

Sensibilité à la casse

- Nom des éléments n'est pas sensible

Ex. **P**{} et **p**{} sont identiques => pas d'erreur

- Le nom des **classes** et **identifiants** sont sensibles

Ex. **Classe1** et **classe1** sont différentes => erreur

Ordre de précedence

- ❑ Plus le sélecteur est spécifique, plus est la précedence

Element#id > id > classe > élément

- Ex. `<p id="p1" class="c1">PARAGRAPHE</p>`

Style1:

`p {color:"green";}`

`#p1 {color:"orange";}`

`.c1 {color:"red";}`

Résultat: la **seconde** règle sera appliquée ie le paragraphe sera donc affiché en couleur 'orange'

Style 2: `p {color:"green";}`
`#p1 {color:"orange";}`
`p#p1 {color:"red";}`
`.c1 {color:"white "};`

Résultat: la 3ème règle sera appliquée ie le paragraphe sera donc affiché en couleur ‘rouge’

❑ Un élément **hérite** le style de son parent si **aucun** style n'est spécifié

• Ex. `<p>` Texte en

`` Gras ``

`</p>`

Code CSS:

```
p { color: "blue "; }
```

=> La couleur bleue sera appliquée à la fois à 'texte en' et 'gras' si la règle `strong {color:'autre couleur';}` n'est pas définie

❑ Si des règles ont la **même** précedence, la **dernière** sera appliquée

- Ex.

```
p { font-size: 10px; }  
p { font-size: 12px; }
```

=> Le paragraphe sera affiché avec une police de 12 pixels. ie la dernière règle sera appliquée.

Notion de cascade

- Elle permet de définir l'ordre d'application des propriétés lors d'un conflit
- L'ordre est défini par priorité croissante:
 - Style par **défaut** du navigateur
 - Source **externe** fichier.css
 - Source **interne** avec un style défini dans la page
 - Source en **ligne** avec un style défini dans l'élément HTML (**haute priorité**)

Commentaire

- Le texte présent entre `/*` et `*/` sera ignoré
- Il ne sera pas affiché par le navigateur

Propriétés CSS

□ Texte et police

- **Couleur**: la propriété '**color**' spécifie la couleur du **texte**
- La couleur peut être spécifiée de **trois** manières:
 - Nom de la couleur (ex. **color**: 'blue')
 - RGB (ex. **color**: 'rgb(0,0, 255)')
 - Hexadécimale (ex. **color**: '#0000ff')

○ Alignement du texte:

● La propriété ‘text-align’ permet de:

- Centrer le texte `text-align:center;`
- Le positionner à gauche `text-align:left;`
- Le placer à droite `text-align:right;`
- Le justifier `text-align:justify;`

○ Décoration du texte

- La propriété '**text-decoration**' permet de:
 - Souligner le texte : **text-decoration:underline;**
 - Rayer ou surligner le texte: **text-decoration:overline;**
 - Laisser le texte normal **text-decoration:none;**

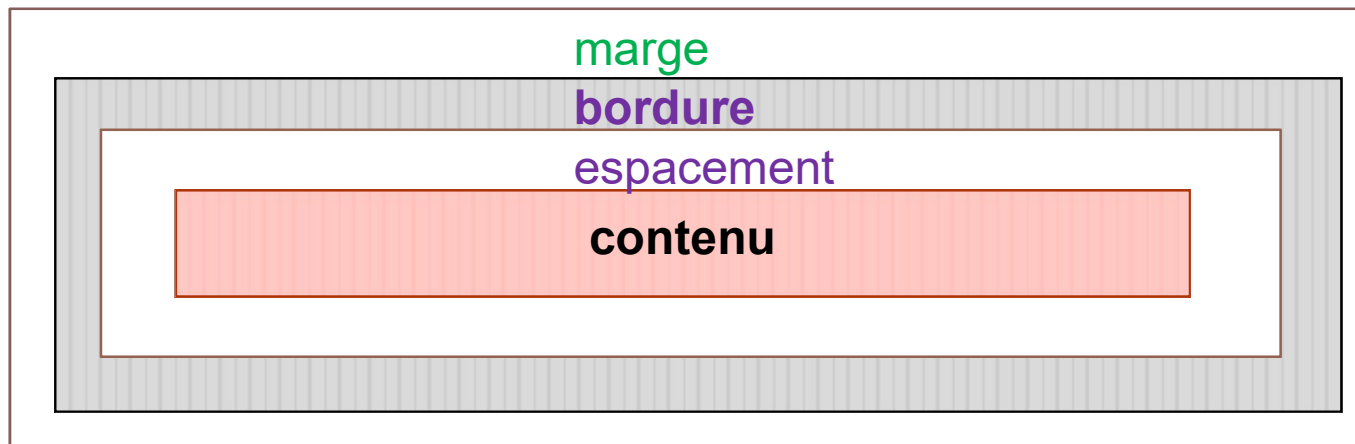
○ Espacement

- **Lettre**: La propriété '**letter-spacing**' permet de:
 - Diminuer l'espace entre les lettres d'un mot (valeur **négative**) **letter-spacing:-2px;**
 - Augmenter l'espace entre les lettres d'un mot (valeur **positive**) **letter-spacing:2px;**
- **Mot**: La propriété '**word-spacing**' permet de:
 - Diminuer l'espace entre les mots (valeur **négative**) **word-spacing:-2px;**
 - Augmenter l'espace entre les mots (valeur **positive**) **word-spacing:2px;**

- **Ligne**: La propriété '**line-height**' permet de spécifier l'espace entre les lignes d'un texte. Elle est exprimée en **pixel**, **pourcentage**, **em**, **cm** et **pt** (point)
- **Ex.** **line-height:20px;**
line-height:20%;

Modèle en boîte

- En CSS, un élément est traité comme une **boîte**
- A chaque élément HTML correspond une zone rectangulaire
- Chaque boîte ou élément possède des **marges** (margin), **bordures** (border), **espacements** (padding) et un **contenu** (content)



- À l'intérieur de la bordure, on trouve les espacements (padding) et le contenu (content)
- À l'extérieur de la bordure, on trouve les marges (margin)

Propriétés

❑ Contenu

- **Width** et **height**: dimension de la zone de **contenu** (largeur et hauteur). Les valeurs sont exprimées en **pixels** (px) ou en **pourcentage** (%)
- Elles peuvent aussi prendre la valeur **auto**
- **Overflow**: parfois la zone de contenu est trop petite pour contenir la totalité du texte. Par conséquent, le texte va déborder. Pour y remédier, on utilise la propriété **overflow**
- Cette dernière peut prendre les valeurs: **visible**, **hidden**, **scroll**, **auto**

❏ Padding et margin

- **Padding**: espace vide entre le contenu et la bordure
- **Margin**: espace entre la bordure et les éléments adjacents
- Ils sont **transparents** (on ne peut pas changer leur couleurs)
- Il existe **04** syntaxes différentes:

- Ex. les 03 premières notations sont équivalentes
 - `Margin-top:10px;`
`Margin-right:20px;`
`Margin-botom:10px;`
`Margin-left:20px;`
 - `Margin: 10px 20px 10px 20px;`
 - `Margin: 10px 20px;`
 - `Margin:10px;` Toutes les marges sont identiques

❑ Bordure

- La bordure possède les caractéristiques suivantes:
largeur, style et couleur.

➤ Largeur

`border-width: thin | medium | thick | nombre positif`

➤ Style

`border-style: none | hidden | dotted | dashed | solid |
double | groove | ridge | inset | outset`

- Cette règle s'applique au 4 bordures:

- `border-top-style:none`
- `border-right-style: solid`
- `border-bottom-style:none`
- `border-left-style:solid`

➤ Couleur

`border-color:rgb(255,0,0) | red | #FF0000 | transparent`

- **Définition globale:** `border:largeur | style | couleur`

Ex. `border:3px solid red;`

- **Bordure arrondie:** `border-radius:20px;`

□ Marges

- Définir une marge autour de chaque élément pour aérer le contenu de la page
- **Margin:valeur**: Cette valeur s'applique à **toute** les marges (haute, droite, basse et gauche) dans le sens des aiguilles d'une montre
- Il est possible de définir les marges individuellement
- **Ex.**
 - **margin-top**: 3px; /* marge **haute**
 - **margin-right**: 3px; /* marge **droite**
 - **margin-bottom**: 2px; /* marge **basse**
 - **margin-left**: 2px; /* marge **gauche**

Validation

- Service de validation du code CSS en ligne:
jigsaw.w3.org/css-validator
- La vérification du code peut se faire, soit par:
 - Adresse URI
 - Chargement du fichier CSS
 - Saisie directe du code CSS