



## 1 Introduction :

**CSS**, pour ***Cascading Style Sheets***, (feuilles de style en cascade), est un langage conçu pour décrire le style d'un document **HTML**. Il décrit comment les éléments **HTML** vont être affichés à l'écran sur papier ou tout autre média.

Le **CSS** est là pour simplifier le travail. Pour un même contenu, il sera facile d'obtenir différentes versions en fonction du navigateur, de la taille de l'écran, du type de périphériques, ordinateur, tablette, smartphone.

Le style peut être inclus dans le code **HTML** comme cela a été le cas jusqu'à maintenant ou mieux regrouper dans un fichier **CSS** spécifique.

La feuille de style est incluse dans l'entête de la page **HTML** avec la balise **link**, « MonStyle.css » est ici le nom du fichier contenant toutes les déclarations pour cette page **HTML**.

```
<html>
  <head>
    <title>Module 1</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="MonStyle.css" rel="stylesheet" type="text/css"/>
  </head>
```

## 2 Syntaxe :

Le CSS correspond à un ensemble de propriétés appliqué à un sélecteur. Un sélecteur peut être désigné par un nom de balise HTML, un identifiant unique ou un ensemble d'éléments désigné par une classe.

### 2.1 Le sélecteur est une balise HTML

Diagramme illustrant la syntaxe CSS :

```

  Sélectionneur
  h1 {
    color: blue ;
    font-size: 12px ;
  }
  Valeur
  Propriété

```

Ici, le texte contenu entre les balises h1 aura la particularité d'avoir comme propriétés une couleur bleue et une taille de police de 12 pixels. Les propriétés sont séparées par un point-virgule ( ; ).

Lorsque le sélecteur est une balise HTML, le style s'applique à toutes les balises de même nom pour les pages ou le fichier CSS est inclus.

Exemple :

```

11  body
12  {
13    background-color: gray;
14  }
15
16  h1
17  {
18    color: brown;
19    font-family: serif;
20    font-size: 20px;
21    margin-left: 20px;
22  }
```

```

8  <html>
9  <head>
10 <title>TODO supply a title</title>
11 <meta charset="UTF-8">
12 <meta name="viewport" content="width=device-width, initial-scale=1.0">
13 <link href="MonStyle.css" rel="stylesheet" type="text/css"/>
14 </head>
15 <body>
16 <h1>
17   Voici le premier titre
18 </h1>
19 </body>
20 </html>
```

Voici le premier titre

## 2.2 Le sélecteur est un identifiant

L'**id** d'un élément HTML permet d'identifier de manière unique une balise HTML. Il ne peut être utilisé qu'une seule fois dans une même page.

```

20  <div id="menu">
21
22  </div>

```

Ici, le conteneur définit par la balise **div** à un identifiant spécifique **menu**. Il permet d'appliquer des propriétés particulières à cette portion de code et non pas à toutes les balises **div** de la page.

Dans le fichier CSS, l'identifiant est précédé par un dièse ( # ).

Exemple :

```

24  #menu
25  {
26      position: absolute;
27      top: 30px;
28      left: 5px;
29      width: auto;
30  }

```

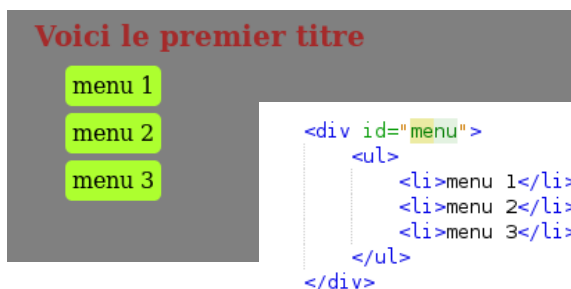
Dans cet exemple, le menu est placé toujours à 30 pixels du haut de page et 5 pixels de sa gauche. Sa largeur s'adapte de manière automatique à la largeur de la page.

Il est également possible de désigner une balise particulière dans cette section de code sans affecter les autres balises du même nom dans le reste de la page.

```

32  #menu>ul
33  {
34      list-style-type: none;
35  }
36
37  #menu>ul>li
38  {
39      padding: 5px;
40      background-color: greenyellow;
41      border-radius: 5px;
42      margin: 5px;
43  }

```



Ainsi, toutes les balises **ul** du menu peuvent avoir les mêmes propriétés sans pour autant affecter les autres balises **ul** de la page en dehors du conteneur **menu**. Ici, les items du menu n'ont plus de disque dans la liste.

Comme le montre l'exemple, il est possible d'ajouter des règles spécifiques à d'autres balises incluses dans les premières. Ici, les balises **li** incluses dans la balise **ul** du **menu** ont un aspect particulier.

## 2.3 Le sélecteur est une classe

Une classe CSS désigne un ensemble d'éléments HTML utilisant le même sélecteur. Dans le fichier CSS, le nom de la classe est précédé par un point ( . ).

Exemple :

```

53  .important
54  {
55      margin: 10px;
56      padding: 5px;
57      background-color: lightsteelblue;
58      border: 1px black solid;
59      text-align: left;
60  }

```

```

32
33  <p class="important">
34      Voci un texte important
35  </p>

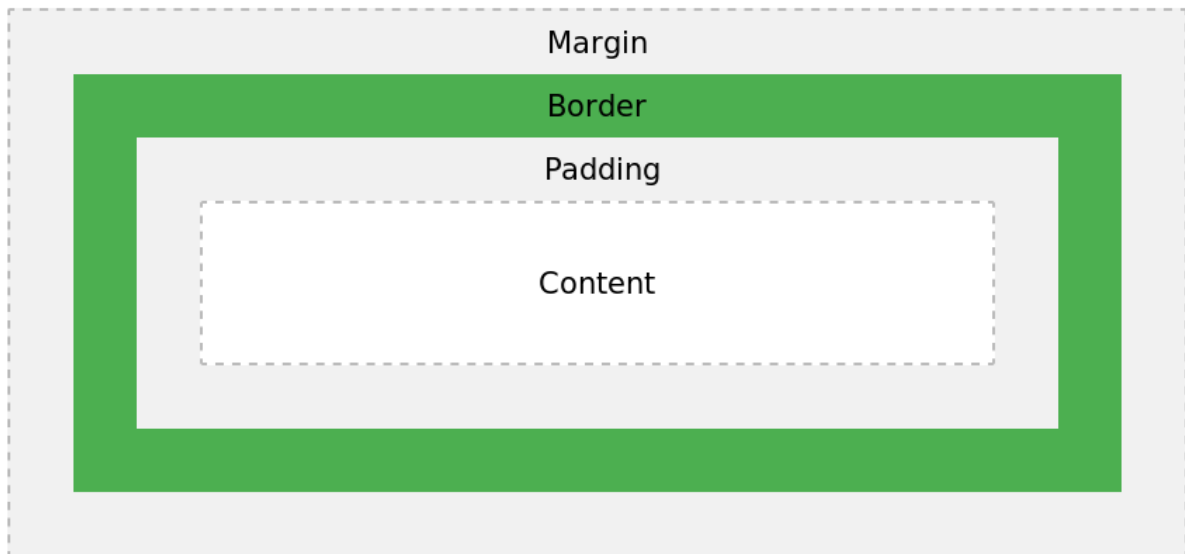
```

Le résultat obtenu est le suivant. Plusieurs paragraphes peuvent utiliser la classe « important » pour mettre en valeur le texte.

Voci un texte important

### 3 Modèle en boîte CSS :

Tout élément HTML peut être considéré comme une boîte, l'image suivante résume cette notion :



<b>Content</b>	<i>Contenu</i>	C'est le contenu de la boîte, cela peut être du texte, une image...
<b>Padding</b>	<i>Rembourrage</i>	C'est l'espace entre le contenu et la bordure. Il est transparent.
<b>Border</b>	<i>Bordure</i>	La bordure entoure le rembourrage et le contenu. Elle peut prendre différents aspects en termes de couleur de style de trait et d'épaisseur.
<b>Margin</b>	<i>Marge</i>	C'est l'espace entre la bordure et les autres éléments HTML adjacents à la boîte. Elle est transparente.

```

62  div
63  {
64      width: 300px;
65      border: 25px solid green;
66      padding: 25px;
67      margin: 25px;
68  }

```

Par défaut, les propriétés **margin** et **padding** agissent dans les quatre directions (haut, bas, gauche, droite) comme dans l'exemple ci-contre.

La propriété **width** comme dans l'exemple ci-contre précise la largeur, elle est exprimée en pixels ou en pourcentage. La propriété **height** quant à elle, détermine la hauteur si nécessaire.

Il est tout à fait possible de modifier la marge uniquement sur la gauche par exemple en agissant sur la propriété **margin-left**. Les quatre directions sont désignées par les termes suivants :

<b>top</b>	<i>haut</i>
<b>left</b>	<i>gauche</i>
<b>right</b>	<i>droite</i>
<b>bottom</b>	<i>bas</i>

```

48
49  #contenu {
50      margin-left: 110px;
51  }

```

← Ici on désigne explicitement la marge gauche.

```

70
71  p {
72      margin: 25px 50px 75px 100px;
73  }

```

← Ici on précise la valeur de chacune des marges (haut, gauche, droite, bas).

La propriété **border-radius** permet d'arrondir l'angle de la bordure. Elle est exprimée en pixels.

```

37  #menu>ul>li
38  {
39      padding: 5px;
40      background-color: greenyellow;
41      border-radius: 5px;
42      margin: 5px;
43  }

```

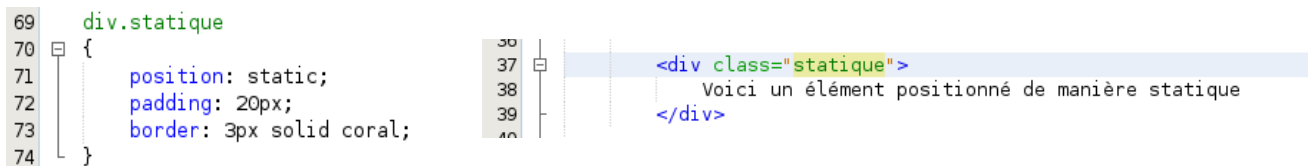
La propriété **background-color** agit sur la couleur de fond de la boîte.

## 4 Positionnement :

La propriété **position** permet de spécifier la manière dont un élément HTML est positionné sur l'écran. Elle peut être soit statique, relative, fixe, absolue ou collante.

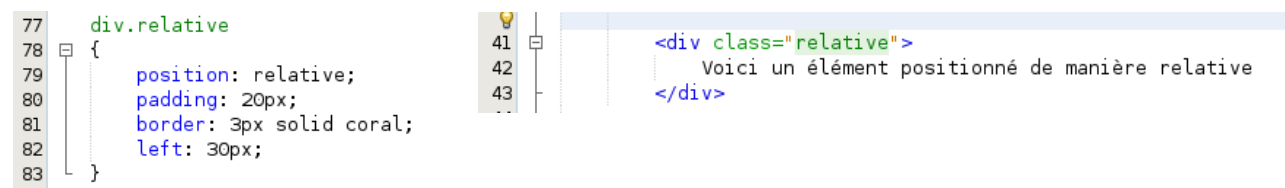
### 4.1 Position : static ;

Les éléments HTML sont positionnés statiquement par défaut. Ils ne sont pas affectés par les propriétés **top**, **bottom**, **left** et **right**. Dans ce cas, il n'est pas positionné de manière particulière, il est toujours positionné en fonction du flux normal de la page.



### 4.2 Position : relative ;

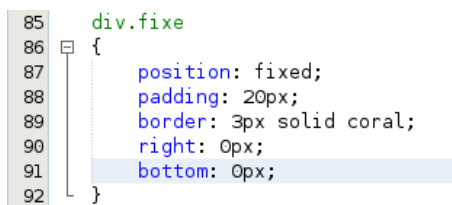
Un élément dont la propriété `position` possède la valeur `relative` est positionné de manière relative à la position qu'il aurait normalement occupée. Il est nécessaire d'agir sur les propriétés **top**, **bottom**, **left** ou **right**.



Un décalage de 30 pixels sur la gauche est ajouté par rapport à la position de la première phrase positionnée de manière statique.

### 4.3 Position : fixed ;

Un élément positionné de manière fixe l'est par rapport à la fenêtre d'affichage. Ainsi il reste au même endroit même si la fenêtre défile. Les propriétés **top**, **bottom**, **left** et **right** affectent la position de l'élément.



C'est la solution pour positionner un menu des boutons ou des liens toujours au même emplacement sur l'écran.

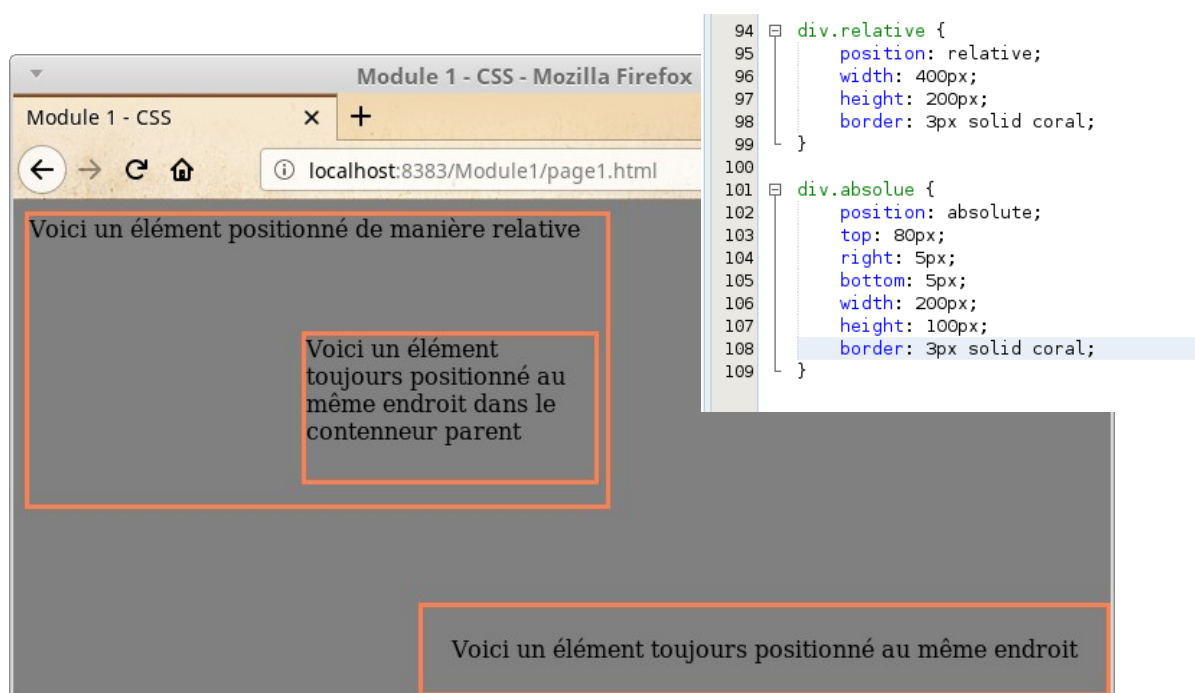
#### 4.4 Position : absolute ;

Un élément positionné de manière absolue l'est par rapport à l'élément qui le contient ou l'ancêtre le plus proche. Au contraire, un positionnement fixe positionne l'élément par rapport à la fenêtre d'affichage. Si un élément positionné en absolu n'a pas d'ancêtre, il est positionné par rapport à la page qui est affichée et est affecté par le défilement.

```

42 <div class="relative">
43   Voici un élément positionné de manière relative
44   <div class="absolue">
45     Voici un élément toujours positionné au même endroit dans le conteneur parent
46   </div>
47 </div>
48
49 <div class="fixe">
50   Voici un élément toujours positionné au même endroit
51 </div>

```



#### 4.5 Position : sticky ;

Un élément positionné de manière collante est placé en fonction de la position de défilement de l'utilisateur. Il bascule entre un positionnement relatif et fixe, en fonction de la position de défilement. Il est positionné de manière relative jusqu'à ce qu'une position de décalage donnée soit satisfaite dans la fenêtre d'affichage, puis il "reste" en place, comme pour la position fixe.

```

111 div.collante
112 {
113   position: sticky;
114   position: -webkit-sticky;
115   top: 30px;
116   padding: 5px;
117   background-color: beige;
118   border: 2px solid cadetblue;
119 }

```

La valeur **sticky** n'est pas toujours reconnue comme le montre l'éditeur de NetBeans. Il est parfois nécessaire de mettre **-webkit-sticky** à la place ou les deux pour que le code s'adapte à différents navigateurs.

## 4.6 Positionnement superposé

Lorsque plusieurs éléments se superposent, par exemple du texte sur une image, il est important de pouvoir choisir celui qui doit se trouver au premier plan. Pour cela, la propriété **z-index** répond à ce besoin.

Elle peut prendre des valeurs positive ou négative. La plus grande valeur sera au premier plan, la plus petite sera plus derrière.

```


121  img
122  {
123      position: absolute;
124      left: 0px;
125      top: 0px;
126      z-index: -1;
127  }

```

```

53  <h1>Voici le titre</h1>
54  
55  <p>
56      Comme l'image à un z-index à -1, Elle est placée derrière le texte.
57  </p>

```



Comme l'image à un z-index à -1, Elle est placée derrière le texte.

## 5 Alignement :

### 5.1 Alignement horizontal d'un texte

La propriété **text-align** permet d'aligner un texte horizontalement, soit sur la page soit dans le conteneur qui le reçoit.

Les valeurs courantes sont :

<code>text-align: center;</code>	Pour centrer le texte
<code>text-align: left;</code>	Pour l'aligner à gauche (valeur par défaut)
<code>text-align: right;</code>	Pour l'aligner à droite
<code>text-align: justify;</code>	Pour justifier le texte

### 5.2 Alignement vertical

La propriété **vertical-align** permet d'aligner un élément verticalement par exemple dans une cellule d'un tableau.

Les valeurs courantes sont :

<code>vertical-align: middle;</code>	Pour centrer sur la hauteur
<code>vertical-align: top;</code>	Pour l'aligner en haut (valeur par défaut)
<code>vertical-align: bottom;</code>	Pour l'aligner en bas

### 5.3 Centrer une image ou un conteneur

Voici deux exemples permettant de centrer en utilisant le réglage des marges de manière automatique. Pour cela, il faut fixer la largeur de l'élément, la valeur **auto** ajuste ensuite en fonction de la place disponible.

```

134  .centrer {
135      margin: auto;
136      width: 50%;
137      border: 3px solid green;
138      padding: 10px;
139  }

```

```

141  img {
142      margin-left: auto;
143      margin-right: auto;
144      width: 40%;
145  }

```

# Application

## Exercice n°1 – Style appliqué aux balises



A partir de l'application proposée lors du TD1 ou le CSS était mélangé avec le code HTML, réalisez un fichier style.css permettant de regrouper toutes les mises en forme effectuées en agissant sur les balises sur vos trois pages HTML.



Que pouvez-vous conclure ?

## Exercice n°2 – Utilisation de classes CSS



Dans un nouveau projet, codez une page WEB de manière à obtenir le rendu ci-dessous. Pour cela, dans un fichier nommé style.css, définissez 5 classes (titre, cadre1, cadre2, texte1 et texte2). Chaque partie du texte est contenue dans une **div** et affectée de la classe correspondante.

Quelques précisions :

- Le cadre de titre occupe 40% et est centré par rapport à la page, le style de la bordure est de type ridge.
- Le cadre de niveau 1 occupe 20%, le style de la bordure est de type groove.
- Le cadre de niveau 2 est décalé de 5 % sur la gauche
- Le texte de niveau 1 possède comme police de caractères : "Courier New" de hauteur 10 Pixels", il est également décalé de 5 % sur la gauche.
- Le texte de niveau 2 possède comme police de caractères : "Time New Roman", il est décalé de 10 % sur la gauche.

### Mon Premier CSS utilisant des classes

#### Cadre de niveau 1

du texte de niveau 1 avec une police de niveau 1

#### cadre de niveau 2

du texte de niveau 2 avec une police de niveau 2

#### Cadre de niveau 1

du texte de niveau 1 avec une police de niveau 1

#### cadre de niveau 2

du texte de niveau 2 avec une police de niveau 2

#### Cadre de niveau 1

du texte de niveau 1 avec une police de niveau 1

#### cadre de niveau 2

du texte de niveau 2 avec une police de niveau 2

## Exercice n°3 - Positionnement

Soit le code HTML suivant :

```
<!DOCTYPE html>

<html>
  <head>
    <title>Les cadres</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div id="cadreHautGauche">je suis dans le cadre haut gauche</div>
    <div id="cadreHautDroit">je suis dans le cadre haut droit</div>
    <div id="cadreBasGauche">je suis dans le cadre bas gauche</div>
    <div id="cadreBasDroit">je suis dans le cadre bas droit</div>
  </body>
</html>
```



Écrivez le fichier `positionnement.css` afin d'obtenir le rendu ci-dessous, même en cas d'agrandissement ou de diminution de la fenêtre. Complétez le fichier HTML pour faire référence à ce fichier.



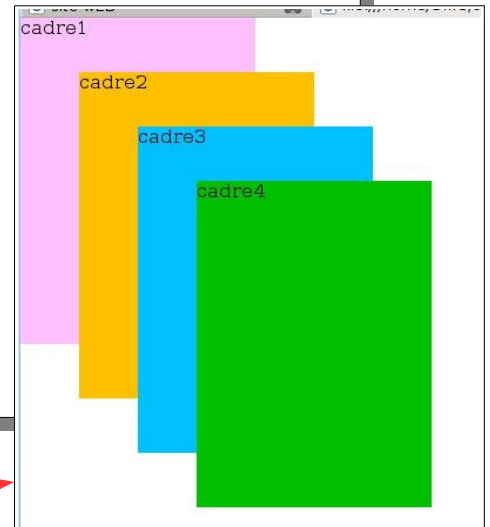




Complétez la feuille de style pour l'appliquer au code HTML d'une page est le suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="td.css">
  </head>
  <body>
    <div class="boite1">cadre1</div>
    <div class="boite3">cadre3</div>
    <div class="boite2">cadre2</div>
    <div class="boite4">cadre4</div>
  </body>
</html>
```

Résultat attendu



## Exercice N°4 – Utilisation des outils de développement du navigateur



Dans un nouveau projet **HTML5Application** sous NetBeans nommé **DrapeauxSimples**. Dans un premier temps, créez un nouveau dossier **images** dans le dossier **public\_html** de votre projet.

Téléchargez les ressources pour le TD et réalisez l'extraction du fichier, dans votre dossier **images**. Créez également un nouveau fichier **drapeauxStyle.css** vous devez obtenir le résultat suivant :



Recopiez ensuite le code proposé ci-dessus dans la page **index.html**, pensez à faire glisser les images de l'onglet Projets dans votre page de code.

Pour chaque alternative, ajoutez le nom du pays au cas où l'image ne serait pas présente comme le montre la ligne suivante.

```

```

Visualisez le résultat dans le navigateur Firefox.

Pour bien visualiser chaque image, dans le fichier **drapeauxStyle.css** ajoutez les lignes suivantes permettant de modifier la couleur de fond de la page.

```

body
{
  background-color: aquamarine;
}

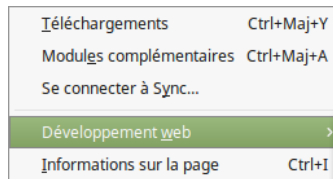
```

Ne pas oublier d'ajouter le lien vers le fichier CSS dans votre page d'index entre les balises **<head>** ... **</head>**. (faire glisser)

Voici le résultat attendu :

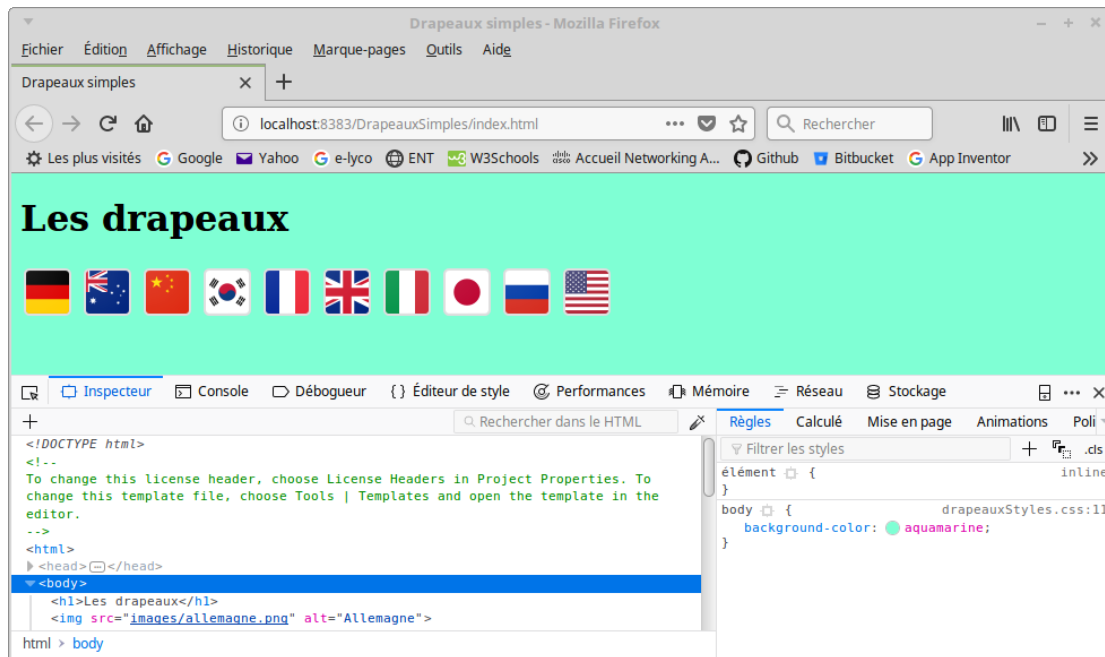


Dans le menu Outils de votre navigateur, observez les différents outils de développement Web à disposition :

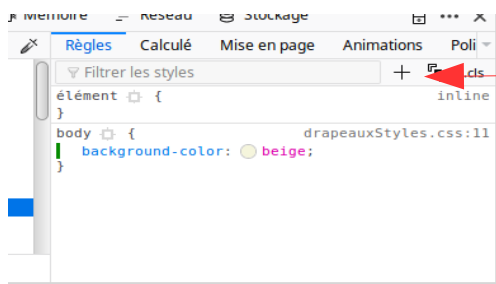


Cliquez sur **Outils de développement** ou faites **Ctrl Maj I** sur votre clavier ou encore appuyez sur la touche **F12**

La partie inférieure de la fenêtre contient maintenant l'outil de mise au point :



Cet outil permet d'agir en direct sur le style de la page, par exemple modifier la couleur de fond de la page en beige. Attention, votre feuille de style n'est pas modifiée pour autant, il est nécessaire de recopier les changements dans le fichier CSS si cela correspond à votre attente.



Ajouter une nouvelle règle en cliquant sur le signe +

```
img {
  border: 1px solid black;
}
```

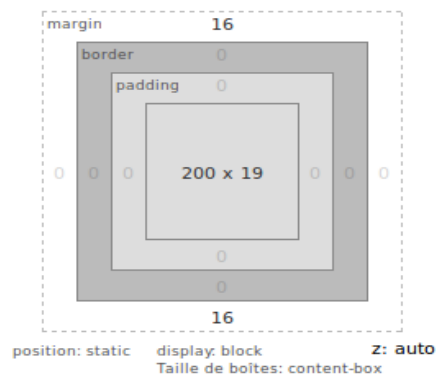
Maintenant, les images de drapeaux disposent d'une bordure. Attention, ceci ne sert que pour la mise au point de notre feuille de style. Il sera donc nécessaire de recopier les différentes règles dans le fichier **drapeauxStyle.css** par la suite.

Entre les accolades de la règle **img**, ajouter une nouvelle propriété **margin** avec comme valeur **10px** (pixels), voyez en direct le résultat, agissez sur les flèches haut et bas pour faire évoluer l'écartement entre chaque image.

Procédez de la même manière avec une nouvelle propriété, **padding**, puis ajoutez une couleur de fond, **white**, à vos images avec la propriété **background-color**.

Lorsque vos essais sont satisfaisants, placez-vous sur la règle `img` et copiez la déclaration de la règle dans le presse-papier à l'aide du bouton droit de la souris. Collez le résultat dans votre feuille de style, le résultat sera effectif chaque fois que vous relancerez votre page.

```
15  img {
16      border: 1px solid black;
17      margin: 10px;
18      padding: 10px;
19      background-color: white;
20  }
```



## Exercice N°5 – Positionnement des éléments – **class**

Toujours dans le même projet, mettre en commentaire les règles **body** et **img** de votre fichier **drapeauxStyle.css**

**Remarques :**      `/* début de commentaires CSS`  
                          `fin de commentaires CSS */`

Ajoutez les lignes suivantes qui serviront à positionner chaque drapeau :

```
24  img{
25      position: absolute;
26  }
27
28  .ligne1{
29      top: 100px;
30  }
31  .ligne2{
32      top: 200px;
33  }
```

```
34  .col1{
35      left: 100px;
36  }
37  .col2{
38      left: 200px;
39  }
40  .col3{
41      left: 300px;
42  }
43  .col4{
44      left: 400px;
45  }
46  .col5{
47      left: 500px;
48  }
```

Ajoutez une ligne et une colonne à image et visualisez le résultat.

```
<body>
  <h1>Les drapeaux</h1>
  
  
  
  
  
  
  
  
  
  
</body>
```

## Exercice N°6 – Positionnement des éléments – id

Maintenant, on souhaite afficher les drapeaux sous forme de rosace en jouant également sur la profondeur de champ, propriété **z-index**.

Cette fois ici, il est nécessaire d'appliquer des règles à chaque image.

Toujours dans le même projet, créez une nouvelle feuille de style nommé **drapeauxHover.css**.

Par calcul, les résultats suivants ont été obtenus dans le tableau suivant :

										
<b>left</b>	115	139	148	139	115	85	61	52	61	85
<b>top</b>	54	72	100	128	146	146	128	100	72	54
<b>z</b>	1	2	3	4	5	6	7	8	9	10

Pour chaque pays, il est nécessaire de créer un id comme le montre l'exemple ci-dessous :

dans **index.html**

```
<body>










</body>
```

dans **drapeauxHover.css**

```
#usa{
    position: absolute;
    left: 148px;
    top: 100px;
    z-index: 3;
}
```

Quelle règle pourrait contenir la propriété **position: absolute** pour que chaque drapeau en soit affecté ?

Ajoutez le code de la pseudo classe **hover** au sélecteur **img** dans votre fichier **drapeauxHover.css**.

```
img:hover{
    border: 2px red ridge;
}
```

Que constatez-vous en promenant votre souris sur les drapeaux ?

Modifiez la profondeur de champ des images avec la valeur 15. Le drapeau ciblé sera ainsi en avant lorsque l'on déplace le curseur de la souris dessus.

Pour que la nouvelle valeur de profondeur de champ soit prise en compte, il est nécessaire d'ajouter la règle CSS **!important**.

```
z-index: 15 !important;
```

Ajoutez un titre à chaque image comme le montre la ligne ci-dessous, visualisez le résultat.

```

```