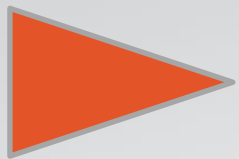


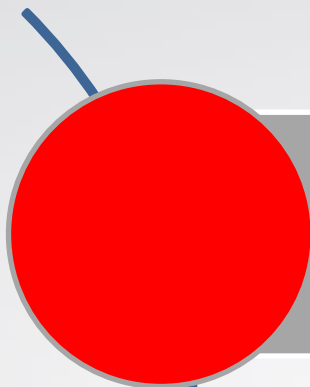
Cours:

CSS3

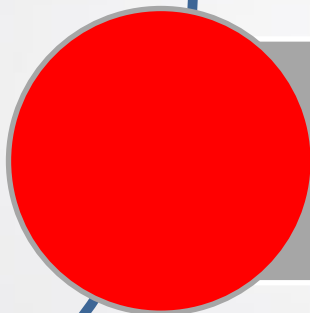
2019-2020



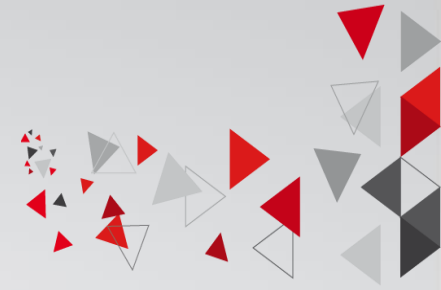
Plan



Les bases en CSS

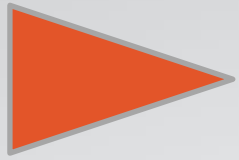


Atelier CSS



Partie 3:

Les bases en CSS

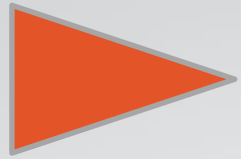


CSS : Définition et intérêt



CSS = Cascading StyleSheets

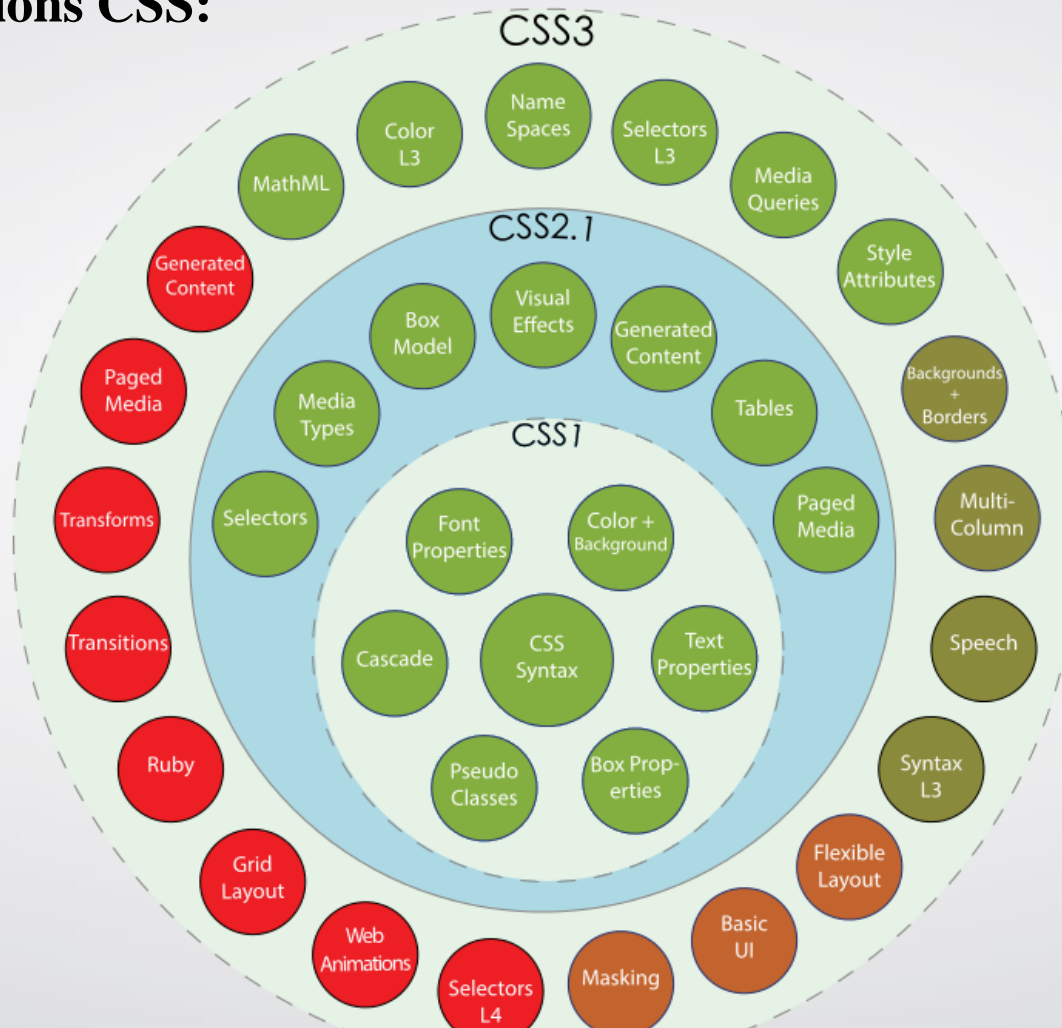
- Langage de style
- Sert à mettre en forme du contenu
- Permet de modifier la taille d'un texte, ajouter des bordures, une couleurs, etc.
- Versions de CSS:

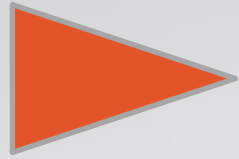


CSS : Définition et intérêt



Les versions CSS:

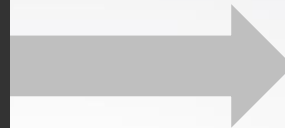




CSS : Exemple de code

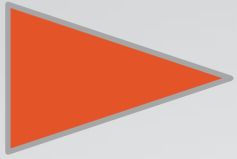


```
h1{  
  color: red;  
  font-size: 14px;  
}  
  
p{  
  color: blue;  
  text-decoration: underline;  
}
```



Ceci est un gros titre

Ceci est un paragraphe



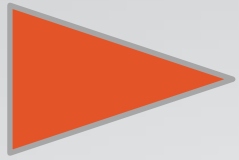
Où écrire le code CSS



- Ecrire le CSS dans la balise ouvrante des éléments HTML

```
<head>
  <title>Où écrire le CSS ?</title>
  <meta charset= "utf-8">

  <style>
    body{
      background-color: orange;
    }
    p{
      color: blue;
      font-size: 16px;
    }
  </style>
</head>
```



Où écrire le code CSS

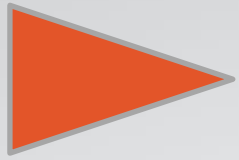


- Ecrire le CSS dans un élément HTML « style »

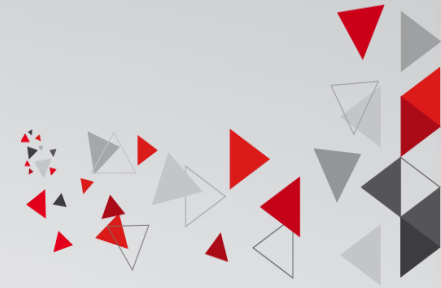
```
<body style="background-color:orange;">
  <h1>Un titre de niveau 1</h1>
  <p style="color: blue;font-size: 16px;">Un paragraphe</p>
  <p>Un deuxième paragraphe</p>
</body>
```

- Ecrire le CSS dans un fichier CSS séparé

```
<head>
  <title>Où écrire le CSS ?</title>
  <meta charset= "utf-8">
  <link rel="stylesheet" href="styles.css">
</head>
```

Les sélecteurs

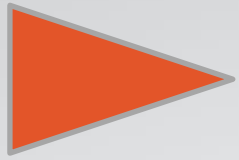


- Pour appliquer un style sur un élément HTML, il faut le **sélectionner**.
- Un **sélecteur** permet de cibler un ou plusieurs éléments HTML
- Il y a deux types de sélecteurs:
 - ✓ Sélecteurs simples.
 - ✓ Sélecteurs complexes.
- Syntaxe:

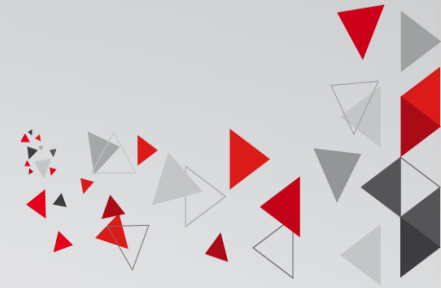
Sélecteur

Déclaration





Les propriétés



- Pour appliquer un style sur un élément HTML, il faut modifier ses caractéristiques
- Les propriétés permettent de choisir les styles à appliquer sur un élément HTML.
- Chaque propriété est accompagnée d'une ou plusieurs valeurs qui vont définir le comportement de cette propriété.

```
p{  
  color: blue;  
  font-size: 14px;  
}
```



```
selecteur{  
  propriété1: valeur1;  
  propriété2: valeur2;  
}
```

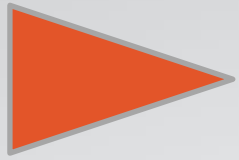


Sélecteurs et propriétés



Les sélecteurs les plus utilisés (communs)

Exemple	Classification	Définition
h1	Sélecteur de type	Sélectionne un élément par son type (nom)
.redLigne	Sélecteur de classe	Sélectionne un élément par la valeur d'attribut de classe , qui peut être réutilisée plusieurs fois par page
#intro	Sélecteur ID	Sélectionne un élément par la valeur d'attribut ID, qui est unique et ne doit être utilisée qu'une fois par page



Sélecteurs et propriétés



Quelle est la différence entre une classe et un id ?

- un *id* s'applique à un **objet unique** : il ne peut pas y avoir deux mêmes id dans une page
- une *classe* peut caractériser **plusieurs objets** (identiques ou non)

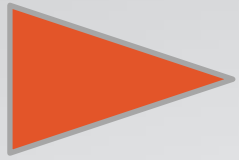
Par exemple, il est possible d'avoir ce code :

```
<div class="toto"></div>  
<div class="toto"></div>
```

Mais il n'est pas correct d'avoir ce code :

```
<div id="toto"></div>  
<div id="toto"></div>
```

=> l'id ne doit désigner qu'un seul objet du document.

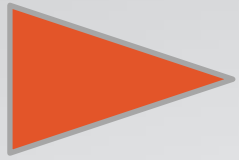


Les propriétés



Exemple de propriétés:

Exemple	Définition	Valeur(s) possible(s)
color	Définit une couleur	Red, yellow, Maroon, #800000, ...
Background-color	Définit la Couleur de l'arrière plan	
font-size	Définit la taille d'une police	15px, 25px, 150%,...
font-style	Définit le style de police d'un texte	Normal, italic, ...
width	Définit la largeur	10px, 15px,
hight	Définit la hauteur	
margin	Définit les marges haut et bas	Auto, 5px,



Modèles d'affichage en C.S.S (



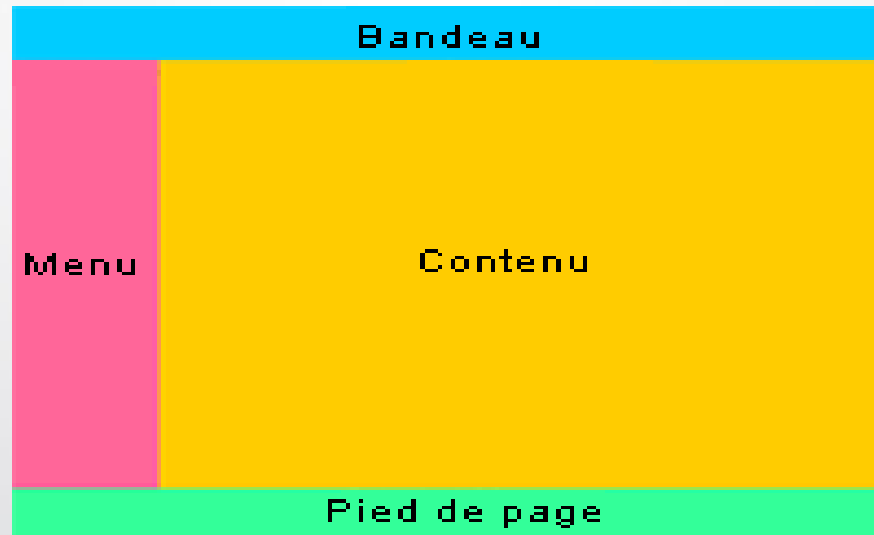
- Containers génériques.

`<div>` permet de structurer le document

`` permet de changer la mise en page de quelques mots dans un texte

`<div>` et `` s'utilisent toujours avec les classes et les ID.

Exemple 1





Modèles d'affichage en C.S.S



Exemple d'utilisation de DIV

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css"
href="styles.css">
<title>Positionnement CSS</title>
</head>
<body>
<div id="bandeau">
Ceci est le bandeau</div>
<div id="menu">
Ceci est le menu</div>
<div id="contenu">
Ceci est le contenu</div>
<div id="pied_page">Ceci est le pied de
page</div>
</body>
</html>
```

```
div {
text-align:center;
}
div#bandeau {
width:600px;
height:50px;

background-color:#00CCFF;
}
div#menu {
float:left;
width:100px;
height:400px;

background-color:#FF6699;
}
div#contenu {
float:left;
width:500px;
height:400px;

background-color:#FFCC00;
}
div#pied_page {
clear:both;
width:600px;
height:50px;
```



Modèles d'affichage en C.S.S (



➤ Les éléments « bloc » : une balise de type block sur votre page web crée automatiquement un retour à la ligne avant et après

- Les éléments blocs s'empilent les uns sur les autres
- Un élément bloc peut contenir d'autres éléments blocs
- Un élément bloc peut contenir des éléments en ligne

exemples : p, h1, h2, h3, h4, h5, h6, div

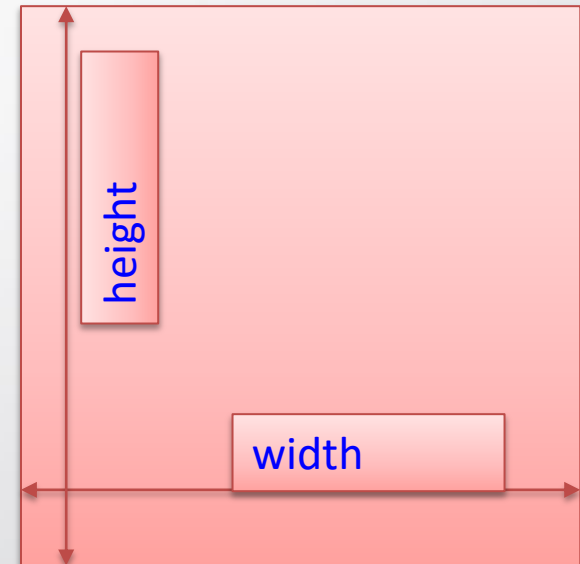
➤ Les éléments « en ligne » : une balise de type inline se trouve obligatoirement à l'intérieur d'une balise block. Une balise inline ne crée pas de retour à la ligne

- Les éléments en ligne se placent les uns à côté des autres
- Un élément en ligne peut contenir d'autres éléments en ligne

exemples : em, strong, b, i, a, span

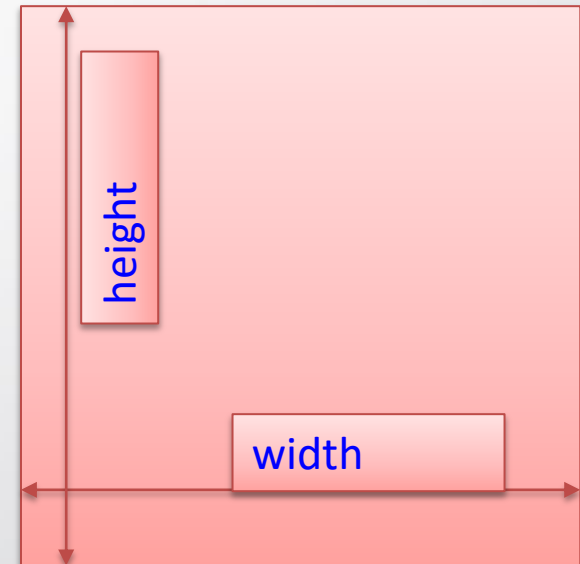
Modèles d'affichage en C.S.S (

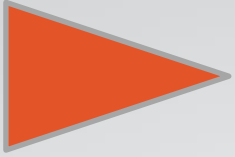
- *width*
 - Permet de spécifier la largeur d'un bloc
- *height*
 - Permet de spécifier la hauteur d'un bloc
- *position*
 - **Absolute** Le bloc est placé de façon absolue
 - **Relative**: Le bloc est placé relativement à la position qu'il aurait du occuper



Modèles d'affichage en C.S.S (

- *width*
 - Permet de spécifier la largeur d'un bloc
- *height*
 - Permet de spécifier la hauteur d'un bloc
- *position*
 - **Absolute** Le bloc est placé de façon absolue
 - **Relative**: Le bloc est placé relativement à la position qu'il aurait du occuper





Les Bordures



❑ Couleur de bordure : propriété **border-color**

➤ Possibilité d'attribuer différentes couleurs pour chaque bordure
(**border-top-color, border-bottom-color, border-right-color, border-left-color**)

➤ Exemples : `p{border-top-color: green;};` `img{bordercolor:#00FF00;}`

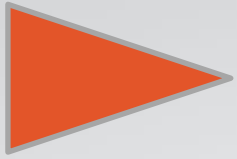
❑ Définition globale de bordure : propriété **border**

➤ Une écriture simplifiée des différentes propriétés de bordure

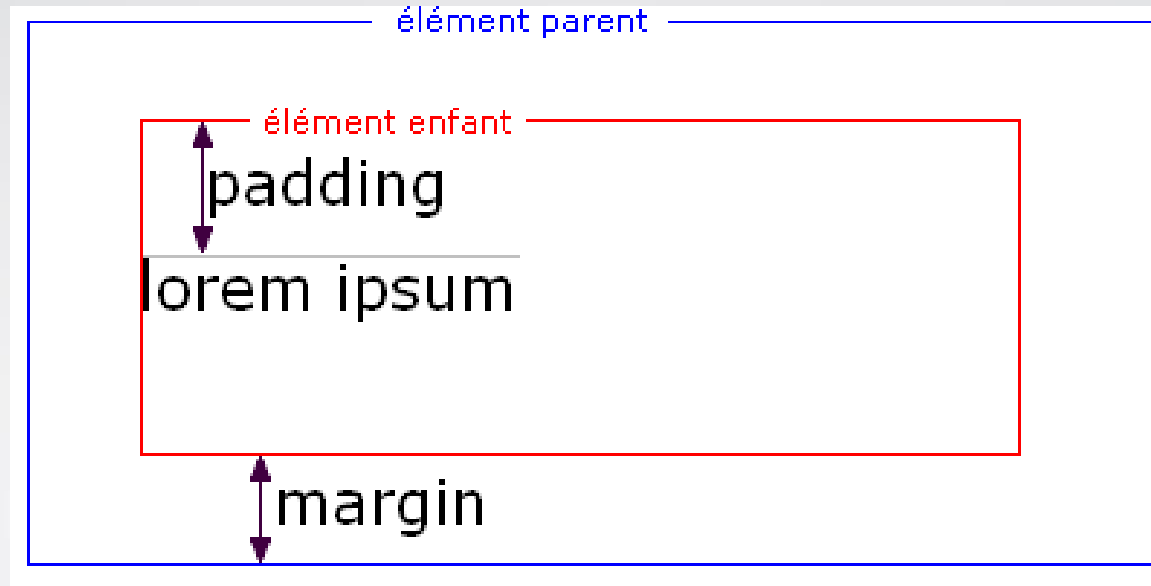
➤ Exemple : `h1{border: 5px double blue;}`

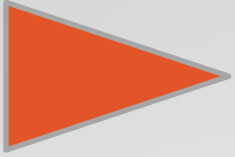
➤ Une écriture simplifiée pour chaque arête de bordure
(**border-top, border-bottom, border-right, border-left**)

➤ Exemple : `p{border-bottom: #00FFFF;}`



Les Marges - Padding





Les Marges



- Les marges peuvent être négatives ; dans ce cas la boîte d'un élément sort de celle de son parent
- Si la largeur de la marge est donnée en pourcentage, elle est calculée par rapport à celle du bloc parent.
- Possibilité de définir de une à quatre marges dans le sens des aiguilles d'une montre (haut, droit, bas, gauche).
- Pour réaliser un design centré quand on ne connaît pas la résolution du visiteur, on utilise **margin : auto**
- Exemples :

```
h1 { margin: 20px 10px 15px 10px; }  
p { margin: 10%; }  
b { margin-left: 3em; }  
.block_centre { margin: auto; width: 200px; }
```

Les Paddings



■ Permet comme la propriété `margin` d'aérer la présentation mais cette fois non pas entre deux éléments voisins mais directement autour du contenu.

■ Possibilité de définir dans l'ordre les espacements haut, droit, bas et gauche de l'élément.

■ Exemples:

```
h1 { padding: 20px 10px; }
```

```
h1 { padding-top: 20px;
```

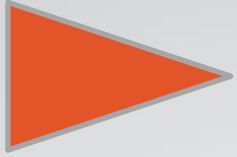
```
padding-bottom: 20px;
```

```
padding-right: 10px;
```

```
padding-left: 10px;
```



Atelier CSS



Références



- <https://css.developpez.com/tutoriels/utiliser-nouveaux-selecteur-css-3/>
- <http://pierre-giraud.com/html-css/cours-complet/selecteurs-proprietes-css.php>