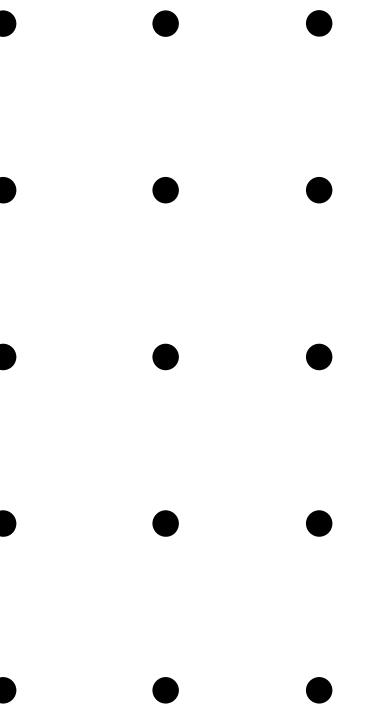
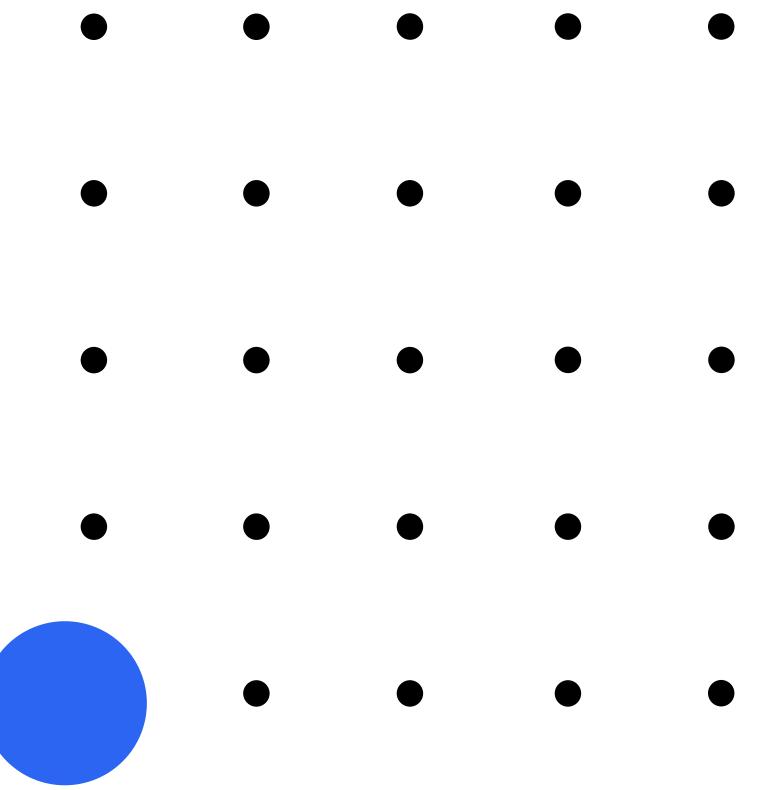
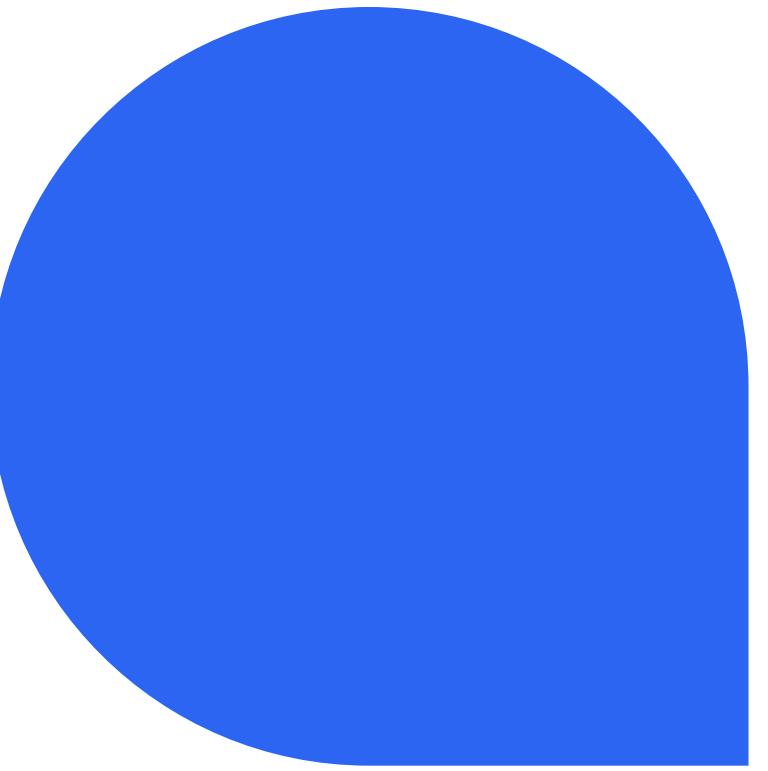


CSS

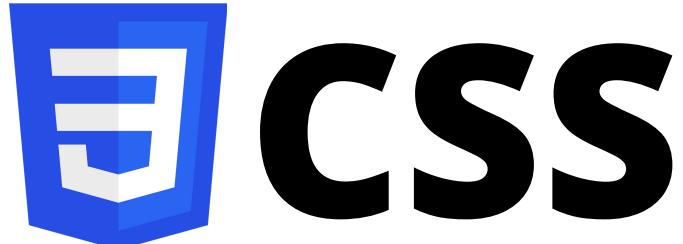




Cascading Style Sheet

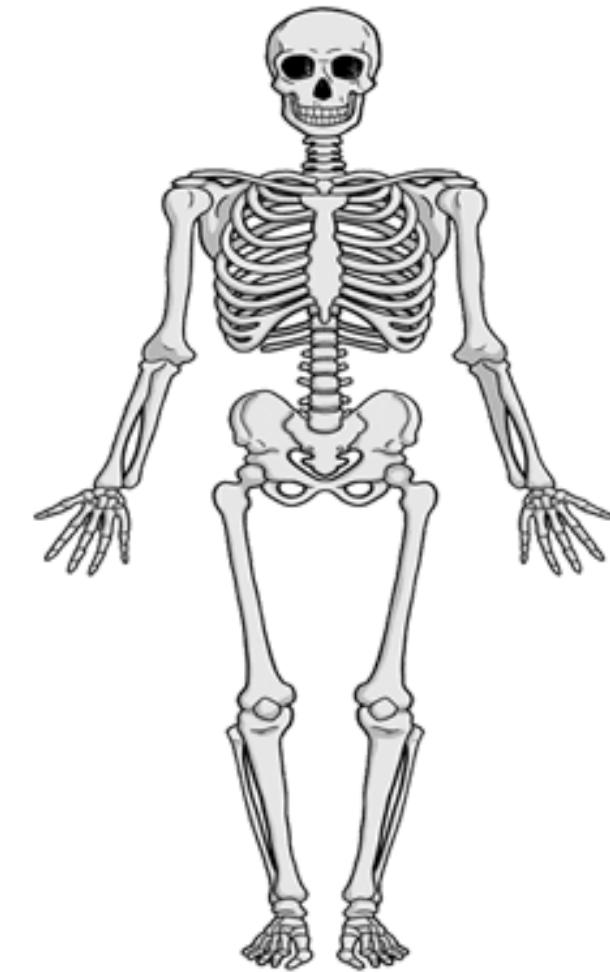
LEVEL UP

Les Feuilles de style en cascade sont utilisées dans la conception de sites web. Plus précisément pendant son "**habillage**".



Différence HTML / **CSS**

HTML



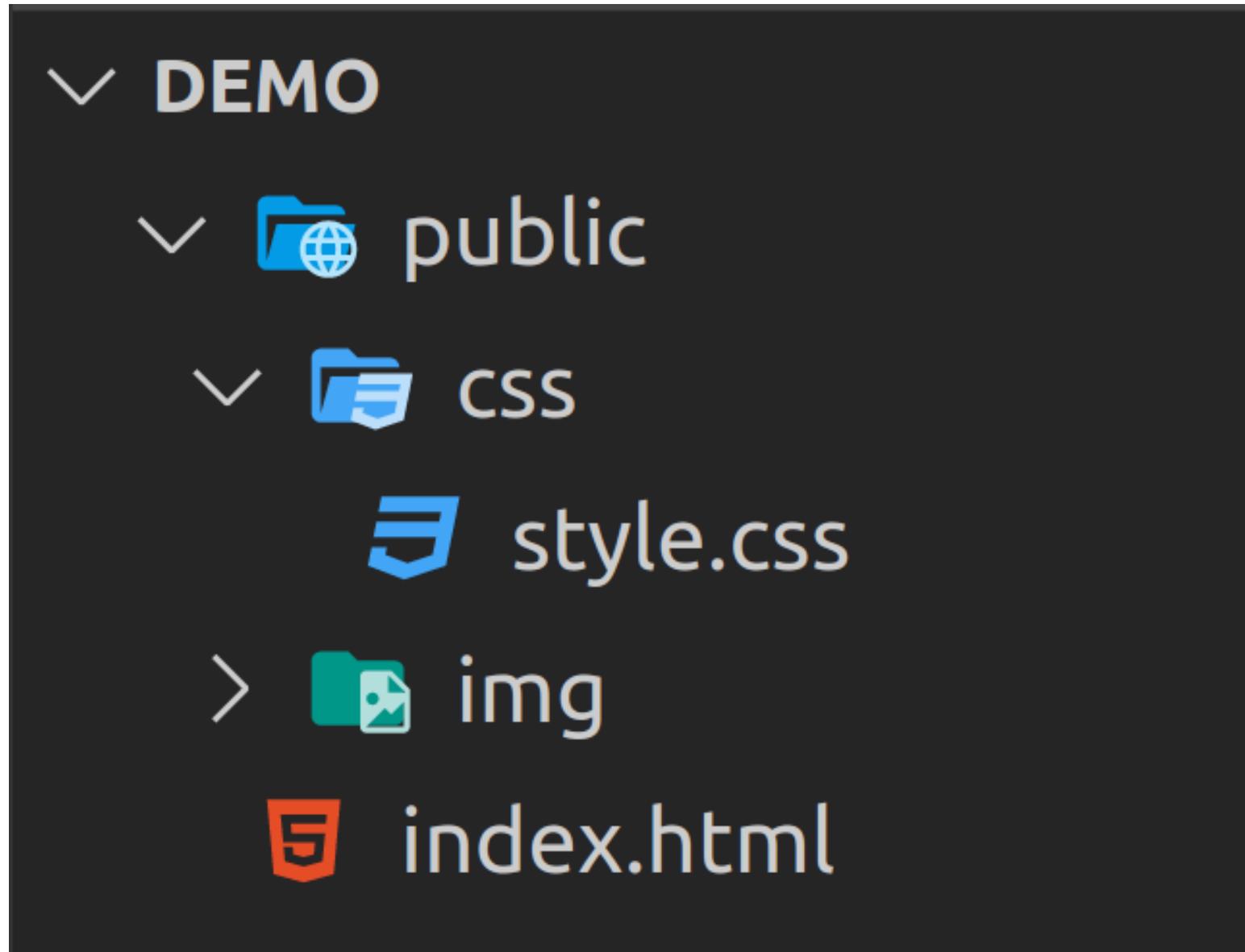
STRUCTURE

HTML + CSS



PRESENTATION

L'HTML représente le **squelette** de notre site et le CSS va lui appliquer un **style**.



Création fichier **CSS**

Dans le dossier public, on crée le dossier **css** dans lequel on fait un fichier **style.css**.



Lier CSS et HTML



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <link rel="stylesheet" href="./public/css/style.css">
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Il faut ajouter un [lien](#) dans le head de notre page HTML.
On va alors utiliser un [chemin relatif](#) pour référencer le fichier [style.css](#)



Syntaxe et sélecteurs



Syntaxe CSS

- Dans le fichier CSS on va d'abord utiliser un **selecteur** pour cibler ce qu'on va utiliser.
- Ensuite on doit **déclarer** le style que l'on souhaite appliquer.
- Pour ce faire, on utilise une **propriété** à laquelle on donne une **valeur**.

Sélecteur

```
p {  
    color: red;  
}
```

Propriété Valeur

Déclaration



ID, Class & Sélecteurs

Dans le CSS il existe un ordre de priorité parmi les sélecteurs :

1. **ID**
2. **Class**
3. **Balise**

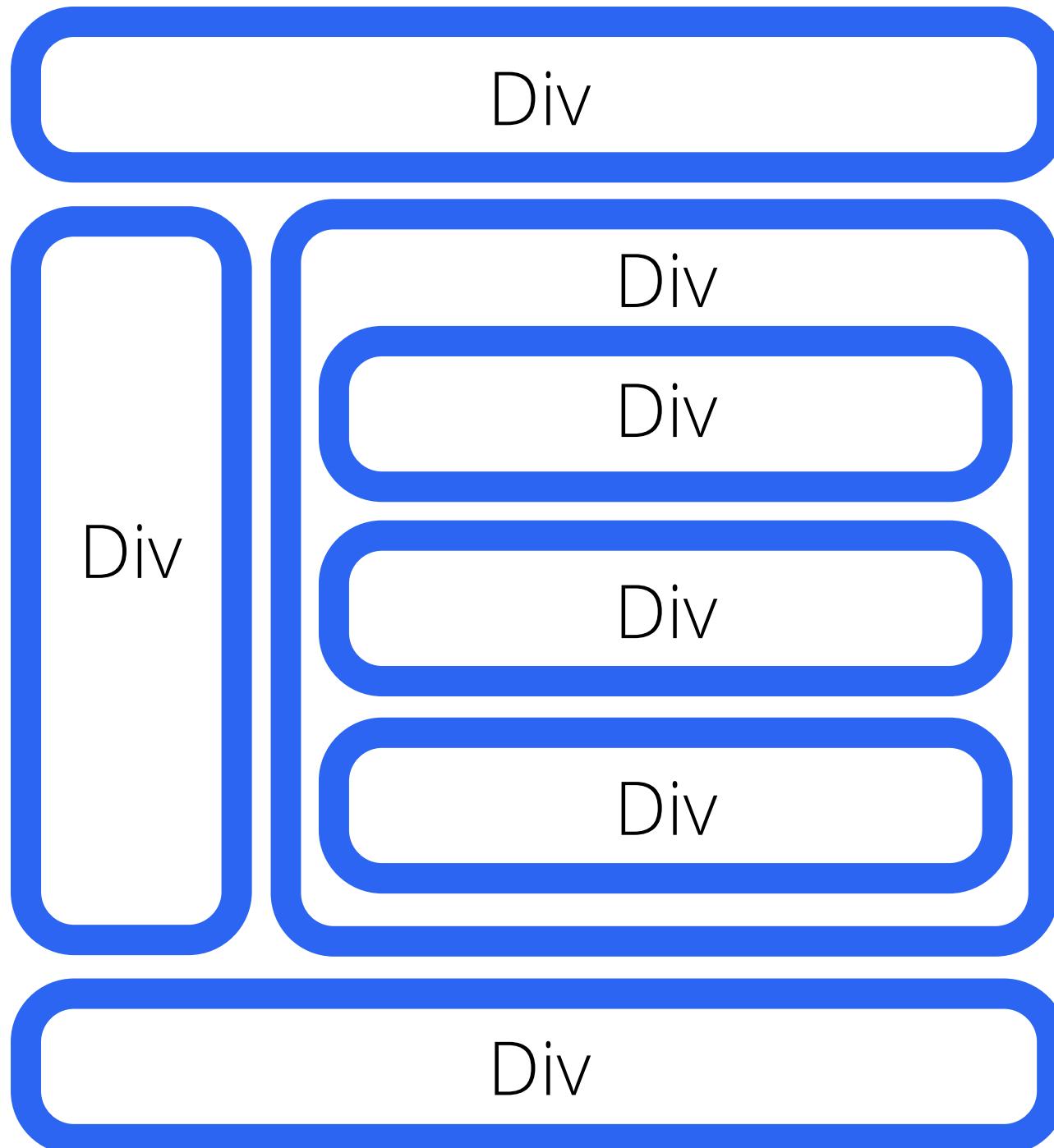
A screenshot of a code editor window titled 'style.css'. The file contains the following CSS code:

```
public > css > style.css > ...
1 #myId{
2   color: red;
3 }
4
5 .myClass{
6   color: red;
7 }
8
9 h1{
10  color: red;
11 }
12
```

The code uses syntax highlighting where '#myId' is yellow, '.myClass' is orange, and 'h1' is yellow. The 'color: red;' part is blue, and the semicolons are orange.



Boites & espacements



Comme déjà expliqué, la div est utilisée pour **regrouper** des éléments sous forme de **boîte**.

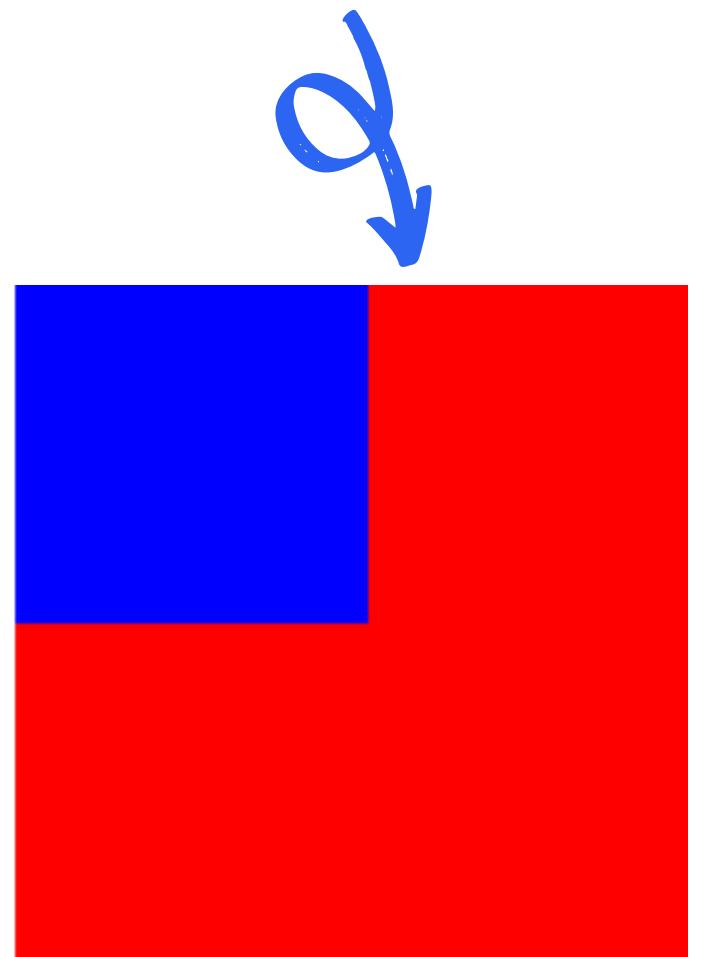
Ces boîtes peuvent **s'imbriquer** dans d'autres boîtes et ainsi former une **structure** comme sur ce schéma.



Tailles d'éléments

- Pour spécifier une hauteur et une largeur à nos éléments on va utiliser les propriétés : **width** et **height**.
- Pour cela, il est possible d'utiliser différentes **unités de mesures**. Dans cet exemple, on spécifie une taille en **pixel** (px) et en **pourcentage** (%).
- La taille en **pourcentage** va se calculer sur l'élément qui le contient. Dans cet exemple la div bleue à un **width** et une **height** de 50%

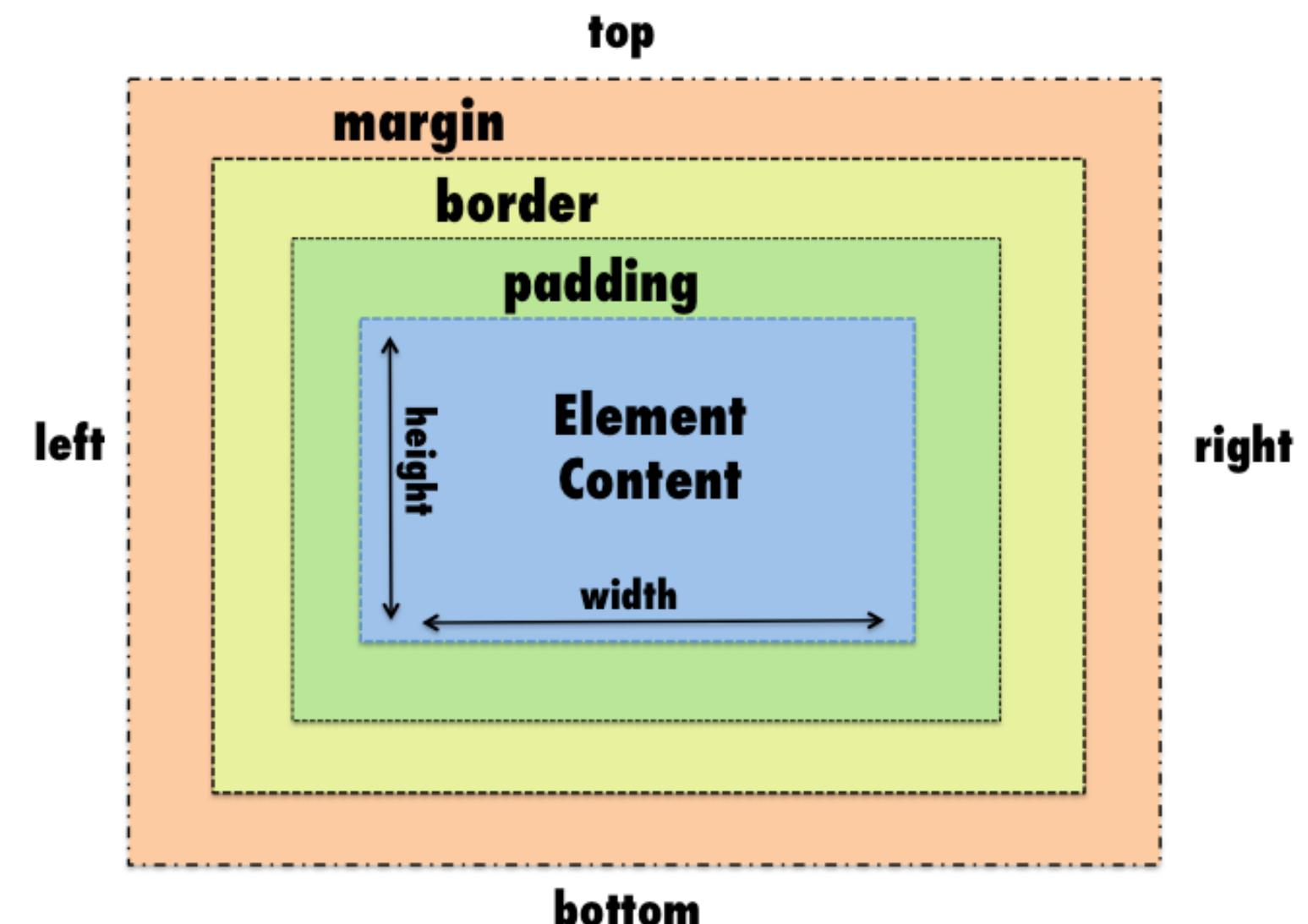
```
1 .myClass{  
2   width: 150px;  
3   height: 350px;  
4 }  
5  
6 #myId{  
7   width: 75%;  
8   height: 25%;  
9 }
```





Margin, Padding & Border

- Le **padding** qui se trouve à **l'intérieur** de la div
- Le **border** qui trouve en **bordure** de div
- Le **margin** qui se trouve à **l'extérieur** de la div





CSS

TOUCH ME!

Hover, focus...

- Il est également possible d'ajouter des **effets** et **animations** via le CSS.
- Pour ce faire, vous pouvez aller chercher de la **documentation** sur le hover, le focus...





Positionnement



Display

Il existe plusieurs types de **display**, nous allons nous concentrer sur les trois suivants :

- **Block**
- **Inline**
- **Inline-Block**

Display : Block

Display: block
Display: block
Display: block

Display : inline

Display: block
Display: block
Display: block
Display: block
Display: block

Display : inline-block

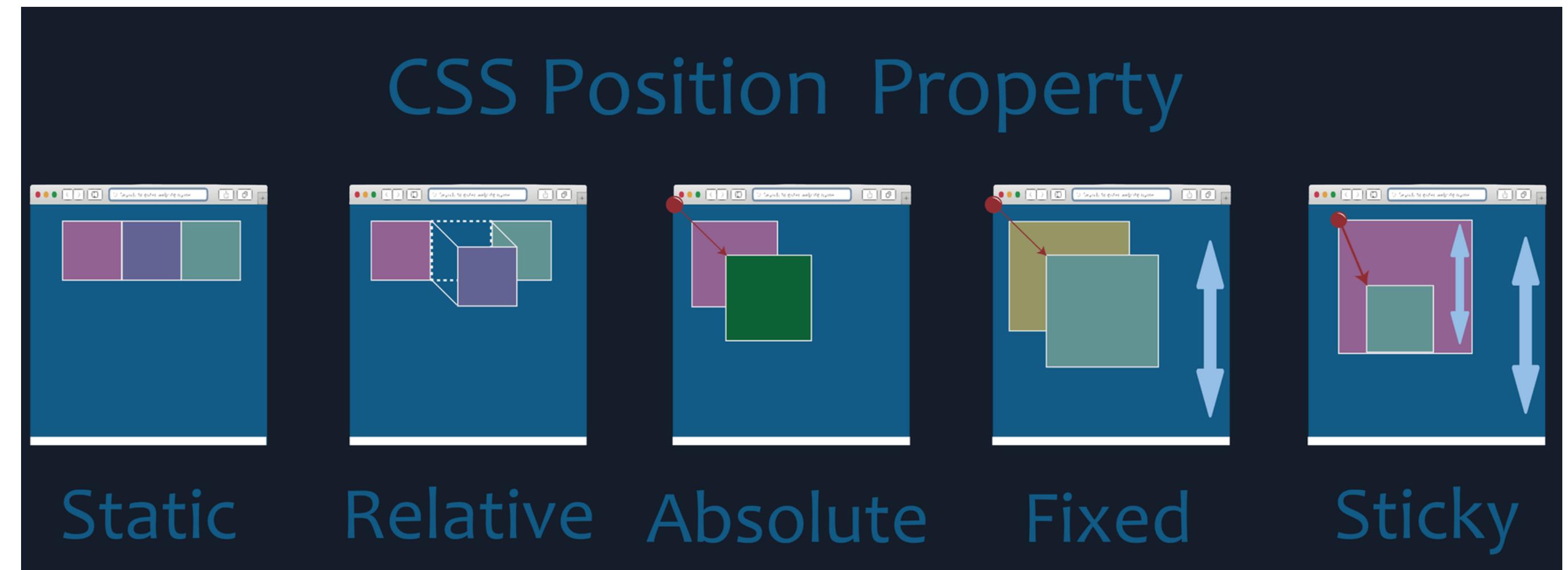
Display: inline-block
Display: inline-block
Display: inline-block
Display: inline-block



Position

Il est possible de donner des **positions** avec des **caractéristiques** différentes à des éléments :

1. **Static**
2. **Relative**
3. **Absolute**
4. **Fixe**
5. **Sticky**





Position

Une fois une position attribuée, on peut lui définir des propriétés : **top**, **bottom**, **right**, **left**.

En pourcentage (**%**) ou en pixels (**px**)

```
1 #relative{  
2   position: relative;  
3 }  
4  
5 #absolute{  
6   position: absolute;  
7 }  
8  
9 #fixed{  
10  position: fixed;  
11 }  
12  
13 #sticky{  
14  position: sticky;  
15 }
```

```
1 #absolute{  
2   position: absolute;  
3   top: 25%;  
4   left: 50%;  
5 }  
6  
7 #fixed{  
8   position: fixed;  
9   bottom: 100px;  
10  right: 420px;  
11 }
```



CSS

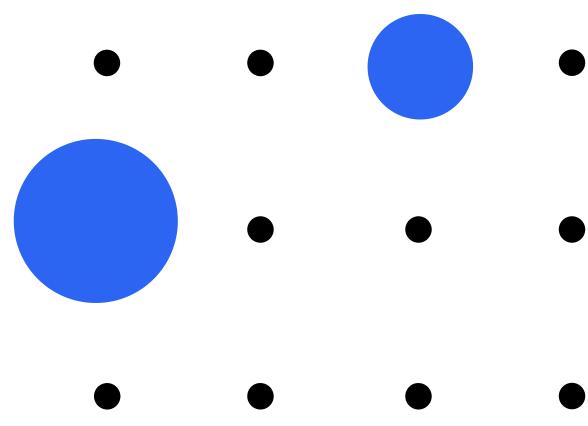


Background

Dans une div on peut styliser **l'arrière-plan** via différentes propriétés.

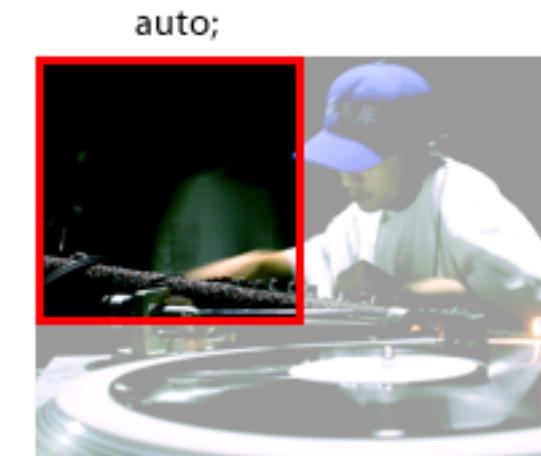
Par exemple, pour y mettre une image on utilise **background-image**.

Pour changer sa couleur on applique un **background-color**.



Background-size

En css, on peut choisir la **taille** d'un background de manière précise & choisir **comment il recouvre** (ou non) toute la surface d'une div.



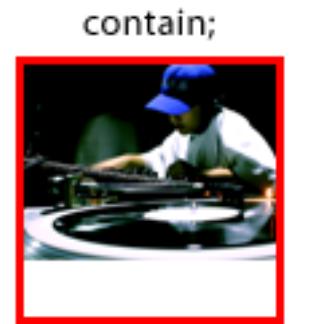
auto;



50% 100%;



cover;



contain;

background-size:



**WELL DONE
GUYS**

