***color :rgb(100%,0%,100%) ;***

On peut également rajouter de la transparence en utilisant la commande :

**color : rgba(r,v,b,alpha_en_%) ;**

La propriété **color** peut s'appliquer à du texte, à des bordures, des cadres, à des blocs en arrière-plan.

Ouvrir et compléter le fichier **style.css,** en renseignant les couleurs du sélecteur **body**.
Compléter les couleurs de liens (classe **.liens**) et leur couleur en cas de survol ( **:hover**)

## Formatage du texte

En html, les balises <h1>, <h2> , … permettent de mettre en place une hiérarchie dans le formatage du texte.

### Police

```css
body
{
  background-color: grey;
  font-family: "Arial", "Helvetica",  sans-serif;
  color: #181818;
}
```

La propriété « font-family » définit la ou les polices à utiliser pour votre page. Le navigateur sélectionne première qu'il trouve installer. On finit souvent par une police générique.

Compléter les données sur la police du fichier html en complétant le sélecteur **body.**

### Style

La propriété « ***font-style*** » permet de définir une police « normal », « italic » ou « oblique ».
La propriété « ***font-weight*** » permet de définir une police « normal », « bold ».

### Taille

En informatique, on donne souvent la taille des éléments graphiques en pixel, reliquat des imprimantes d'antan... Le souci est que la taille d'un pixel peut varier en fonction du diffuseur (écran HD, 4K, mobile, retina, …).
On parle alors de taille absolue (cm, px) ou relative (rem, em, %).

Pour cela, il vaut mieux éviter d'utiliser cette dimension **absolue** et préférer une unité relative : **le rem** : on spécifie la taille d'un élément en fonction de la dimension par défaut du navigateur. Sur les ordinateurs, cette valeur est souvent de 16px.

| | | | |
|---|---|---|---|
| Donc si on note : | **1rem** | ça signifie | **16px** sur un ordi |
| | **.5rem** | | **8px** |

Différence entre **rem** et **em** : Si on veut spécifier une taille de police particulière pour le site, on peut rajouter dans le **body** une propriété :

```
font-size : X % ;
```

Ensuite on peut travailler en **em** qui dimensionne l'élément par rapport à ce X %.
**rem quant à lui reste basé sur la taille défaut du navigateur.**

Modifier le fichier CSS en remplaçant toutes les dimensions absolues (px) par des **rem** (petit calcul nécessaire).

### Alignement

La propriété « ***text-align*** » permet de définir la position du texte : « left », « center », « right » ou « justify ».
Modifier le fichier pour le texte contenu dans les blocs **article** et **aside** soit justifié et pour que le titre du bloc **aside** soit centré horizontalement.
Compléter le fichier pour que le titre de l'Id « **titre_principal** » soit en taille **3rem**

## Site source

Les propriétés sont très nombreuses et on ne peut pas toutes les parcourir, voici un site qui peut vous aider à chercher une propriété qui répond à votre besoin.

https://www.w3schools.com/css/default.asp

# Mise en page de la page HTML

Il existe beaucoup de méthode de mise en page. Ce qui suit n'est qu'une proposition…

On peut disposer la page en grille en utilisant le **grid layout** et ensuite on positionne les éléments de chaque bloc à l'aide du **flex layout**

## Disposition de la page en grille

Exercice : https://cssgridgarden.com/#fr
Pour cela on va définir une grille qui permettra de placer chacun de nos composants.

On déclare cette grille comme une classe « **.grille** » qui sera appelée dans une balise ** ** au niveau du fichier html.

```
<body>
    <div class="grille1">
      <header><!-- début du header -->
            <div id="titre_principal">
                <div class="logo">
```

1. On déclare d'abord le nombre de colonne que l'on souhaite avec **grid-template-columns.**

```
.grille1
{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-gap: 1rem;
  grid-template-areas:
    "h h h"
    "b b b"
    "ar ar as"
    "f f f";
}
```

Dans le cas 1, on a 3 colonnes de largeur identique. La largeur de colonne est donnée par **1fr.** On alloue une fraction de la largeur de l'écran à chaque colonne.

2. On spécifie l'espace entre chaque colonne.
3. **Grid-template-areas** permet de représenter le schéma de la page. Chaque bloc du html est placé dans la grille

```
header
{
  grid-area: h;
  display: flex;
  justify-content:space-around;
}
```

```
.banniere
{
  grid-area: b;
}
```

```
article
{
  grid-area: ar;
  margin-left: 1rem;
}
```

```
aside
{
  grid-area: as;
  margin-right: 1rem;
}
```

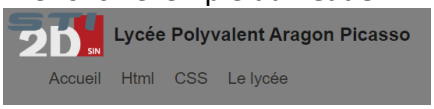
```
footer
{
  grid-area: f;
  padding: 1rem;
  display: flex;
  justify-content: space-between;
}
```

Placé dans chaque bloc ; la propriété **grid-area** permet de référencer le bloc concerné en rapport avec la grille.

Rien ne vous empêche de créer un autre format de grille pour modifier l'organisation de la page ou pour organiser une autre page différemment, en créant une autre classe **.grille2** qui sera appelée dans le html.
Compléter le fichier css en créant la classe **grille1** et en complétant les 5 balises qui composera notre page.

## Disposition interne de chaque bloc en flex

https://flexboxfroggy.com/#fr
Prenons l'exemple du **header.** En l'état actuel des choses, il ressemble à cela :




Mais je voudrais qu'il ressemble plutôt à cela

On remarque qu'il y a 2 blocs dans le **header**.

```
<header><!-- début du header -->
    <div id="titre_principal">
        <div class="logo">
            <img src="images/SIN_logo.png" alt="Logo SIN" />
            <h1>Lycée Polyvalent Aragon Picasso </h1>
        </div>
    </div>
    <nav><!-- Début du nav -->
        <ul>
            <li><a href = "index.html">Accueil</a></li>
            <li><a href = "page1.html">Html</a></li>
            <li><a href = "page2.html">CSS</a></li>
            <li><a href = "https://aragon-picasso.ent.auvergnerhonealpes.fr" target="_blank">Le lycée</a></li>
        </ul>
    </nav><!-- Fin du nav -->
</header><!-- Fin du header -->
```

Par défaut, ils sont l'un au-dessus de l'autre.

Pour modifier cela, déclarer un positionnement par **flex** pour le bloc **header :**

```
header
{
  grid-area: h;
  display: flex;
  justify-content: space-between;
}
```

Spécifier quelle disposition horizontale de flex on souhaite utiliser.

SI on essaie de diminuer la taille de la fenêtre, on se rend compte que le menu « se démonte » car l'espace se rétrécit. Pour permettre de garder une cohérence dans le menu, on peut rajouter une option **flex-wrap** afin de permettre au menu de basculer sur la ligne d'en dessous si besoin.

```
header
{
  grid-area: h;
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
}
```

Notre haut de page devrait ressembler à cela :


On remarque que le texte n'est pas centré verticalement sur l'image, ce qui n'est pas très esthétique.

```
<div class="logo">
    <img src="images/SIN_logo.png" alt="Logo SIN" />
    <h1>Lycée Polyvalent Aragon Picasso </h1>
</div>
```

En reprenant la même logique que pour le bloc **header**, dans le bloc **class = logo :**
Positionner verticalement les deux blocs (balises <img> et <h1>). On peut le faire avec la propriété **align-items** appliqué à la classe **logo.**

```
.logo
{
    display: flex;
    align-items: center;
}
```

**De même mettre en place un « flex-layout » pour les balises suivantes :**

- **footer**
- **.logo**
- **Nav ul (pour les éléments de la liste du menu)**
- **#banniere_image**
- **#social**
- **#listes_sites**