

The logo of the University of Bordeaux is displayed against a background with a blue diagonal stripe in the top left and a dark grey diagonal stripe in the bottom right. The text 'université' is in a dark brown sans-serif font, with the 'u' and 'e' featuring blue highlights. Below it, 'de' is in a smaller dark brown font, and 'BORDEAUX' is in a larger, bold, dark brown sans-serif font.

université  
de **BORDEAUX**

# Bloc 1

## Cascading Style Sheets (CSS)

# Cascading Style Sheets (CSS – v3)

## → Séparation entre la **structure** et la **présentation**

- › structure : contenu HTML (mais aussi XML et SVG)
- › présentation : règles de style

## → **Avantages**

- › contenu et mise en forme développés indépendamment
- › un style pour plusieurs pages HTML
- › plusieurs styles – au choix – pour une page HTML
- › un style pour chaque type d'affichage / média (écran, impression papier, etc.)

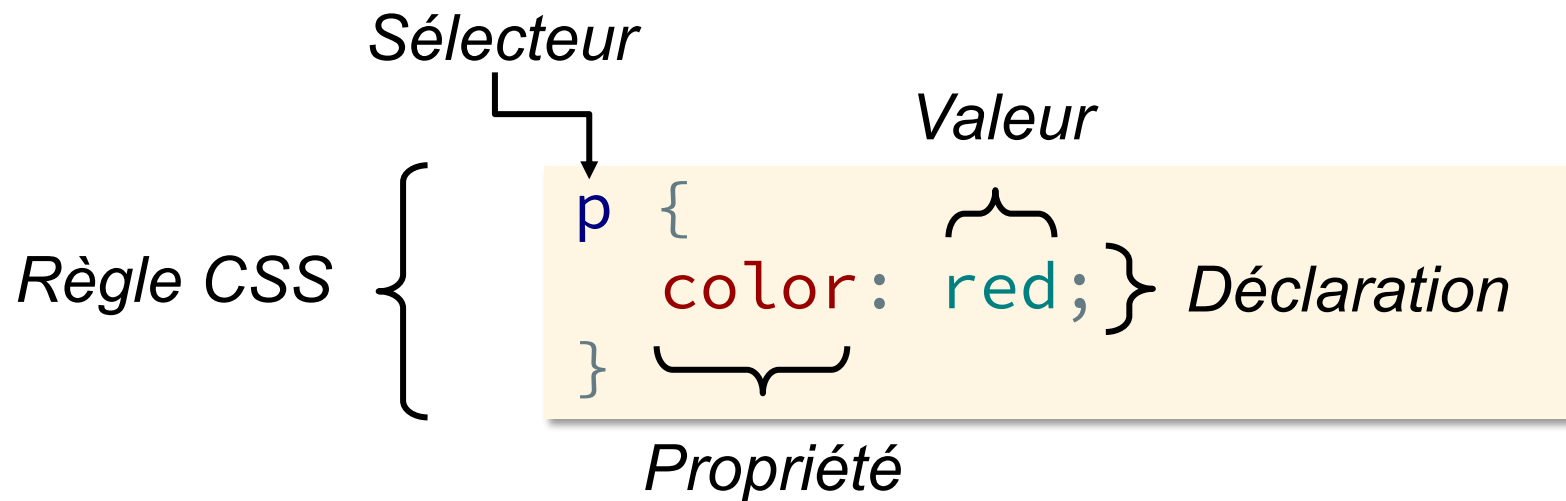
## → Changer de style suivant ses humeurs

- › <http://www.csszengarden.com/>

# Feuille de styles

## → Objectifs

- › Définir des règles de style (couleur, police, marge ...)
- › Sélectionner les éléments du DOM sur lesquels portent les règles
- › Règles appliquées par le navigateur lors de l'affichage de la page



# Où définir les règles (1/3)

- Dans un fichier **css** séparé (**recommandé**)
  - › Lien vers le fichier de style définit par la balise `<link>`
  - › Possibilité de lier plusieurs feuilles de styles à un même fichier HTML

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

style.css

```
<!DOCTYPE html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8" />  
    <link rel="stylesheet" href="style.css"  
  />  
  </head>  
  <body>  
    <h1>On aime le CSS</h1>  
    <p>Un paragraphe en rouge</p>  
  </body>  
</html>
```

example.html

# Où définir les règles (2/3)

→ Dans l'entête du document HTML

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <style type="text/css">
      p {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>On aime le CSS</h1>
    <p>Un paragraphe en rouge</p>
  </body>
</html>
```

# Où définir les règles (3/3)

- Dans l'attribut « *style* » des éléments HTML
  - > **à éviter autant que possible**

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>On aime le CSS</h1>
    <p style="color:red">Un paragraphe en rouge</p>
  </body>
</html>
```

# Propriétés graphiques de base (1/2)

→ Couleurs, mise en forme du texte, etc.

```
body {  
    background-color: rgb(255,0,255);  
    margin: 10px;  
}  
h1 {  
    font-style: italic;  
    text-align: center;  
    text-decoration: underline;  
}  
p {  
    color: red;  
    text-indent: 12px;  
}
```



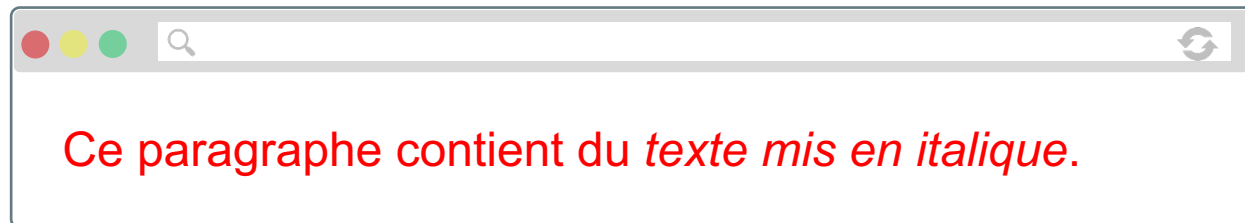
# Propriétés graphiques de base (2/2)

→ Propriétés **héritées** par les éléments imbriqués

```
p      { color: red; }  
span { font-style: italic; }
```



```
<p>Ce paragraphe contient du <span>texte mis  
en italique</span>.</p>
```



# Sélecteurs d'éléments

→ Le classique : **tous** les paragraphes `<p>`

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

→ Les paragraphes `<p>` à l'intérieur d'une balise `<div>`

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

→ Les `<h1>` qui suivent un `<p>`

```
p + h1 { color: red; }
```

→ Les `<h1>` et les `<h2>`

```
h1, h2 { color: red; }
```

# Sélecteur de classe

→ On peut ajouter une **classe** à toute balise

```
<h1 class="beau">Joli titre</h1>  
<p class="beau">Ce paragraphe est  
beau.</p>  
<h2>Moins beau!</h2>
```

→ Style appliqué à **toutes** les balises de la **classe**

```
.beau { color: red; }
```



Un élément peut avoir **plusieurs** classes

# Pseudo classes

- Classe automatique attribuée à un élément lorsqu'un **événement** se produit
- › Clic ou survol d'un lien avec la souris
  - › Lien déjà visité
  - › Premier caractère d'un paragraphe
  - › Etc.

```
a:link { color: black; }  
a:visited { color: gray; }  
h1:hover { font-weight: bold; }  
p:first-letter { font-size: 200% }  
ul:first-child { color: green; }
```

# Sélecteur d'identificateur

- On peut ajouter un **identificateur unique** à toute balise

```
<div id="entete">Joli titre</div>
```

- Style appliqué à l'**unique élément** ayant cet identificateur

```
#entete { color: red; }
```



Plus aucune balise ne peut utiliser ce style !

# Cascade et priorités

→ Même propriété définie par **plusieurs règles**

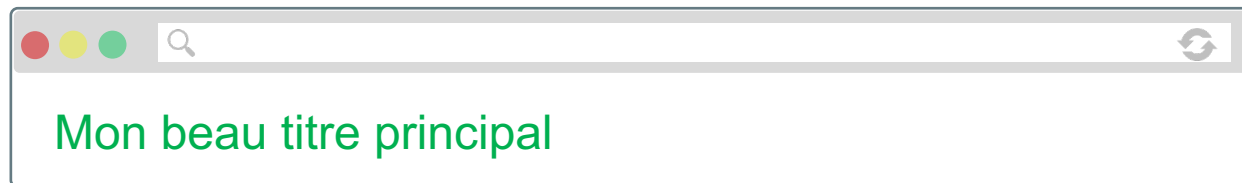
```
h1 { color: black; }  
#principal { color: green; }  
.beau { color: blue; }
```



```
<h1 id="principal" class="beau"> Mon beau  
titre principal </h1>
```

→ Règles de priorités (assez complexes)

→ À priorités égales, ordre linéaire dans le CSS

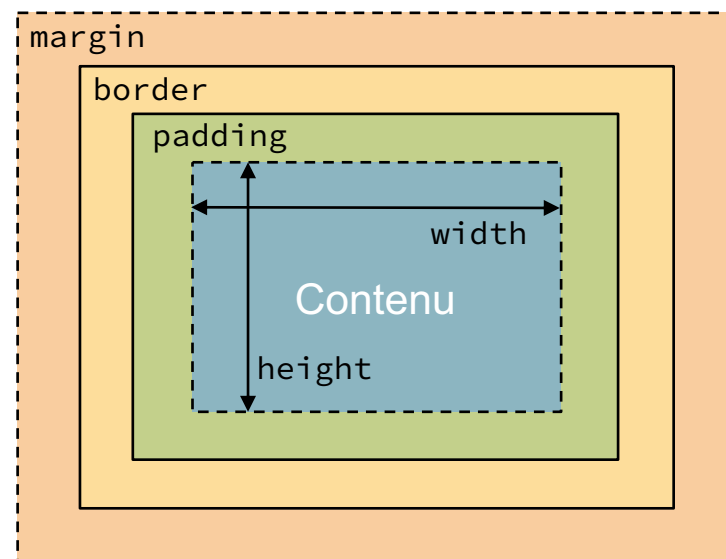


# Pagination : Modèle de boîtes

## Un élément HTML = une boîte

- › dimensions (width/height)
- › marges intérieures (padding)
- › marges extérieures (margin)
- › encadrement (border)

→ boîtes affichées au kilomètres



## Éléments **imbriqués** = boîtes imbriquées

- › contrôle du positionnement (position) et des débordements (overflow)
- › rapidement complexe

# Boîte *block*

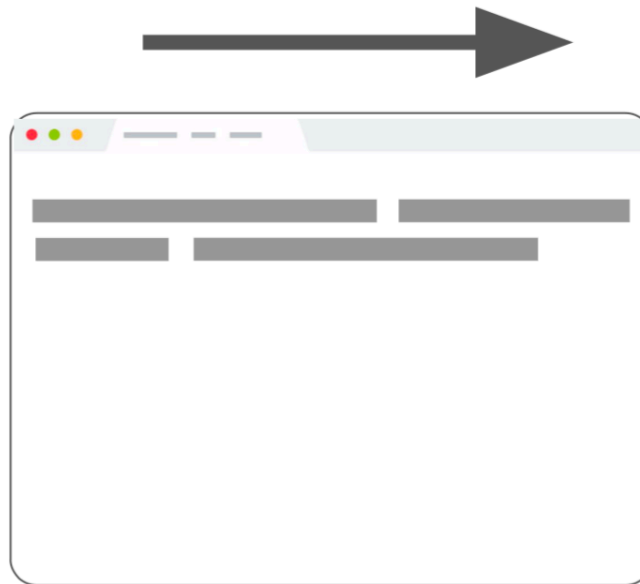
- Éléments : `<p>`, `<h1>`, `<div>`, `<ul>`, `<ol>` ...
- › Prend toute la largeur de la page
    - Flux du haut vers le bas
  - › Possède une hauteur et largeur
  - › Peut avoir des éléments fils de type *block* ou *inline*





# Boîte *inline*

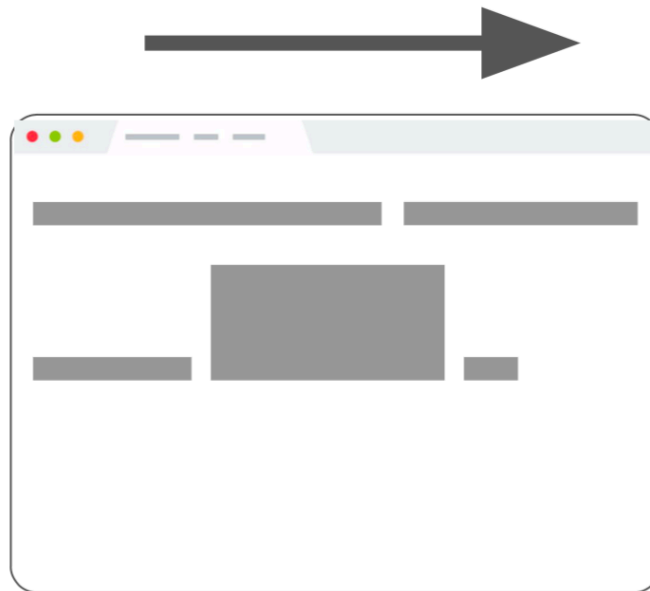
- Éléments : `<a>`, `<em>`, `<strong>`, `<span>`...
- › Prend juste la largeur nécessaire
    - Flux de la gauche vers la droite
  - › **Ne peut pas** avoir de hauteur et de largeur
  - › **Ne peut pas** avoir d'élément fils de type *block*
  - › **Ne peut pas** être positionnée (*float*, *position*...)



# Boîte inline-block

## → Élément : `<img>`

- › Prend juste la place nécessaire
- › **Peut** avoir de hauteur et de largeur
- › **Peut** avoir d'élément fils de type *block*
- › **Peut** être positionnée (*float, position ...*)



# Unités de mesure

→ Taille d'une police, largeur d'une bordure, hauteur d'un élément, etc.

→ Unités absolues

- › **in** pouce anglais : 25.4mm
- › **cm** centimètre
- › **mm** millimètre
- › **pt** point : 1/72 de pouce
- › **pc** pica : 12 points : 1/6 de pouce

→ Unités relatives

- › **ex** hauteur du x dans la police utilisée
- › **px** un pixel
- › **em** taille de la police utilisée

# Outil des développeurs

## Onglet « Styles »

The screenshot shows the Chrome DevTools 'Styles' panel. The 'body' element is selected, and its styles are listed on the left. The 'user agent stylesheet' is active, showing default styles for 'display' and 'margin'. On the right, a visual box model diagram illustrates the element's dimensions and spacing: a blue box for the content (1799 x 252.719px), a green box for padding, a yellow box for border, and an orange box for margin (8px). Below the diagram, a list of styles is shown with expandable arrows.

Filter	:hov .cls +
element.style {	
}	
body {	styles.css:1
background-color: black;	
}	
body {	user agent stylesheet
display: block;	
margin: 8px;	
}	

margin 8

border -

padding -

1799 x 252.719

8 - - 8

8

Filter ☐ Show all

- ▶ background-color: black
- ▶ display: block
- height: 252.719px
- ▶ margin-bottom: 8px
- ▶ margin-left: 8px
- ▶ margin-right: 8px
- ▶ margin-top: 8px
- width: 1799px

→ Édition interactive du CSS

- **Feuille de style** pour l'HTLM et le XML  
(~~pas un langage de programmation~~)
- Ensemble de **règles** décrivant la **présentation**  
d'un document structuré
- Utilisé par les **navigateurs** pour mettre en forme  
les pages web
- Plus de 300 propriétés CSS pour des **effets avancées**  
(pagination adaptée à l'écran, menus, transitions animées, etc.)