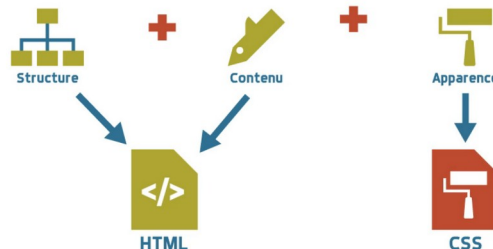


I) Historique, utilité, emplacement

Depuis 1990, les internautes peuvent partager et relier des documents structurés grâce au HTML. C'est très pratique, fonctionnel, mais le résultat est austère et peu esthétique. Il faudra attendre 1996 avec la sortie de CSS1 pour que les auteurs puissent ajouter du style à leur document HTML.

Les **feuilles de style en cascade**, généralement appelées **CSS** de l'anglais *Cascading Style Sheets* forment un langage informatique qui décrit la présentation des documents HTML.



Il y a 3 manières d'inclure du CSS dans un fichier HTML

- Sur les éléments. C'est une mauvaise pratique

```
<p style="color:red;">
ce paragraphe est rouge
</p>
```

- Dans l'entête du fichier HTML. C'est rarement acceptable.

```
<head>
  <style>
    p {color:red;}
  </style>
</head>
```

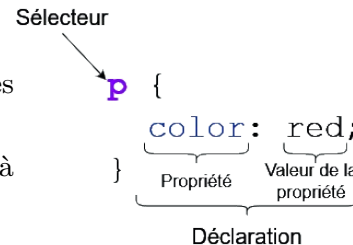
- La bonne pratique est d'inclure le css dans un fichier séparé en indiquant dans l'en-tête:

```
<link rel="stylesheet" href="chemin/vers/fichier.css">
```

II) Sélecteur

Pour indiquer à quels éléments on applique un style, on utilise des sélecteurs.

On considère pour l'instant les sélecteurs de type qui consistent à indiquer le type d'élément sélectionnés. Par exemple **strong**, **div**, **p**, ...



Anatomie d'une règle css

III) Propriétés

Pour chaque sélection d'éléments, on indique quelles propriétés de style on souhaite modifier. On fournit ci-dessous quelques propriétés élémentaires et le lecteur curieux trouvera des listes plus exhaustives dans la documentation en ligne.

Propriété	Description
color	Couleur du texte
font-size	Taille du texte
font-weight	Graisse du texte
font-family	Police(s) de caractères
text-align	Alignement du texte
background-color	Couleur d'arrière-plan
background-image	Image d'arrière-plan

Exemple 1: Les liens sont rouges et grands

```
a {
  color: red;
  font-size: large;
}
```

Exemple 2: Le texte des paragraphes est bleu sur fond jaune.

```
p {
  color: blue;
  background-color: yellow;
}
```

Remarque: Héritage

Certaines propriétés se transmettent aux enfants:
On parle d'héritage. Dans la liste ci-dessus, seules les propriétés d'arrière-plan ne s'héritent pas.

HTML	Règle CSS
<pre><div> <p>Ce texte sera rouge</p> </div></pre>	<pre>div { color: red; }</pre>

IV) Valeurs et unités

Longueurs

Certaines propriétés acceptent les valeurs de type **longueur** comme, par exemple, **font-size** .
De nombreuses unités peuvent être utilisées: **cm**, **mm**, **Q** (quart de mm), **in** (2,54 cm), **pt**, **px**, ...

Unités de longueur absolue

En CSS, le **px** CSS est la plus petite longueur qui peut être vu confortablement par l'œil humain. La longueur dépend de la résolution et de la distance de l'écran mais le confort de lecture reste stable.
Pour un écran à distance de bout de bras, **1px** \approx 0,265mm.

Les unités comme le **cm**, le **mm**, ou le pouce (**in**) sont ensuite définis par rapport au **px**. Par exemple **1cm** = 38**px** et **1in** = 96**px**. Ainsi, mettre une valeur CSS à 1cm ne veut pas dire que la taille des éléments fera 1cm à l'écran.

Remarque: Lors d'une impression les unités sont converties pour suivre le système métrique. Ainsi, lors d'une impression, 1cm fera bien 1cm.

Unités de longueur relative:

Les unités de longueur relative permettent d'exprimer des quantités relatives à quelque chose d'autre comme la taille de la police de l'élément parent ou la taille de la zone d'affichage.

On peut aussi utiliser l'unité **%** qui correspond à la taille de l'élément parent.

Unité	Relative à
em	Taille de police de l'élément parent
rem	Taille de police de l'élément racine
vh	1% de la hauteur de la zone d'affichage
vw	1% de la largeur de la zone d'affichage

Remarque: Le [W3C recommande](#) l'utilisation des unités **em**, **px** et **%** pour un affichage sur écran.

Exercice 1: A faire sur ce [lien codepen](#)

Consignes	HTML	CSS
1. Donner la taille de police des lettres A , B , C et D . 2. A l'aide d'unité relative (em ou rem), modifier le CSS pour que la taille d'affichage des lettres soient: A: 20px - B: 10px - C: 5px – D:15px	<pre><html> <body> A <div> B <p> C D </p> </div> </body> </html></pre>	<pre>html { font-size: 10px; } div { font-size: 2em; } p { font-size: 0.25em; } b { font-size: 3rem; }</pre>

Couleurs:

Les couleurs sur un écran d'ordinateur s'obtiennent par addition de trois couleurs primaires, le rouge, le vert et le bleu. Une couleur est donc codée comme une suite de trois nombres R, V, B (en anglais RGB) indiquant l'intensité de ces trois couleurs. Chaque nombre est codé sur un octet ce qui correspond à 256 intensités pour chaque couleur (de 0 à 255). Au total, cela fait 16 777 216 possibilités théoriques de couleurs différentes, c'est-à-dire plus que ne peut en discerner l'œil humain (environ 2 millions).

Pour coder une couleur en CSS, on peut utiliser la fonction `rgb()` qui accepte trois nombres en paramètres.

Exemples: `rgb(0,0,0)` représente le noir. `rgb(255,0,0)` représente le rouge.

Remarque: La fonction `rgb()` accepte un quatrième argument optionnel: l'opacité. C'est un nombre entre 0 et 1 ou un pourcentage. La valeur 1 ou 100% correspond à l'opacité maximum.

Cette suite de trois nombres RGB peut également être codée en hexadécimal précédé du symbole `#`. Chaque nombre entre 0 et 255 s'écrit en hexadécimal par un nombre entre 00 et ff.

Exemple: `#ff0000` représente la couleur rouge.

Nous avons déjà rencontré les valeurs de couleurs `red` et `blue`. CSS accepte une liste de 148 couleurs: Les 147 couleurs **X11** plus la couleur **rebeccapurple**.

Exercice 2: Donner la couleur et la taille de caractère d'affichage des lettres A, B, C, D et E.

Réponses		Fichier HTML	Fichier <code>index.css</code>
	Couleur	Taille	<pre> <!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Exemple de html</title> <link rel="stylesheet" href="index.css"> </head> <body> <h1> A </h1> <div> <p> B C D </p> E </div> </body> </html> </pre>
A			<pre> a { font-size:16px; } h1 { color: red; font-size: 20px; } strong { color: #00FFFF } div { color:blue; font-size:14px; } p { color:rgb(0, 255, 0); } </pre>
B			
C			
D			
E			

Exercice 3: voir dossier ci-joint.