

CSS

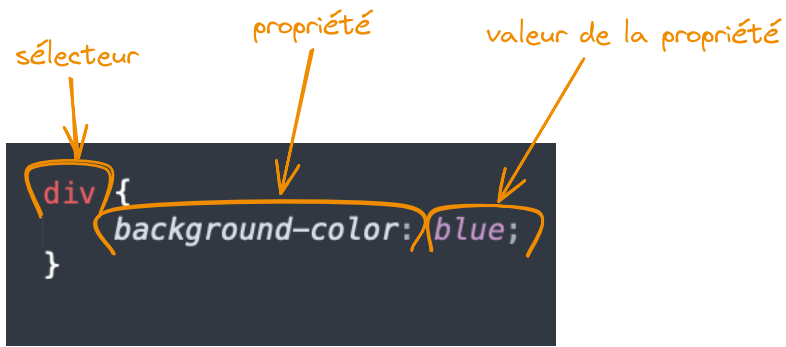
On a vu que le HTML servait à à définir la structure et le contenu du document.

Quid de la forme du document ?

CSS = CASCADING STYLE SHEETS

On utilise alors le CSS pour définir le style de la page !

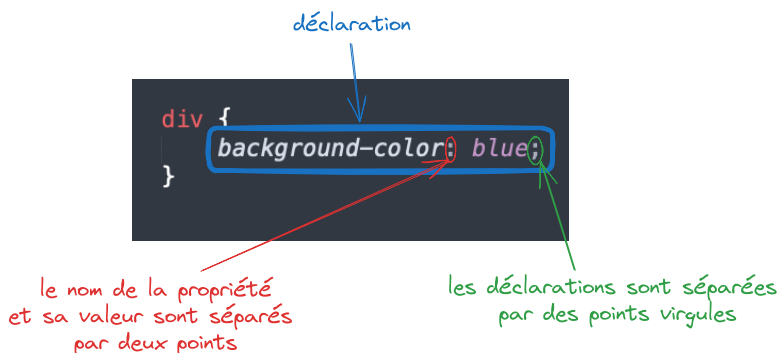
Syntaxe d'une règle en CSS



Sélecteur

C'est l'ensemble d'éléments à qui on veut appliquer un style.

S'il y a plusieurs sélecteurs on les sépare par des virgules.



On peut mettre plusieurs éléments dans le sélecteur pour appliquer le même style à plusieurs éléments. Par ex :

```
p, td, h1 {  
  color: blue;
```

```
}
```

Le CSS inclus directement dans le HTML

Le style en attribut de balise (ou "en ligne")

On utilise l'attribut `style` de la balise qu'on veut customiser.

La valeur qu'on donne à cet attribut est de la forme :

`key1: value1; key2: value2 etc`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Une page HTML avec du CSS</title>
  </head>
  <body>
    <h1 style="color: yellow;">Le titre 1 de ma page</h1>
    <p style="background-color: green;">Un paragraphe avec des
choses écrites dedans.</p>
  </body>
</html>
```

Une feuille de style dans le HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Une page HTML avec du CSS dans le HEAD</title>
    <style>
      h1 {
        color: yellow;
      }
      p {
        background-color: green;
      }
    </style>
  </head>
  <body>
    <h1>Le titre 1 de ma page</h1>
    <p>Un paragraphe avec des choses écrites dedans.</p>
```

```
    </body>
</html>
```

Appliquer une feuille de style dans un document HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Une page HTML avec du CSS</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>Le titre 1 de ma page</h1>
    <p>Un paragraphe avec des choses écrites dedans.</p>
  </body>
</html>
```

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

où `style.css` est un fichier qui existe dans le même dossier que `index.html`

Si le fichier css est dans un autre fichier, on met le chemin vers ce fichier (soit absolu, soit relatif à l'emplacement du fichier html).

Gérer plusieurs fichiers

Par exemple, disons que notre dossier de projet à la forme suivante :

```
└─ projet/
   └─ index.html
   └─ styles
      └─ style1.css
      └─ style2.css
```

alors pour inclure les deux fichiers CSS dans `index.html`, on écrira :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Une page HTML avec deux fichiers CSS</title>
    <link rel="stylesheet" href="style/style1.css" />
    <link rel="stylesheet" href="style/style2.css" />
```

```
</head>
<body>
    <h1>Le titre 1 de ma page</h1>
    <p>Un paragraphe avec des choses écrites dedans.</p>
</body>
</html>
```

Cascade

CSS = Cascading Style Sheets

Pourquoi cascade???

--> l'ordre des règles CSS a un impact

test :

```
h1 {
    color: yellow;
}

h1 {
    color: green;
}
```

À votre avis, de quelle couleur sera mon titre ?

Mais, alors, comment faire si, par exemple, j'ai plusieurs titres `h1` mais je veux qu'ils aient un style différent ?

La solution qui paraît simple est d'ajouter le style "en ligne" (cf plus haut) mais on a vu qu'il valait mieux ne pas procéder de cette façon.

C'est donc à ça que sert l'attribut `class` des balises en HTML.

Dans l'exemple suivant, le titre auquel je veux donner un style particulier a un nouvel attribut :

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>Une page HTML avec des attributs class</title>
        <link rel="stylesheet" href="style/style.css" />
    </head>
    <body>
        <h1 class="titre-important">Le titre le plus important de la
```

```
page !!</h1>
        <p>Un paragraphe avec des choses écrites dedans.</p>
        <h1>Un titre important mais moins que le premier.</h1>
    </body>
</html>
```

et dans le fichier CSS, on procèdera ainsi :

```
h1.titre-important {
    color: yellow;
}

h1 {
    color: green;
}
```

ou :

```
.titre-important {
    color: yellow;
}

h1 {
    color: green;
}
```

--> tous les h1 ont la couleur verte SAUF s'ils ont la class `titre-important` : pour ceux-là la première règle prend le dessus sur l'autre
on appelle ça la **spécificité**

Héritage

On essaiera de comprendre la notion d'héritage en TP en jouant avec les propriétés.

Sélecteurs

tous les sélecteurs : https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors

Par nom de balise

```
h1 {

}
```

Par rapport à l'attribut `class`

```
.nom-de-la-classe {  
  
}
```

Par rapport à l'attribut `id`

```
#valeur-id {  
  
}
```

Différence `id` vs `class`

Un id est unique, alors qu'une classe peut être trouvée sur plusieurs balises (pas forcément de même type, ie il peut y avoir un h1 et un h2 avec la même classe)

Sélecteurs par attribut

Pour sélectionner toutes les balises `<a>` dont l'attribut `title` est défini :

```
a[title] {  
  
}
```

Pour sélectionner toutes les balises `<h1>` dont l'attribut `class` est égal à la chaîne de caractères "titre" :

```
h1[class="titre"] {  
  
}
```

pour voir plus, cf https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Attribute_selectors#presence_and_value_selectors

pseudo-classes et pseudo-éléments

Soit la liste suivante :

```
<ul>  
  <li> un </li>  
  <li> deux </li>
```

```
<li> trois </li>
</ul>
```

Disons qu'on veut donner un style spécifique au premier élément de cette liste.

On a bien sûr l'option de préciser une classe en tant qu'attribut du premier élément.

Mais si on veut que notre code reste flexible, c'est-à-dire que si on ajoute un élément à la liste en tout premier, alors l'élément avec l'attribut classe ne sera plus le premier

CSS nous propose alors des **pseudo-classes** telles que `:first-child` qu'on utilise ainsi :

```
ul:first-child {
    color: red;
}
```

et maintenant, quelle que soit la première ligne de la liste, elle sera toujours en rouge !

Un autre exemple de pseudo-élément est `:hover`

```
a:hover{
    color: red;
}
```

qui change le style quand la souris survole l'élément `<a>` .