CSS

Michel Gagnon
École Polytechnique de Montréal
Eliana de Mattos Pinto Coelho
Université de Montréal

Introduction

- CSS (Cascading Style Sheets)
- Par un ensemble de règles appliquées, permet de spécifier l'apparence de chaque élément du document HTML
- Cascade:
  - On a plusieurs niveaux de styles appliqués séquentiellement, dans un ordre précis
  - Pour chaque niveau, un ensemble de règles est appliqué, dans leur ordre d'apparition

Format d'une règle CSS

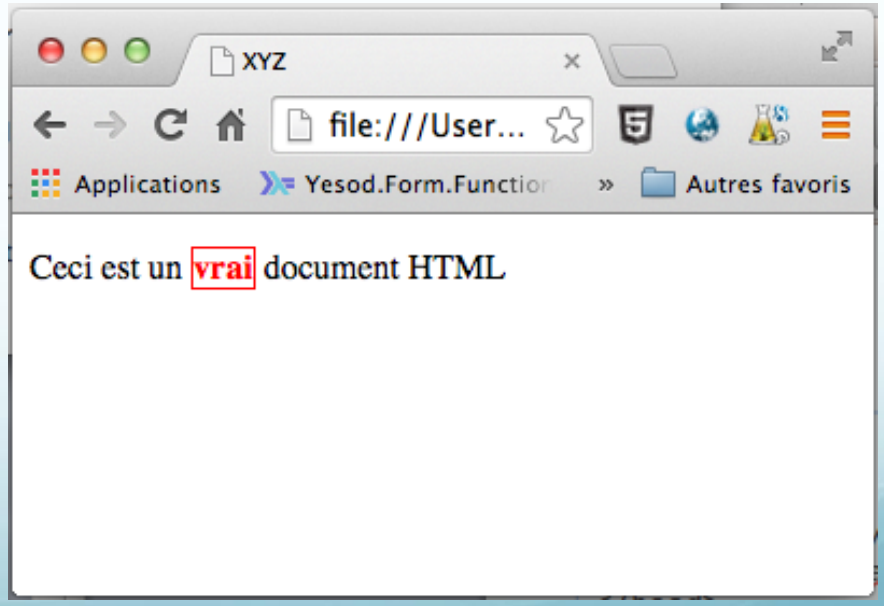
```
sélecteur, …, sélecteur {
      attribut: valeur ;
      …
      attribut: valeur ;
}
```

- Le sélecteur permet d'identifier le (ou les) élément(s) concerné(s)
- Pour chaque attribut spécifié, on indique sa valeur
- Important:
  - Comme nous le verrons plus loin dans le principe de cascade, si deux valeurs sont possibles pour un même attribut, c'est la dernière déclaration qui sera prise en compte
  - Si un attribut n'est pas reconnu par le navigateur, il est tout simplement ignoré

Exemple simple

```
<!doctype html>
<html>
  <head>
    <title>XYZ</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p> Ceci est un
        <strong>vrai</strong>
        document  HTML </p>
  </body>
</html>
```

```
strong {
    color: red;
    font-weight: bold;
    border: solid 1px;
}
```



Sélecteurs simples

- Sélecteur universel « * »: la règle s'applique à n'importe quel élément
- Sélecteur d'élément (on utilise le nom d'une balise): la règle s'applique à tous les éléments correspondant à cette balise
- Sélecteur par identificateur: de la forme **#id**, il concerne le seul élément identifié par **id** dans le document HTML
- Sélection par classe: de la forme **.classe**, il s'applique à tous les éléments correspondant à cette classe dans le document HTML

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
em {
    ...
}
```

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
em {
    ...
}
```

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p {
    …
}
```

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p {
    …
}
```

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
#wrap {
    ...
}
```

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
#wrap {
    ...
}
```

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item {
    ...
}
```

Sélecteurs simples - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item {
    ...
}
```

Sélecteurs hiérarchiques

- Si on utilise la forme suivante

```
sel1 sel2 sel3 {
    …
}
```

  On cherche un élément désigné par **sel3** qui se trouve à l'intérieur d'un élément désigné par **sel2**, lui-même à l'intérieur d'un élément désigné par **sel1**

- Il n'est pas nécessaire que ce soit des enfants directs
- Si on veut spécifier un enfant direct, on utilise le symbole « > »

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
div a {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a> </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
div a {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item em {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item em {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item * {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item * {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
body p {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
body p {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
body > p {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
body > p {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
body p > a {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
body p > a {
    ...
}
```

Sélecteurs de voisins

- Forme **elem1 ~ elem2**:
  - L'élément qui nous intéresse est celui indiqué à droite, soit **elem2**
  - Il faut qu'il soit situé après **elem1** et au même niveau (ils ont le même parent)
- Forme **elem1 + elem2**
  - L'élément qui nous intéresse est celui indiqué à droite, soit **elem2**
  - Il faut qu'il soit situé immédiatement après **elem1** et au même niveau

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item ~ .item {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item ~ .item {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item ~ p {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item ~ p {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item + p {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em>phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item + p {
    ...
}
```

Sélecteurs par attribut

- Forme **sel[attribut=valeur]**
  - On identifie l'élément à la fois par un sélecteur et un attribut
  - Le sélecteur est facultatif
  - La valeur est elle aussi facultative (dans ce cas, seule l'existence de l'attribut est prise en compte, peu importe sa valeur)

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p[title=important] {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p[title=important] {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item[title=important] {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
.item[title=important] {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
[class] {
    ...
}
```

Sélecteurs hiérarchiques Exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
[class] {
    ...
}
```

Pseudo-éléments

- Les pseudo-éléments sont précédés des symboles **::**
- **::after** et **::before** permettent d'ajouter un élément virtuel (on utilise pour ce faire l'attribut **content**) juste avant ou juste après un élément de référence
- **::first-letter** pour désigner la première lettre d'un élément contenant du texte
- **::first-line** pour désigner la première ligne d'un élément contenant du texte

Pseudo-éléments - exemples

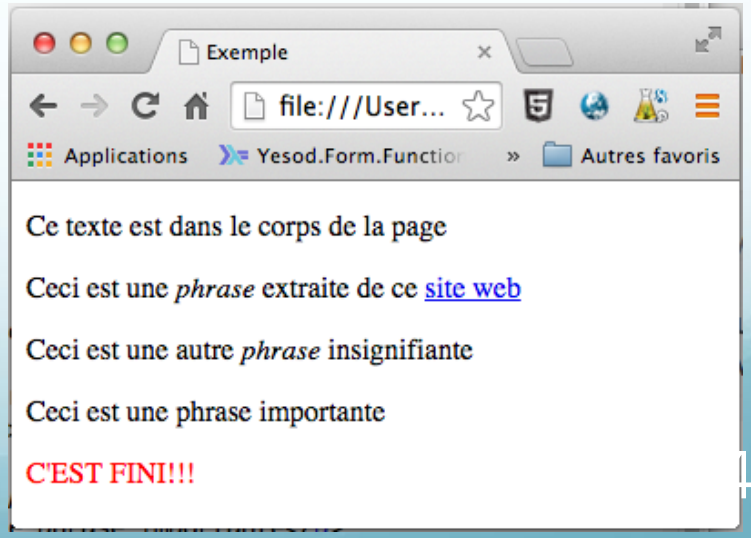
```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la page  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
#wrap::after {
   content: "C'EST FINI!!!";
   color: red;
}
```

Pseudo-éléments - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la page  </p>
    <div id="wrap">
       <p class="item" title="important">
       Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
       </p>
       <p> Ceci est une autre <em>phrase</em> insignifiante </p>
       <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
#wrap::after {
   content: "C'EST FINI!!!";
   color: red;
}
```



Pseudo-éléments - exemples

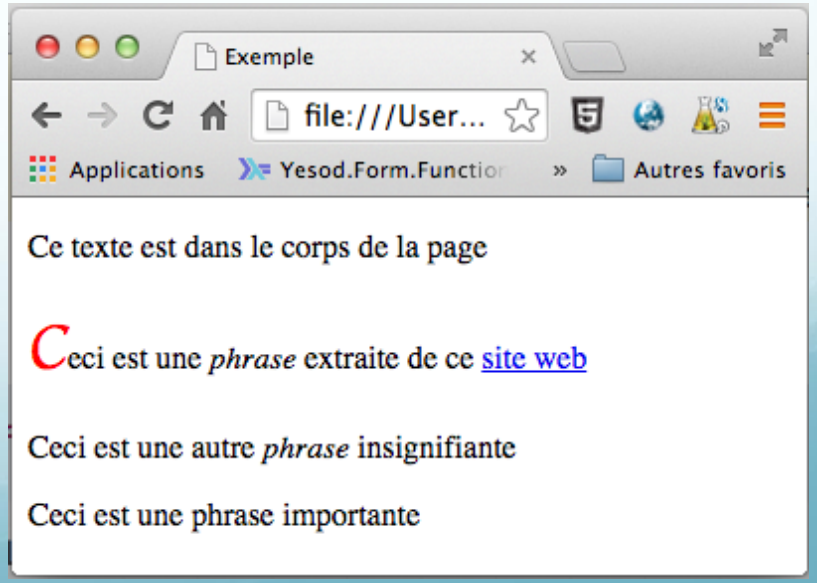
```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la page </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p[title=important]::first-letter {
   font-family: cursive;
   font-size: xx-large;
   color: red;
}
```

Pseudo-éléments - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la page  </p>
    <div id="wrap">
       <p class="item" title="important">
       Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
       </p>
       <p> Ceci est une autre <em>phrase</em> insignifiante </p>
       <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p[title=important]::first-letter {
   font-family: cursive;
   font-size: xx-large;
   color: red;
}
```



Pseudo-classes

- Les pseudo-classes sont précédées du symbole **:**
- Sert en général à indiquer l'état dans lequel se trouve un élément
- Exemples:
  - **:hover** : La souris est au dessus de l'élément
  - **:active** : L'élément est actif (on a cliqué sur le bouton de la souris mais on ne l'a pas encore relâché)
  - **:first-child** et **:last-child** : L'élément est est le premier et dernier fils (respectivement) de son parent
  - **:visited** : Il s'agit d'un hyper-lien qui a déjà été visité
  - **:not()** : Pour représenter la négation d'un sélecteur
  - **:focus** : Désigne l'élément qui a le focus actuellement
  - **:empty** : L'élément n'a aucun fils (texte ou élément)

Pseudo-classes - exemples

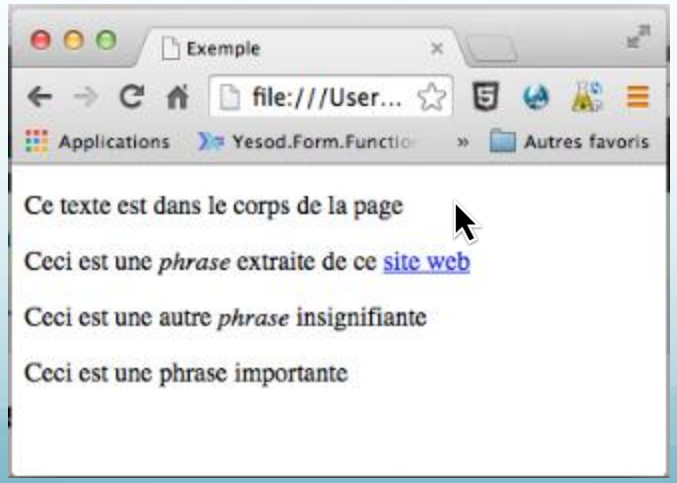
```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la page  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

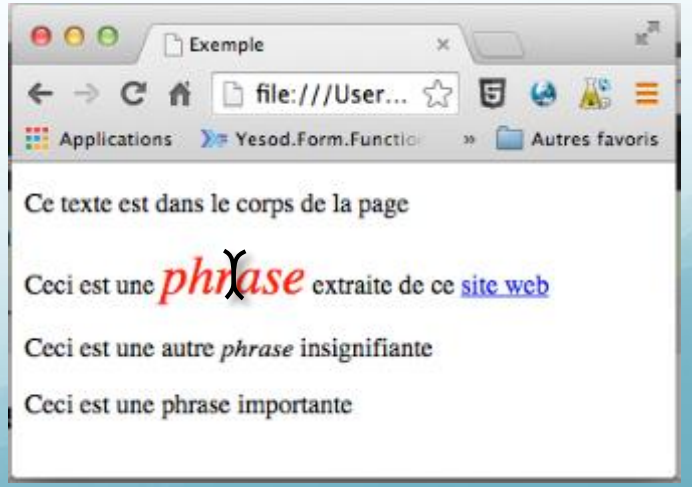
```
em:hover {
    color: red;
   font-size: xx-large;
}
```

Pseudo-classes - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la page  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
em:hover {
   color: red;
   font-size: xx-large;
}
```





Pseudo-classes - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p:last-child {
   …
}
```

Pseudo-classes - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p:last-child {
    ...
}
```

Pseudo-classes - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p:not(.item) {
   ...
}
```

Pseudo-classes - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p:not(.item) {
    ...
}
```

Pseudo-classes - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p:not(:first-child):not(.item) {
   ...
}
```

Pseudo-classes - exemples

```
<!doctype html>
<html>
  <head><title>Exemple</title></head>
  <body>
    <p> Ce texte est dans le corps de la <a href="#">page</a>  </p>
    <div id="wrap">
        <p class="item" title="important">
        Ceci est une <em >phrase</em> extraite de ce <a href="#">site web</a>
        </p>
        <p> Ceci est une autre <em>phrase</em> insignifiante </p>
        <p class="item">Ceci est une phrase importante</p>
    </div>
  </body>
</html>
```

```
p:not(:first-child):not(.item) {
   ...
}
```

Types de valeurs

- Mot-clé:
  - Il s'agit d'un mot-clé désignant une valeur spécifique
  - Exemples: **red**, **bold**, **solid**
- Dimension:
  - Il y a 3 unités relatives pour représenter une dimension:
    - **em**: taille de la police courante
    - **ex**: taille de la lettre x de la police courante
    - **px**: taille d'un pixel
  - Unités absolues: **mm** (millimètre), **cm** (centimètre), **in** (pouce), **pt** (point = 0,353mm), **pc** (pica = 12 pts)
  - Pourcentage: une dimension peut aussi être exprimée en pourcentage (en général par rapport à la largeur d'un bloc)

Types de valeurs

- Couleur:
  - Certaines couleurs sont représentées par un mot-clé: **red**, **yellow**, **fuchsia**, etc. (cet usage est déconseillé)
  - En général, on utilise la forme **#rrggbb**, où **rr**, **gg** et **bb** sont des nombres hexadécimaux représentant les tons de rouge, vert et bleu, respectivement (à noter qu'on peut utiliser seulement 3 chiffres, par exemple **#09f**, ce qui correspond aux 3 chiffres doublés: **#0099ff**
  - On peut aussi utiliser la forme **rgba(r,g,b,o)**, où **r**, **g** et **b** sont des valeurs entre 0 et 255 et où **o** est une valeur entre 0 et 1 représentant l'opacité (si **o** = 1 tout ce qui derrière est caché par la couleur, si **o** = 0 la couleur est complètement transparente)
  - On peut aussi utiliser la forme **#rrggbb** et définir l'opacité séparément en utilisant l'attribut **opacity**
  - Il existe aussi deux autres formats: **hsl** et **hsla**

Types de valeurs

- Valeur numérique
- Chaîne de caractères: délimitée par « " » ou « ’ »
- Une URI: **url(URI)**, où **URI** pointe sur une ressource (en général une image) qui sera téléchargée
- Notation abrégée:
  - Dans certains cas, plusieurs propriétés semblables s’appliquent à un objet
  - Par exemple, les quatre marges peuvent être établies de deux manières équivalentes (notez l’ordre *top*, *right*, *bottom*, *left*)

```
article {
    margin-top:     10pt ;
    margin-right:   12pt ;
    margin-bottom:  5pt ;
    margin-left:    0pt ;
}
```

```
article {
    margin:  10pt 12pt 5pt 0pt ;
}
```

Types de valeurs

- Notation abrégée (suite):
  - On peut aussi retirer la quatrième valeur (dans ce cas, *left* et *right* auront la même valeur):

```
article {
    margin-top:     10pt ;
    margin-right:   12pt ;
    margin-bottom:  5pt ;
    margin-left:    12pt ;
}
```

```
article {
    margin: 10pt 12pt 5pt ;
}
```

  - Si on laisse deux valeurs, la première correspond à *top* et *bottom*, alors que la seconde correspond à *right* et *left*:

```
article {
    margin-top:     10pt ;
    margin-right:   12pt ;
    margin-bottom:  10pt ;
    margin-left:    12pt ;
}
```

```
article {
    margin: 10pt 12pt ;
}
```

  - Finalement, si on met une seule valeur, elle sera égale pour tous les quatre: `article { margin: 10pt }`

Types de valeurs

- Notation abrégée (suite):
  - Dans d'autres cas, les valeurs sont des mots-clés distincts
  - On peut donc les combiner, totalement ou partiellement en un seul attribut
  - Pour définir un arrière-plan, les deux règles suivantes sont équivalentes:

```
article {
   background-color: #fff ;
   background-image: url(bg.png) ;
   background-repeat: no-repeat ;
   background-position: right top ;
}

article {
  background: #fff url(bg.png) no-repeat right top ;
}
```

  - Remarque: dans le cas d'une valeur qui n'est pas spécifiée, une valeur par défaut sera utilisée

Attention!!!

Si vous écrivez ceci:

```
article {
   background: #fff ;
   background: url(bg.png) ;
}
```

Vous n'aurez pas un fond blanc, car la deuxième déclaration écrase la première!

Cascade

- Il faut imaginer que les règles CSS sont regroupées en niveaux et que ces niveaux sont traités en cascade
- Ces niveaux sont ordonnés selon quatre critères: *importance*, *origine*, *spécificité* et *ordre* d'apparition des déclarations
- Si malgré cette cascade on ne réussit pas à identifier une valeur pour un attribut, on pourra dans certains cas en hériter de l'élément parent

Cascade

- Pour identifier la valeur d'un attribut d'un élément, la cascade fonctionne de la manière suivante:
  - Trouver toutes les déclarations CSS qui s'appliquent
  - Trier les déclarations selon leur niveaux d'importance et leur origine
  - Dans chacun des niveaux, trier les déclarations selon leur spécificité
  - Lorsque des déclarations ont la même spécificité, les trier en ordre d'apparition, la dernière occurrence étant la plus prioritaire

Origine et importance

- Les origines possibles de déclarations combinées avec l'importance, nous donnent les 5 niveaux suivants:
  - Styles par défaut définis par le navigateur
  - Déclarations normales définies par l'utilisateur du navigateur
  - Déclarations normales utilisées par le document HTML
  - Déclarations importantes utilisées par le document HTML
  - Déclarations importantes définies par l'utilisateur du navigateur
- Une déclaration est importante si elle est suivie du terme **`!important`**:

```
strong {
   color: red !important ;
   font-weight: bold ;
   border: solid 1px ;
}
```

Spécificité

- Pour chaque niveau, les déclarations sont triées selon leur spécificité
- La spécificité est un nombre à 4 chiffres ABCD où
  - A = 1 s'il s'agit d'une déclaration faite directement à l'intérieur de la balise dans un document HTML, 0 sinon (exemple: `<p style="color: red;">bla bla </p>`)
  - B = nombre d'identificateurs apparaissant dans le sélecteur
  - C = nombre de classes, attributs et pseudo-classes apparaissant dans le sélecteur
  - D = nombre d'éléments et pseudo-éléments apparaissant dans le sélecteur
- Plus la valeur de spécificité est élevée, plus la déclaration est prioritaire

Spécificité - exemple

```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

Adapté de http://www.sitepoint.com/web-foundations/specificity/

Spécificité - exemple


```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

Spécificité - exemple

```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

Spécificité - exemple

```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

Spécificité - exemple

```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

0011

0211

0111

0213

Spécificité - exemple

```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

0011

0211

0111

0213

0001

Spécificité - exemple

```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

0011

0211

0111

0213

0001

0213

Spécificité - exemple

```
p.message {
  color: lime;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
body#home div#warning p.message {
  color: blue;
}
p {
  color: orange;
}
* div#home > div#warning p[title=important] {
  color: red;
}
#warning p {
  color: black;
}
```

0011

0211

0111

0213

0001

0213

0101

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

Déclarations triées selon leur spécificité.

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

```
<body>
  <div>
    <p> ... <p>
  </div>
</body>
```

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

```
<body>
  <div>
    <p class= "message"> ... <p>
  </div>
</body>
```

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

```
<body>
  <div id="warning">
    <p> ... <p>
  </div>
</body>
```

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

```
<body>
  <div id="warning">
    <p class= "message"> ... <p>
  </div>
</body>
```

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

```
<body>
  <article id="home">
    <div id="warning">
      <p class= "message"> ... <p>
    </div>
  </article>
</body>
```

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

```
<body>
  <div id="home">
    <div id="warning">
      <p class= "message"> ... <p>
    </div>
  </div>
</body>
```

Spécificité - exemple

```
* div#home > div#warning p[title=important] {
  color: red;
}
body#home div#warning p.message {
  color: blue;
}
#home #warning p.message {
  color: yellow;
}
#warning p.message {
  color: fuchsia;
}
#warning p {
  color: black;
}
p.message {
  color: lime;
}
p {
  color: orange;
}
```

0213

0213

0211

0111

0101

0011

0001

```
<body>
  <div id="home">
    <div id="warning">
      <p title="important"> ... <p>
    </div>
  </div>
</body>
```

Exemple de cascade

```
<!DOCTYPE  html>
<html>
  <head>
    <style type="text/css">
      body {
        color: #000;
        background-color: #fff;
      }
      #wrap {
        font-size: 2em;
        color: #333;
      }
      div {
        font-size: 1em;
      }
      em {
        color: #666;
      }
      p.item {
        color: #fff;
        background-color: #ccc;
        border-style: dashed;
      }
      p {
        border: 1px solid black;
        padding: 0.5em;
      }
    </style>
  </head>
  <body>
    <p> Ce texte est dans le corps de
          la page </p>
    <div id="wrap">
      <p class="item">
        Ceci est une <em>cascade</em> en
        <a href="#">action</a>
      </p>
    </div>
  </body>
</html>
```

Adapté de http://www.sitepoint.com/web-foundations/cascade/

Exemple de cascade

Exemple de cascade

- On remarque que deux règles s'appliquent pour déterminer la bordure du deuxième élément p:

```
p.item {
  color: #fff;
  background-color: #ccc;
  border-style: dashed;
}
p {
  border: 1px solid black;
  padding: 0.5em;
}
```

- C'est la première qui s'applique puisqu'elle est plus spécifique

Exemple de cascade (suite)

- On remarque que la taille de la police pour le deuxième élément p est plus grande
- Aucune règle ne spécifie cette taille pour l'élément p
- Par contre, la taille est définie pour son élément parent:

```
#wrap {
    font-size: 2em;
     color: #333;
}
```

- Il hérite donc de cette valeur

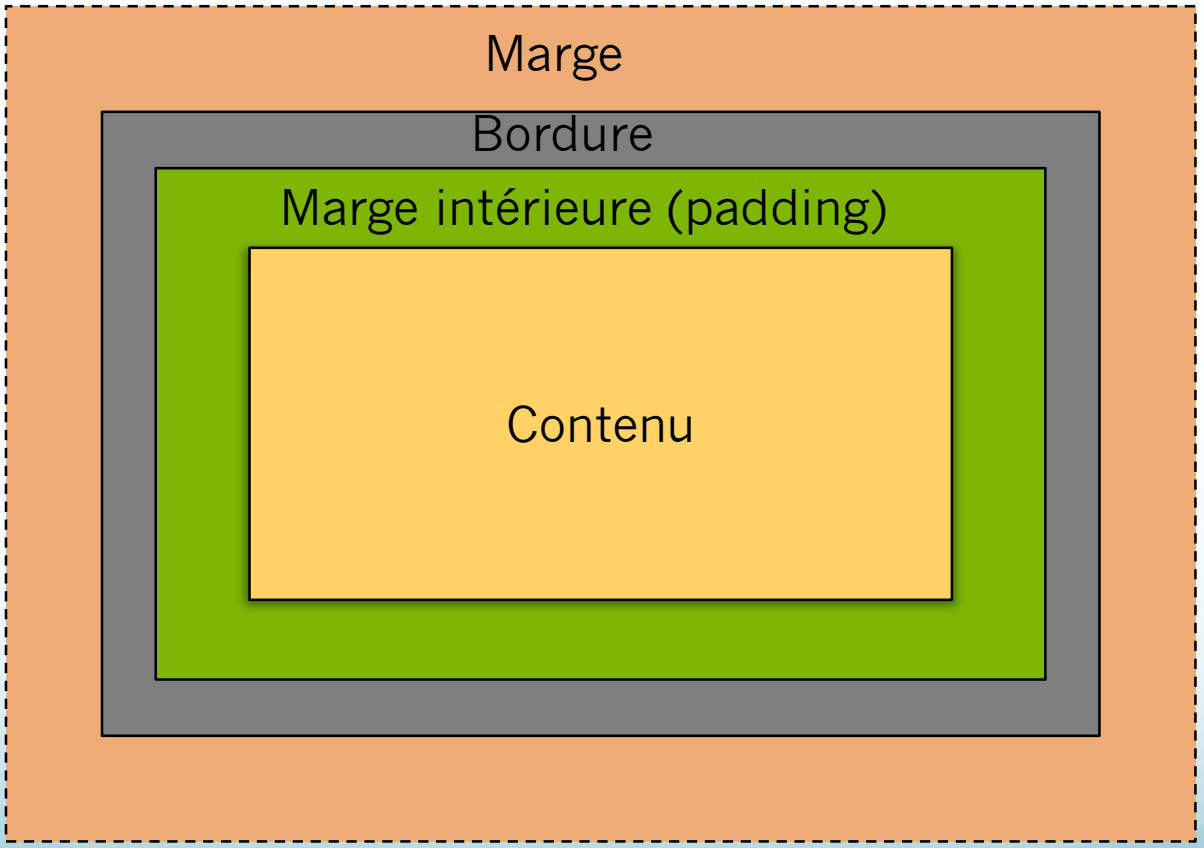
Propriétés

- Propriétés de boîte: *height*, *min-height*, *max-height*, *width*, *min-width*, *max-width*, *margin*, *padding*, *border*
- Propriétés de mise en page: *display*, *position*, *float*, *clear*, *visibility*, *top*, *right*, *bottom*, *left*, *z-index*, *overflow*, *clip*
- Propriétés de listes
- Propriétés de tables
- Couleurs et propriétés d'arrière-plan
- Propriétés de textes
- Ajouts de CSS3: transitions, animations, gradient, coins arrondis,

Trois modèles de mise en page

- Modèle de boîtes positionnées
- Modèle de boîtes flexibles
- Modèles de grille

Boîtes positionnées



Boîtes positionnées

- S'applique aux éléments de type bloc
- Les dimensions de la boîte de contenu sont déterminées par les attributs **width** et **height**
- Une boîte peut être positionnée de cinq manières différentes:
  - *Statique* (aucun positionnement spécifié – c'est la valeur par défaut)
  - *Fixe* (positionné par rapport au coin supérieur gauche de la fenêtre)
  - *Absolue* (positionné par rapport au coin supérieur gauche de la boîte englobante)
  - *Relative* (on décale la boîte par rapport à son emplacement naturel dans le flux
  - *Flottante* (la boîte se place à droite ou à gauche et le texte la contourne)
- Attention: en général, les marges de deux boîtes successives seront combinées en une seule, soit la plus grande des deux.

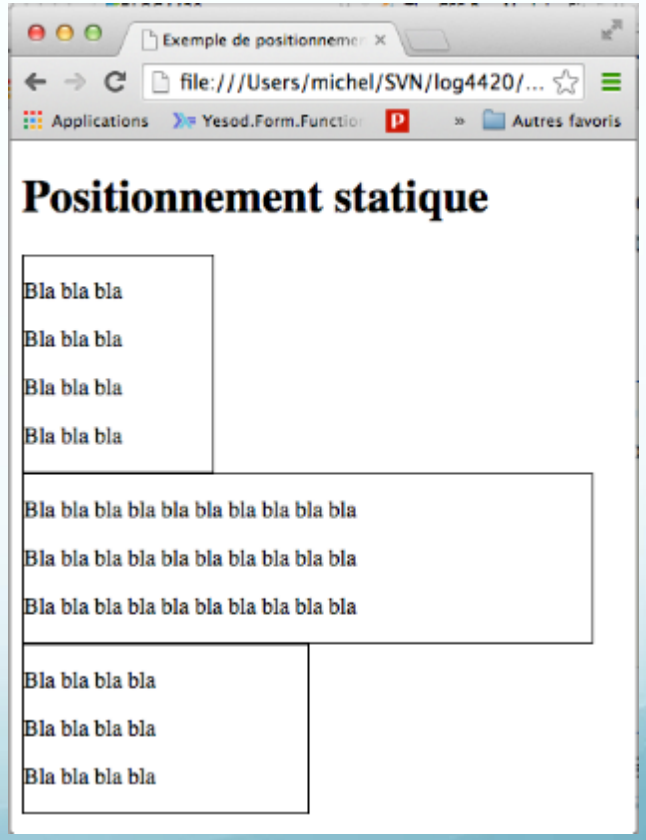
Propriété **display**

- Valeurs possibles:
  - **block** : pour traiter comme un bloc ce qui ne l'est pas normalement
  - **inline** : permet de traiter comme un item en-ligne ce qui serait normalement un bloc
  - **block-inline** : on traite l'élément à la fois comme bloc et item en-ligne (ce qui signifie qu'on peut lui attribuer des dimensions)
  - **none** : aucune boîte ne sera générée (le document sera affiché comme si l'élément n'existait pas)
  - D'autres valeurs pour présentation sous forme de table et de liste
  - D'autres valeurs pour les boîtes flexibles

Positionnement statique

- Les boîtes sont positionnées selon un flux naturel, soit une au dessous de l'autre:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <style>
    div { border: solid 1pt; }
    #div1 { width: 100pt; }
    #div2 { width: 300pt; }
    #div3 { width: 150pt; }
    </style>
  </head>

  <body>
    <h1>Positionnement statique</h1>
    <div id="div1"> ... </div>
    <div id="div2"> ... </div>
    <div id="div3"> ... </div>
  </body>
```

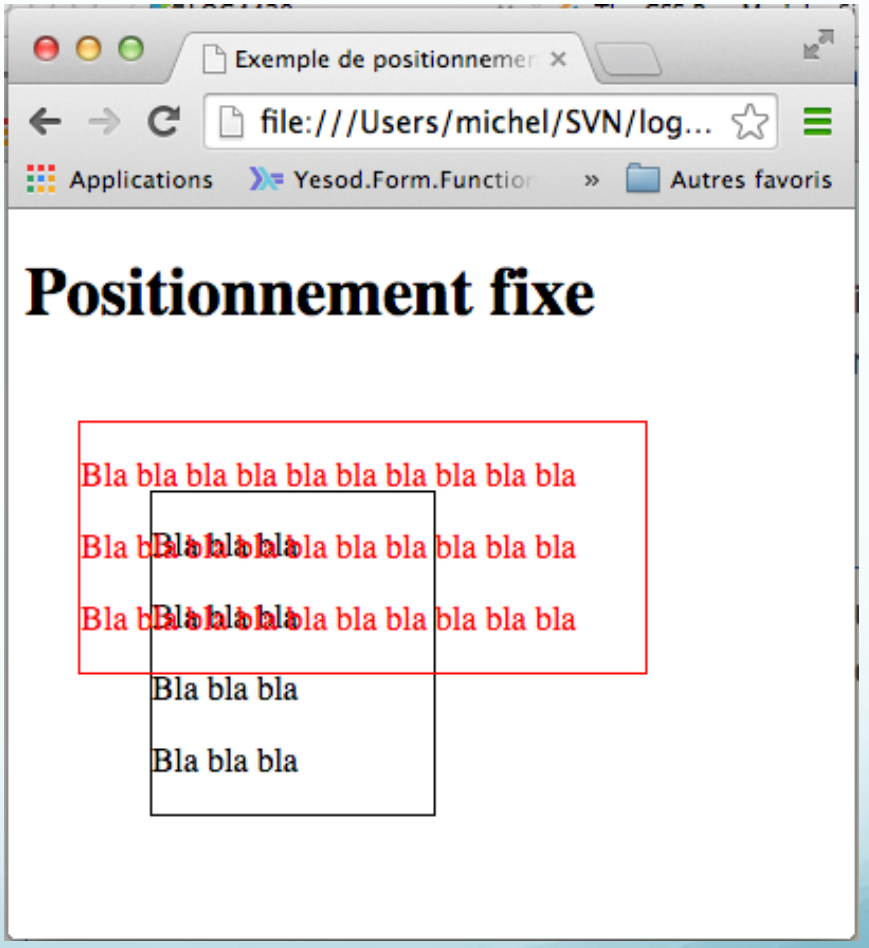


Positionnement fixe

- La position d'un coin de la boîte est déterminé par rapport au coin correspondant de la fenêtre
- On fixe la position avec les attributs suivants:
  - **left** : distance par rapport à la bordure gauche de la fenêtre
  - **top** : distance par rapport à la bordure supérieure de la fenêtre
  - **right** : distance par rapport à la bordure droite de la fenêtre
  - **top** : distance par rapport à la bordure inférieure de la fenêtre

Positionnement fixe

```
div {
   position: fixed;
   border: solid 1pt;
}
#div1 {
   width: 100pt;
   top: 100pt;
   left: 50pt;
}
#div2 {
   width: 200pt;
   top: 75pt;
   left: 25pt;
   color: red;
}

<body>
    <h1>Positionnement fixe</h1>
    <div id="div1"> ... </div>
    <div id="div2"> ... </div>
</body>
```



Positionnement absolu

- La position du coin supérieur gauche de la boîte est déterminé par rapport au coin supérieur gauche de la boîte englobante
- La boîte englobante est la première trouvée qui est positionnée de manière *fixe*, *relative* ou *absolue*
- On fixe la position avec les attributs suivants:
  - **left** ou **right** : distance par rapport à la bordure gauche (ou droite) de la boîte englobante
  - **top** ou **bottom** : distance par rapport à la bordure supérieure (ou inférieure) de la boîte englobante

Positionnement absolu

```
div {
   border: solid 1pt;
}
#div0 {
    position: relative;
    width: 100pt;
    height: 100pt;
    color: red;
}
#div1 {
    position: absolute;
    width: 50pt;
    top: 25pt;
    left: 25pt;
    color: blue;
}
```

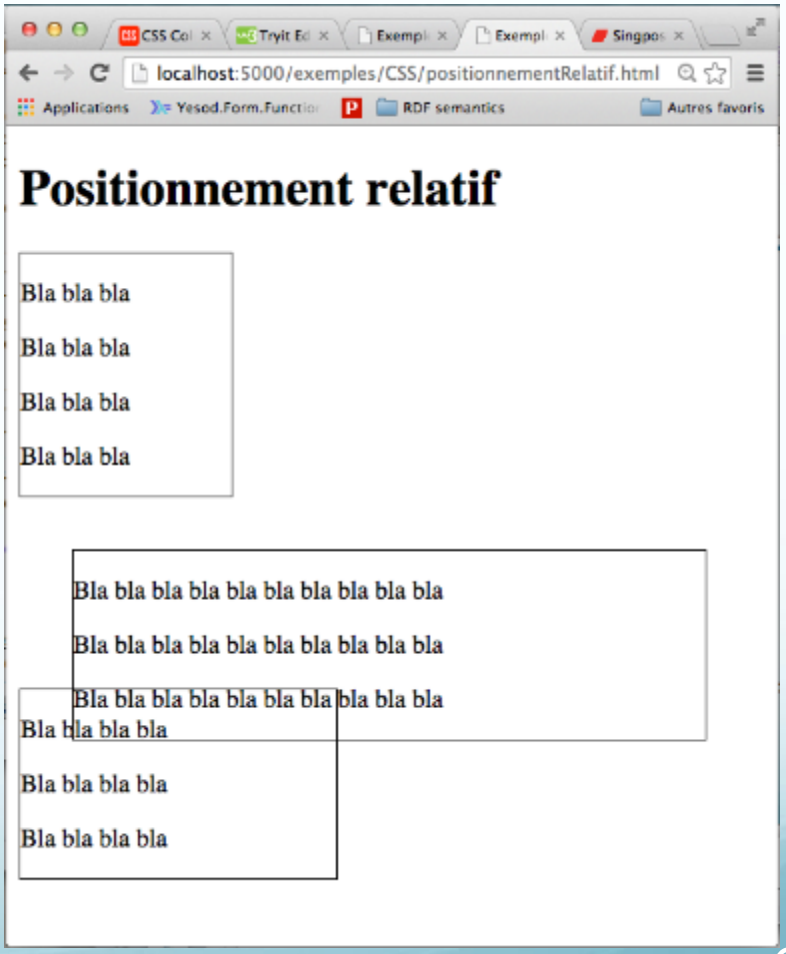
```
<body>
   <h1>Positionnement absolu (avec boîte
         englobante) </h1>
   <div id="div0">
       <div id="div1"> ... </div>
   </div>
</body>
```

Positionnement relatif

- On calcule d'abord quelle serait la position normale de la boîte
- Ensuite, on la décale par rapport à cette position normale

Positionnement relatif

```
div {
    border: solid 1pt;
}

#div1 {
    width: 100pt;
}

#div2 {
    position: relative;
    width: 300pt;
    left: 25pt;
    top: 25pt;
 }
#div3 {
    width: 150pt;
 }
<body>
    <h1>Positionnement relatif</h1>
    <div id="div1"> ... </div>
    <div id="div2"> ... </div>
    <div id="div3"> ... </div>
</body>
```

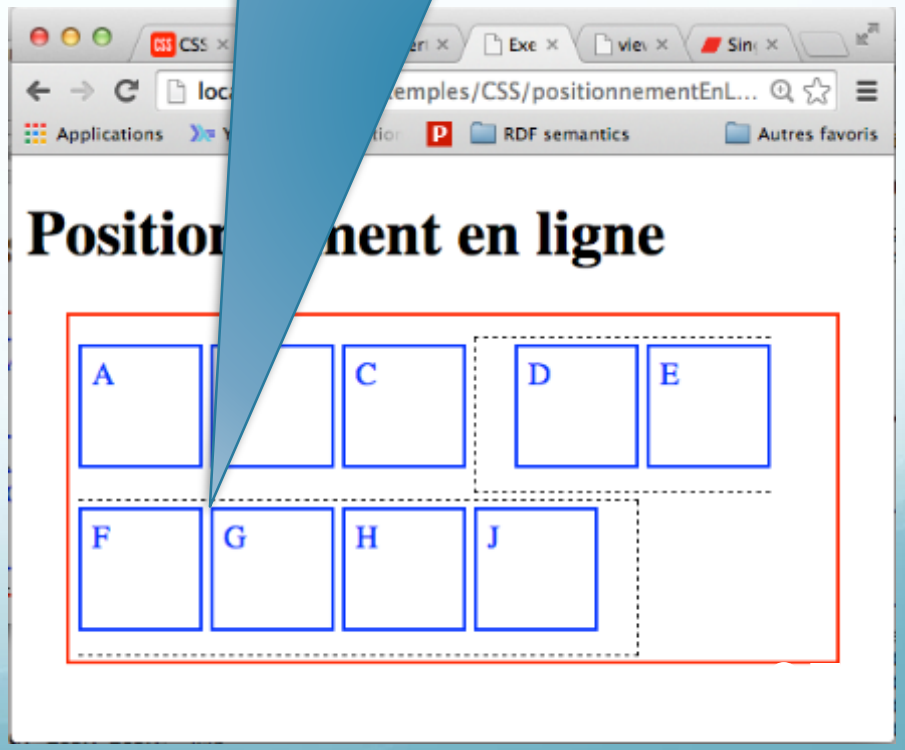


Positionnement en ligne

```
div {
    position: relative;
    border: solid 2px;
    color: red;
    margin: 20px 20px;
    padding: 5px;
}
#div2 div {
    display: inline-block;
    height: 50px;
    width: 50px;
    border: solid 2px;
    color: blue;
    margin: 10px auto;
}
#div2 span {
    border: dashed black 1px;
    padding: 10px 5px 40px;
}
```

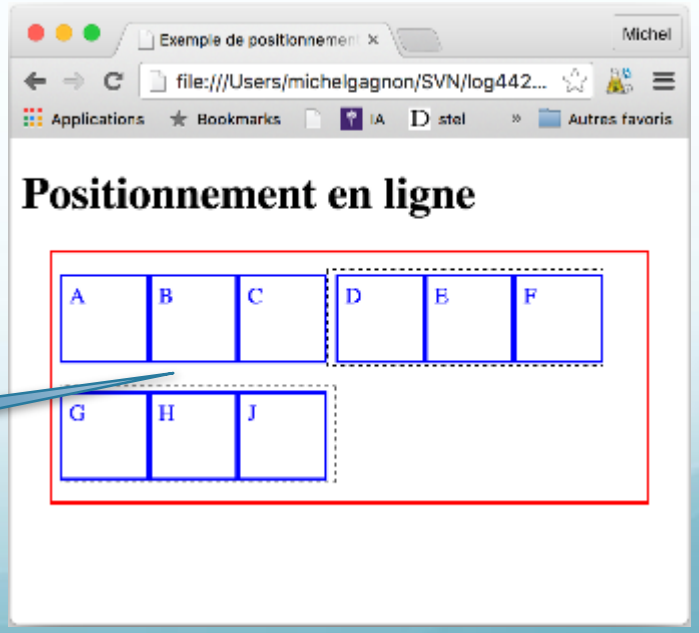
```
<body>
  <h1>Positionnement en
ligne</h1>
  <div id="div2">
    <div>A</div>
    <div>B</div>
    <div>C</div>
    <span>
      <div>D</div>
      <div>E</div>
      <div>F</div>
      <div>G</div>
      <div>H</div>
      <div>J</div>
    </span>
  </div>
</body>
```

Remarquez le petit espace qui s'ajoute entre les blocs (parce que le passage à la ligne suivante dans le HTML est interprété comme une espace)



Positionnement en ligne

```
<body>
  <h1>Positionnement en ligne</h1>
  <div id="div2">
      <div>A</div><div>B</div><div>C</div><span><div>D</div><div>E</div><div>F</div<div>G</div><div>H</div><div>J</div></span>
  </div>
</body>
```



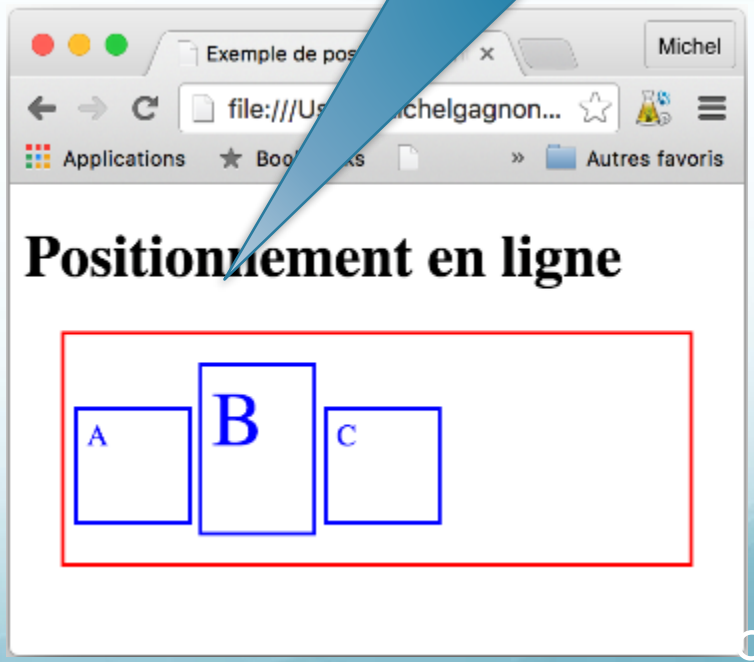
Les espaces ont disparu

Positionnement en ligne

```
div {
    position: relative;
    border: solid 2px;
    color: red;
    margin: 20px 20px;
    padding: 5px;
}
#div2 div {
    display: inline-block;
    height: 50px;
    width: 50px;
    border: solid 2px;
    color: blue;
    margin: 10px auto;
}
#div2 #div3 {
    font-size:  42px;
    height: 80px;}
```

```
<body>
  <h1>Positionnement en
ligne</h1>
  <div id="div2">
    <div>A</div>
    <div = "div3">B</div>
    <div>C</div>
</div>
</body>
```

Si les blocs sont de tailles différentes, ils sont alignés en fonction du texte qu'ils contiennent

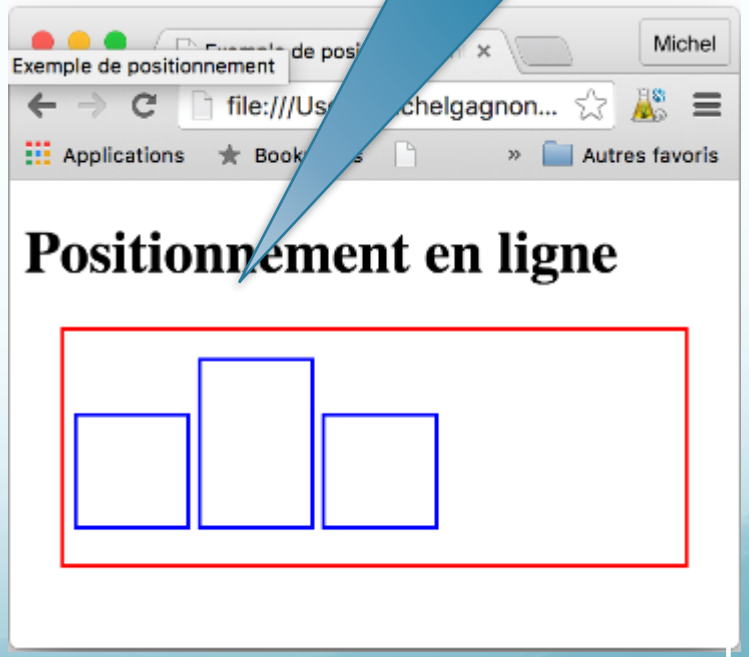


Positionnement en ligne

```
div {
    position: relative;
    border: solid 2px;
    color: red;
    margin: 20px 20px;
    padding: 5px;
}
#div2 div {
    display: inline-block;
    height: 50px;
    width: 50px;
    border: solid 2px;
    color: blue;
    margin: 10px auto;
}
#div2 #div3 {
    font-size:  42px;
    height: 80px;}
```

```
<body>
  <h1>Positionnement en
ligne</h1>
  <div id="div2">
    <div></div>
    <div id= "div3"></div>
    <div></div>
</div>
</body>
```

Si les blocs sont de tailles différentes, ils sont alignés par la base s'il n'y a pas de contenu textuel

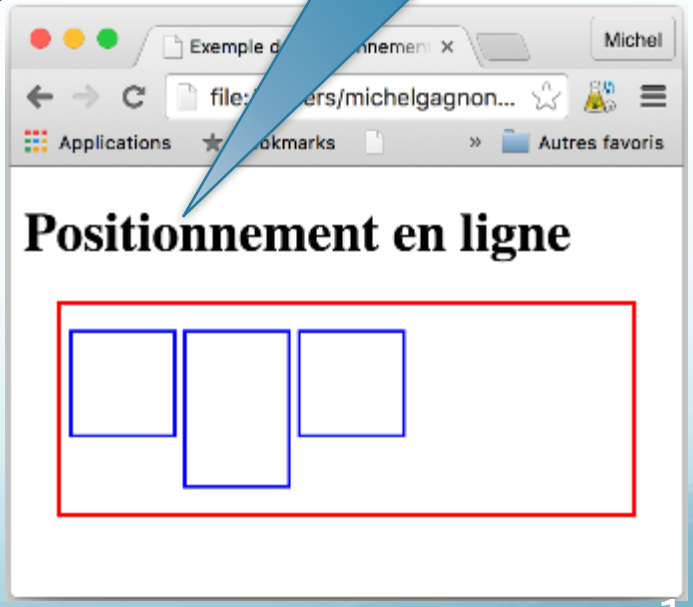


Positionnement en ligne

```
div {
    position: relative;
    border: solid 2px;
    color: red;
    margin: 20px 20px;
    padding: 5px;
}
#div2 div {
    display: inline-block;
    vertical-align: top;
    height: 50px;
    width: 50px;
    border: solid 2px;
    color: blue;
    margin: 10px auto;
}
#div2 #div3 {
    font-size:  42px;
    height: 80px;}
```

```
<body>
  <h1>Positionnement en
ligne</h1>
  <div id="div2">
    <div></div>
    <div id= "div3"></div>
    <div></div>
</div>
</body>
```

On peut changer ce comportement avec l'attribut vertical-align



Positionnement flottant

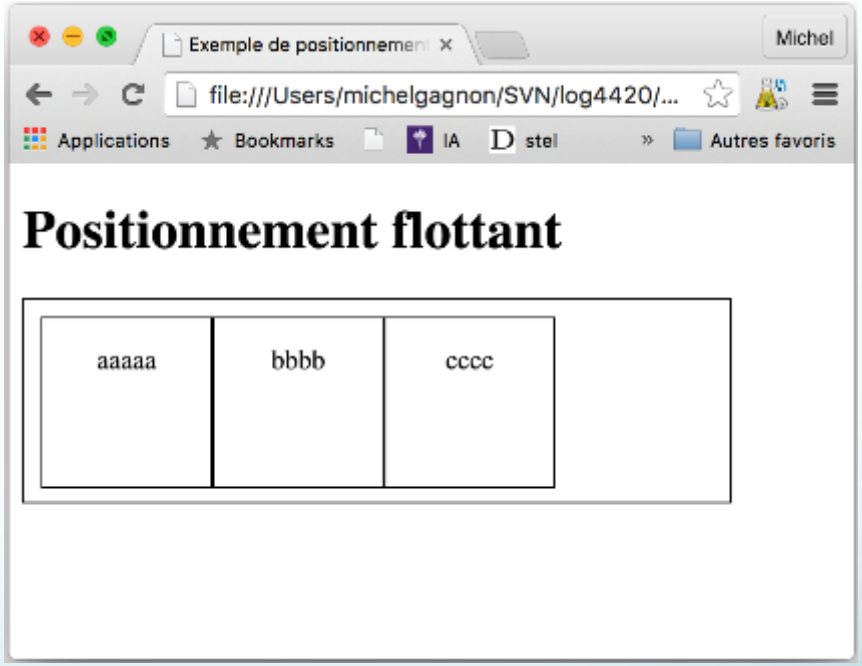
```
div {
   border: solid 1pt;
}
#div0 {
   width: 200pt;
}
#div1 {
   float: right;
   height: 50pt;
   width: 50pt;
   background-color: #ccc;
}

<body>
   <h1>Positionnement flottant</h1>
   <div id="div0">
      <div  id="div1">
            <p> Bla bla bla</p>
      </div>
      <p> ...</p>
   </div>
</body>
```

Positionnement flottant

Les boîtes s'empilent dans le sens indiqué

```
div { border: solid 1pt; }
#div0 {width: 400px;
        height: 100px;
        padding: 10px }
.flottant { float: left;
          height: 100px;
          width: 100px;
          border: solid 1px;  }
p { text-align: center }
```

```
<body>
   <h1>Positionnement flottant</h1>
   <div id="div0">
       <div class="flottant”><p> aaaaa</p></div>
       <div class="flottant”><p> bbbb</p></div>
       <div class="flottant”><p> cccc</p></div>
   </div>
</body>
```



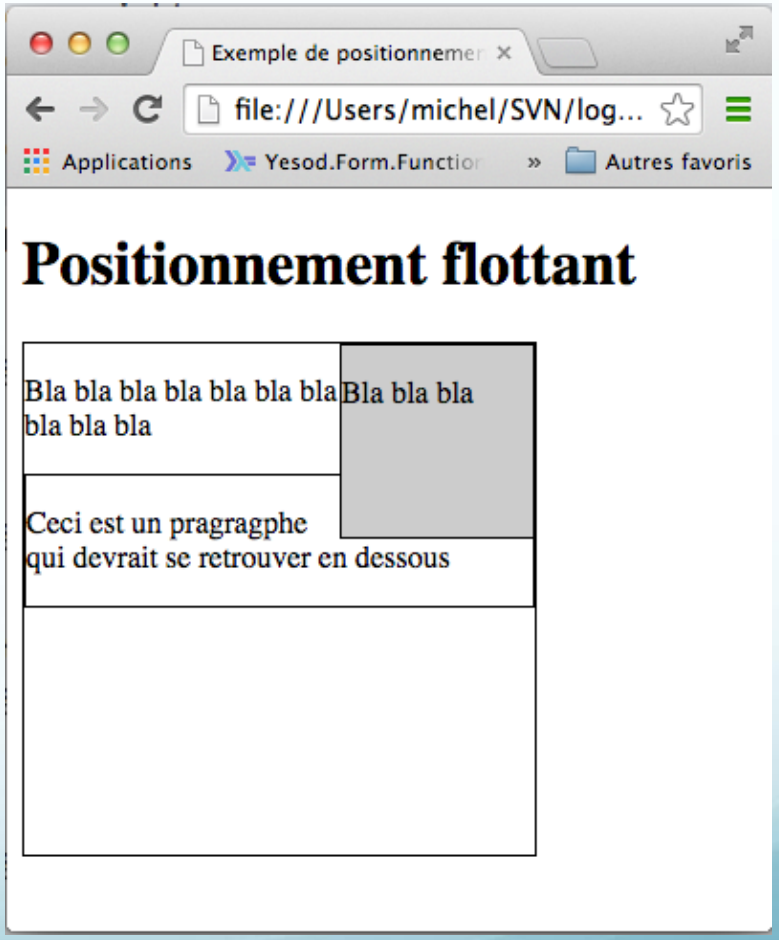
Positionnement flottant

- Dans certains cas, on voudrait que ce qui suit se retrouve en dessous de la boîte flottante
- Dans ce cas, on utilise l'attribut **clear**
- Trois valeurs possibles:
  - **right** – pour descendre en dessous d'une boîte flottante à droite
  - **left** – pour descendre en dessous d'une boîte flottante à gauche
  - **both** – pour descendre en dessous de boîtes flottantes des deux côtés

Attribut **clear**

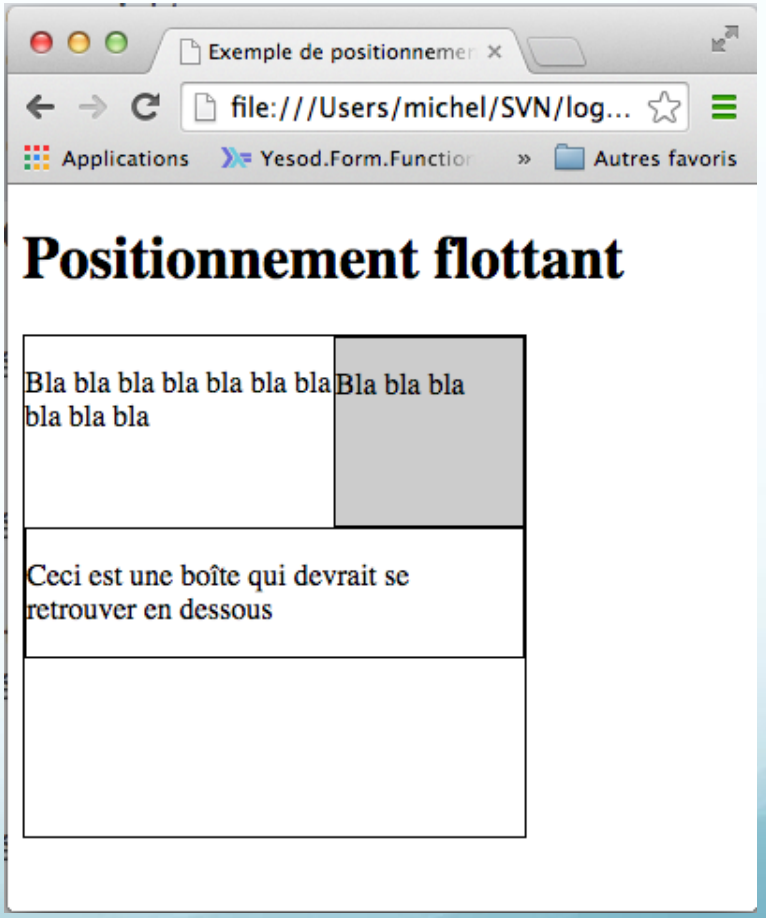
```
div {
    border: solid 1pt;
}
#div0 {
    width: 200pt;
    height: 200pt;
}
#div1 {
    float: right;
    height: 75pt;
    width: 75pt;
    background-color: #ccc;
}
```

```
<body>
    <h1>Positionnement flottant</h1>
    <div id="div0">
      <div  id="div1”><p> Bla bla bla</p></div>
      <p> Bla bla bla bla bla bla bla bla bla bla</p>
      <div> <p> Ceci est une boîte qui devrait
              se retrouver en dessous</p></div>
    </div>
</body>
```



Attribut **clear**

```
div { border: solid 1pt; }
#div0 {
    width: 200pt;
    height: 200pt;
}
#div1 {
    float: right;
    height: 75pt
     width: 75pt;
     background-color: #ccc;
}
#div2 {
     clear: right;
 }

<body>
  Positionnement flottant</h1>
    <div id="div0">  <h1>bla</p>
      <div  id="div1”><p> Bla bla </div>
      <p> Bla bla bla bla bla bla bla bla bla bla</p>
      <div id=“div2”> <p> Ceci est une boîte qui devrait
              se retrouver en dessous</p></div>
    </div>
</body>
```



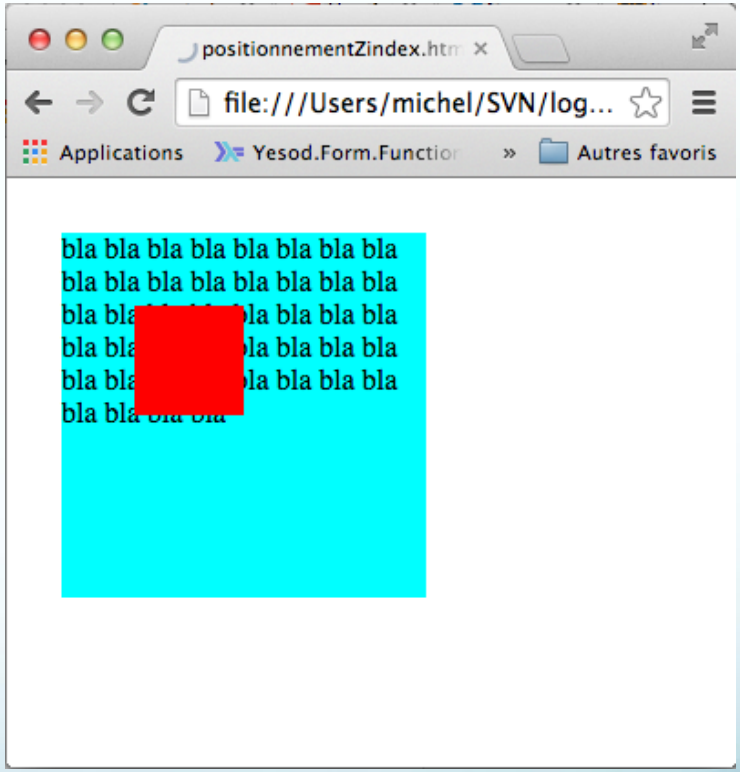
Propriété **z-index**

- Permet de contrôler la superposition d'éléments
- Basée sur le contexte d'empilement, composé de 7 couches (du fond vers le dessus):
  1. L'arrière-plan et sa bordure
  2. Les contextes d'empilement des éléments inclus qui ont un z-index négatif
  3. Les blocs inclus qui n'ont pas de valeur z-index (qui ne sont pas positionnés)
  4. Les éléments inclus qui sont déclarés flottants
  5. Les éléments en-ligne inclus
  6. Les éléments positionnés inclus dont la valeur z-index est 0 ou **auto**
  7. Les contextes d'empilement des éléments inclus qui ont un z-index positif
- Il faut que l'élément soit positionné de manière fixe, absolue, ou relative

Propriété **z-index**

```
#in {
    position: absolute;
    top: 40px;
    left: 40px;
    width: 60px;
    height: 60px;
    background: red;
    z-index: 6;
}
#div1 {
    position: fixed;
    left: 30px;
    top: 30px;
    background: aqua;
    width: 200px;
    height: 200px;
}
```

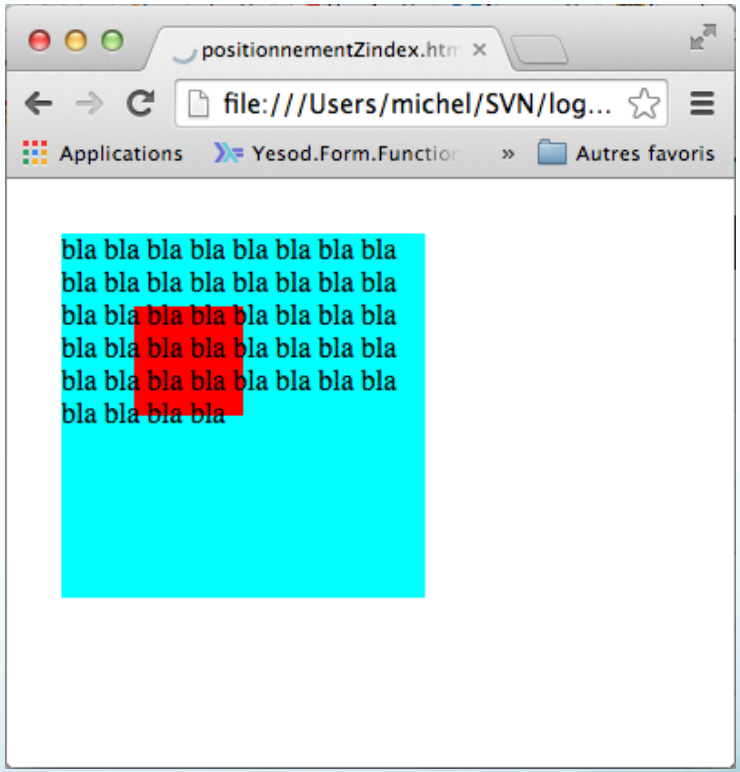
```
<body>
  <div id="div1">
     <div id="in"></div>
     <p>bla ...bla</p>
  </div>
</body>
```



Propriété **z-index**

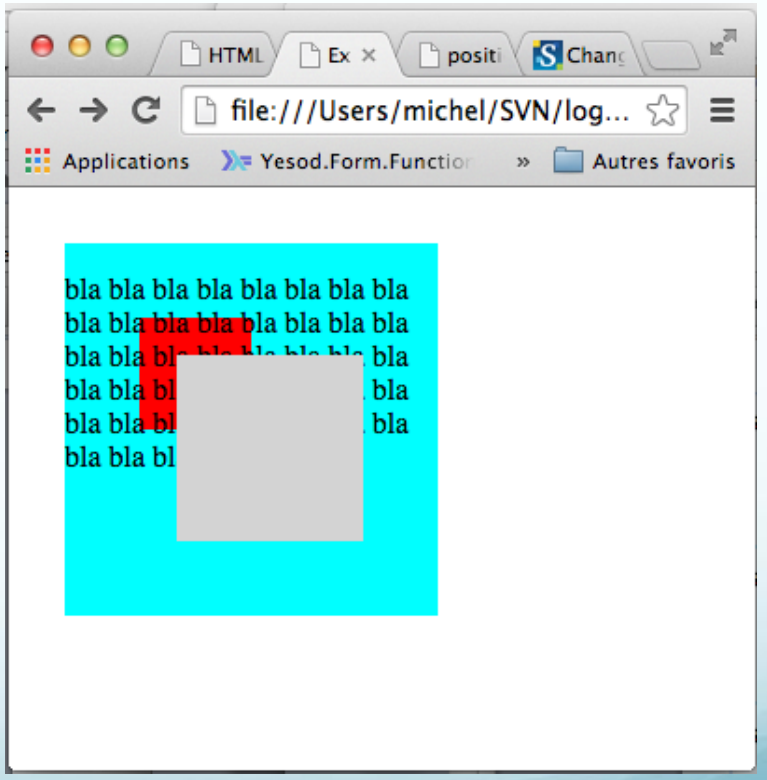
```
#in {
    position: absolute;
    top: 40px;
    left: 40px;
    width: 60px;
    height: 60px;
    background: red;
    z-index: -6;
}
#div1 {
    position: fixed;
    left: 30px;
    top: 30px;
    background: aqua;
    width: 200px;
    height: 200px;
}
```

```
<body>
  <div id="div1">
     <div id="in"></div>
     <p>bla ...bla</p>
  </div>
</body>
```



Propriété **z-index**

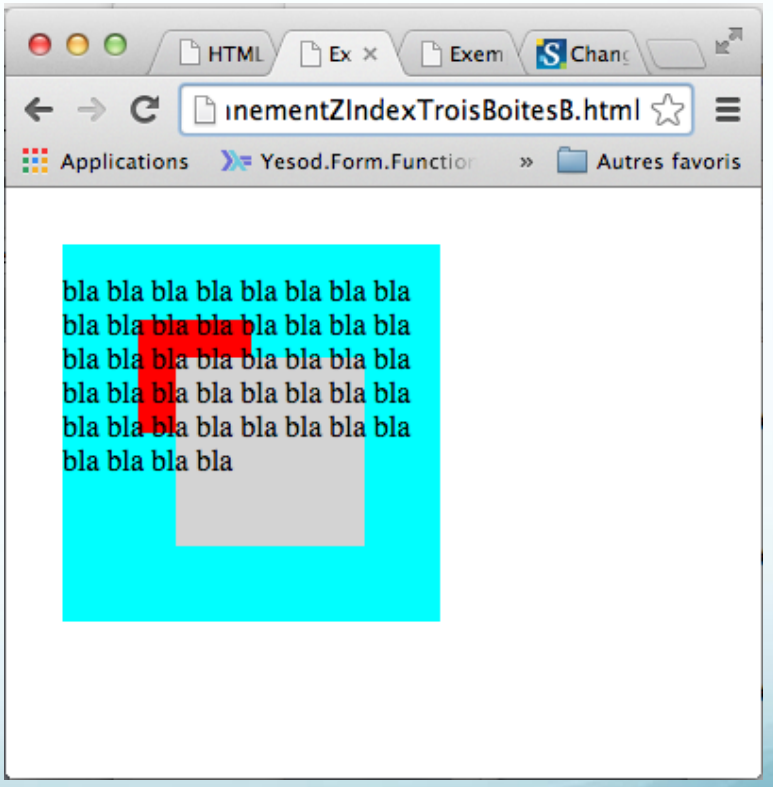
```
#in {
    ...
    background: red;
    z-index: -6;
}
#div1 {
    ...
    background: aqua;
}
#div2 {
    position: absolute;
    left: 60px;
    top: 60px;
    background: lightGray;
    width: 100px;
    height: 100px;
    z-index: 1;
}
<body>
   <div id="div1">
        <div id="in"></div>
        <p>bla ...bla</p>
        <div id="div2"></div>
   </div>
</body>
```



Propriété **z-index**

```
#in {
    ...
    background: red;
    z-index: -6;
}
#div1 {
    ...
    background: aqua;
    z-index: -2;
}
#div2 {
    position: absolute;
    left: 10px;
    top: 10px;
    background: lightGray;
    opacity: 0.8;
    width: 150px;
    height: 150px;
}
```

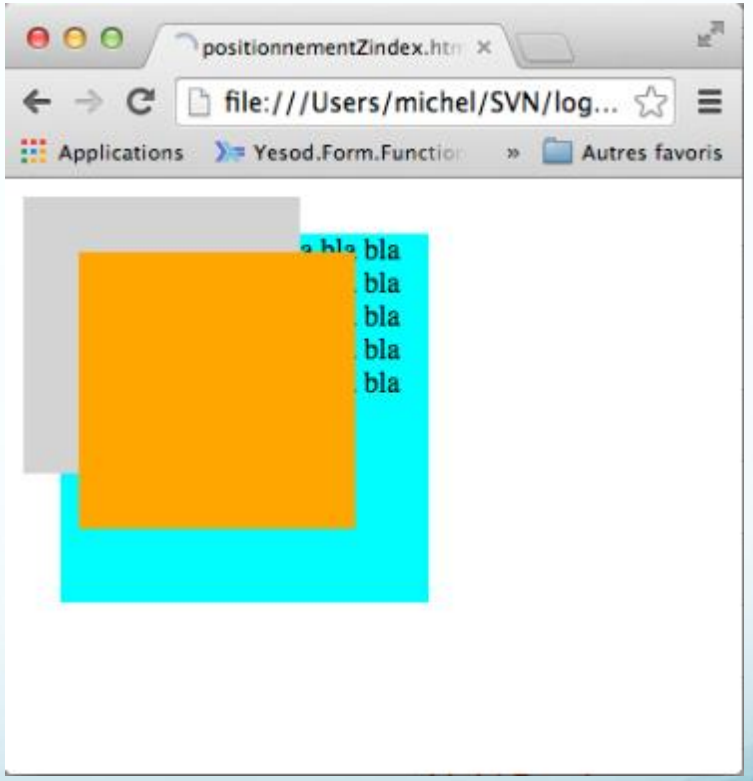
```
<body>
  <div id="div1">
        <div id="in"></div>
        <p>bla ...bla</p>
  </div>
  <div id="div2"></div>
</body>
```



Propriété **z-index**

```
#in {
    ...
    background: red;
}
#div1 {
    ...
    background: aqua;
    z-index: 1;
}
#div2 {
    ...
    background: lightGray;
    z-index: 2;
}
#div3 {
    ...
    background: orange;
    z-index: 5;
}
```

```
<body>
  <div id="div1">
        <div id="in"></div>
        <p>bla ...bla</p>
  </div>
  <div id="div2"></div>
  <div id="div3"></div>
</body>
```

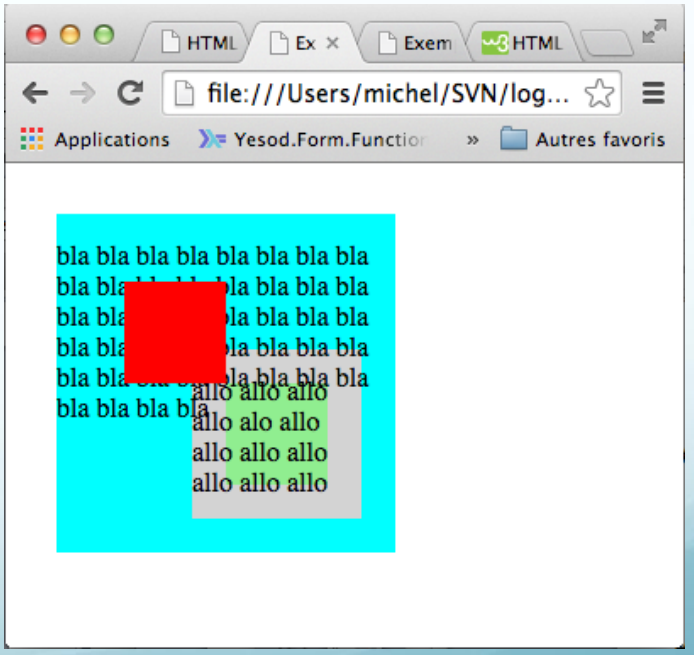


Propriété **z-index**

```
#div1 {
    position: fixed;
    left: 30px;
    top: 30px;
    background: aqua;
    width: 200px;
    height: 200px;
}
#div2 {
    position: absolute;
    left: 80px;
    top: 80px;
    background: lightGray;
    width: 100px;
    height: 100px;
    z-index: -1;
}
#in1 {
    position: absolute;
    top: 40px;
    left: 40px;
    width: 60px;
    height: 60px;
    background: red;
}
```

```
#in2 {
    position: absolute;
    top: 20px;
    left: 20px;
    width: 60px;
    height: 60px;
    background: lightGreen;
    z-index: -1;
}
```

```
<body>
<div id="div1">
  <div id="in1"></div>
  <p> bla ... bla</p>
   <div id="div2">
       <div id="in2"></div>
       <p>allo ... allo<p>
   </div>
</div>
</body>
```



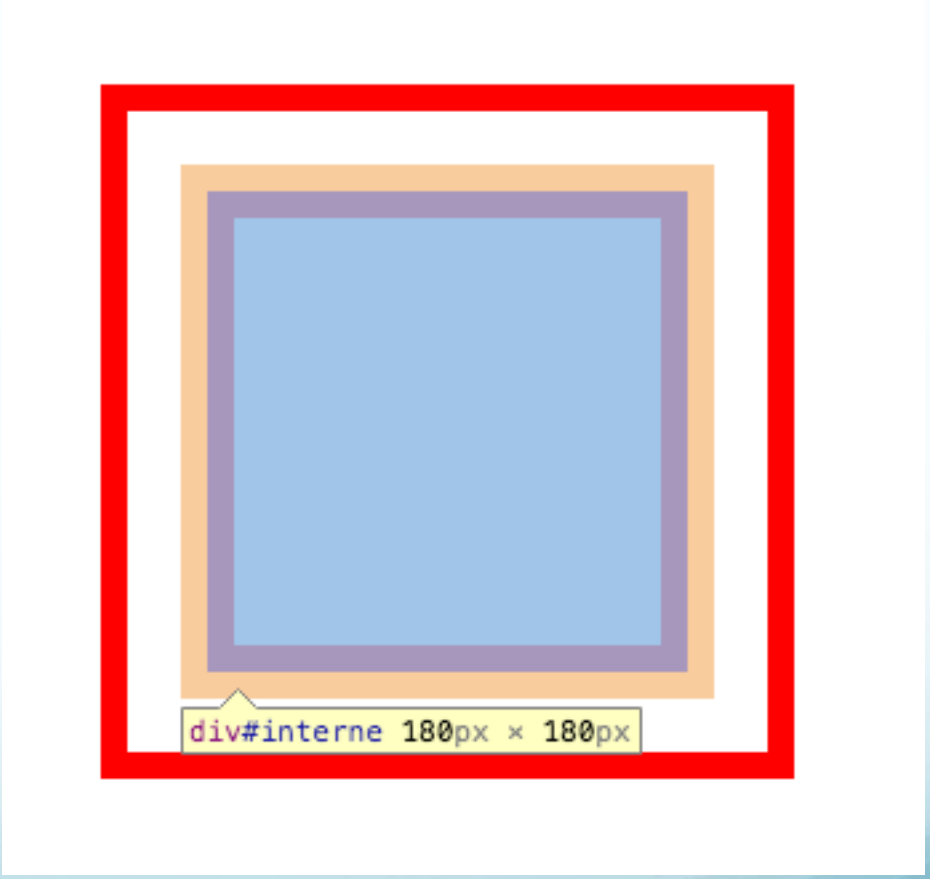
Largeur de la boîte

- Pour un positionnement fixe, elle est toujours spécifiée
- Pour un positionnement statique ou relatif, on peut utiliser la valeur **auto**, pour laisser le navigateur calculer automatiquement la largeur
- Pour un bloc positionné relativement ou statiquement, la largeur calculée sera celle de la boîte englobante, moins les marges et les padding horizontaux, les bordures et la barre de défilement s'il y a lieu
- Pour un bloc flottant ou positionné de manière absolue, la largeur sera celle du contenu
- Attention: il ne faut pas confondre la largeur de la boîte et la largeur du contenu (cette dernière est spécifiée par la propriété **width**)

Largeur de la boîte

```
#externe {
   position: fixed;
   top: 100px;
   left: 100px;
   width: 200px;
   height: 200px;
   border: solid 10px red;
   padding: 20px;
   margin: 10px;
}
#interne {
  margin: 10px;
  border: solid 10px blue;
  height: 160px;
}
```

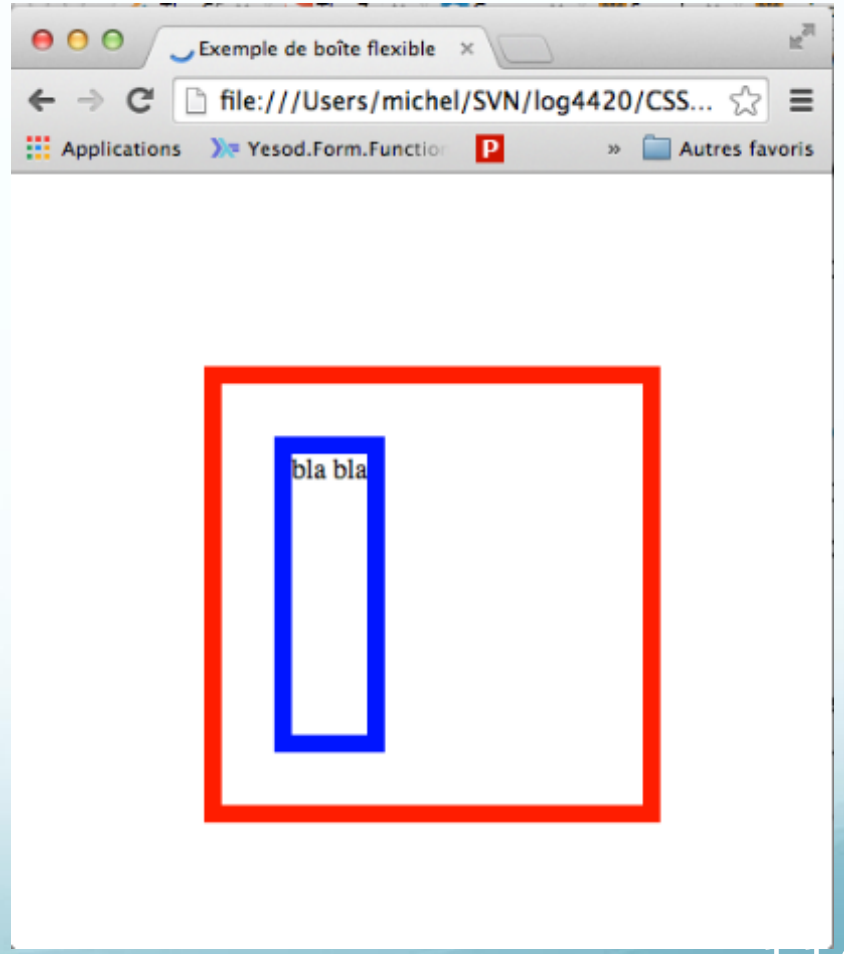
```
<body>
  <div id="externe">
    <div id="interne"></div>
  </div>
</body>
```



Largeur de la boîte

```
#externe {
   position: fixed;
   top: 100px;
   left: 100px;
   width: 200px;
   height: 200px;
   border: solid 10px red;
   padding: 20px;
   margin: 10px;
}
#interne {
  position: absolute;
  margin: 10px;
  border: solid 10px blue;
  height: 160px;
}
```

```
<body>
  <div id="externe">
    <div id="interne">bla bla</div>
  </div>
</body>
```



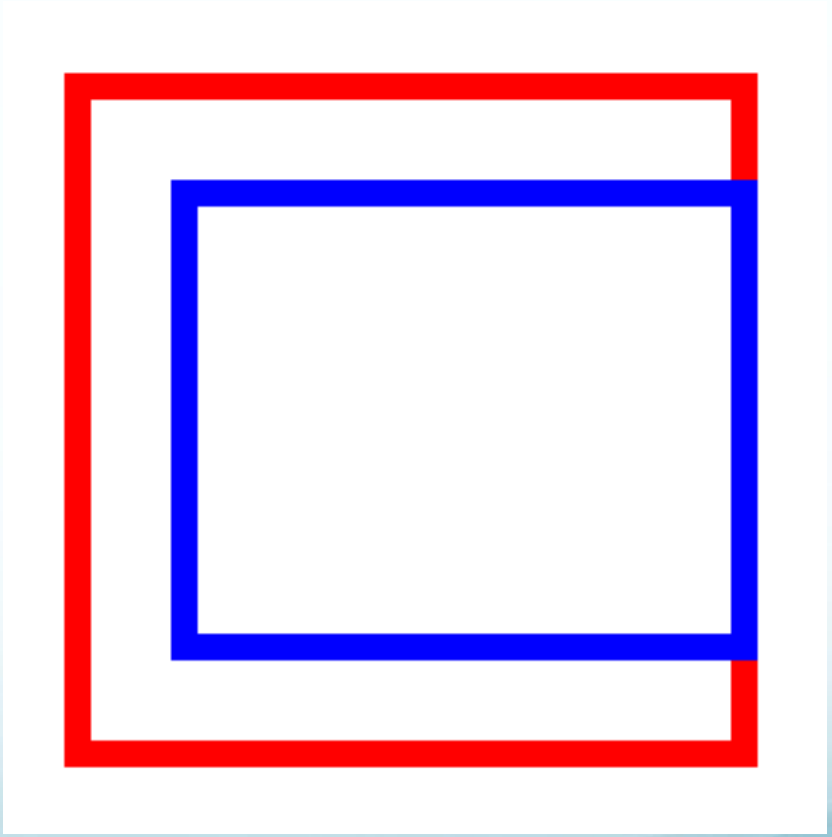
Remarque importante sur une largeur en pourcentage

- Pour une largeur indiquée en pourcentage, il est préférable de n'avoir ni padding, ni bordures, ni marges
- Par exemple, si on veut fixer une largeur égale à 100% de la boîte englobante, l'espace occupé débordera
- Si on veut absolument avoir ces items supplémentaires, voici la solution:
  - Ajouter une boîte dont la largeur sera fixée à 100% de la boîte englobante
  - Insérer la boîte qui nous intéresse dans cette nouvelle boîte, en utilisant une largeur fixée automatiquement

Largeur en pourcentage

```
#externe {
   position: fixed;
   top: 100px;
   left: 100px;
   width: 200px;
   height: 200px;
   border: solid 10px red;
   padding: 20px;
   margin: 10px;
}
#interne {
  margin: 10px;
  border: solid 10px blue;
  height: 160px;
  width: 100%;
}
```

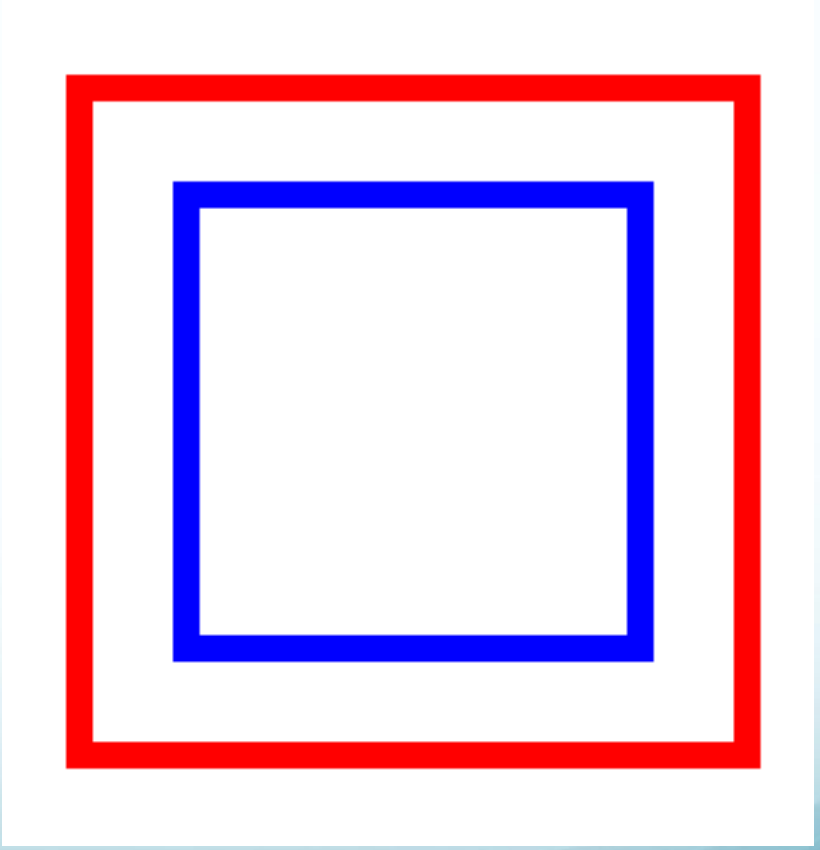
```
<body>
  <div id="externe">
    <div id="interne"></div>
  </div>
</body>
```



Largeur en pourcentage

```
#externe {
  ...
}
#inter {
  width: 100%;
}
#interne {
  margin: 10px;
  border: solid 10px blue;
  height: 160px;
}
```

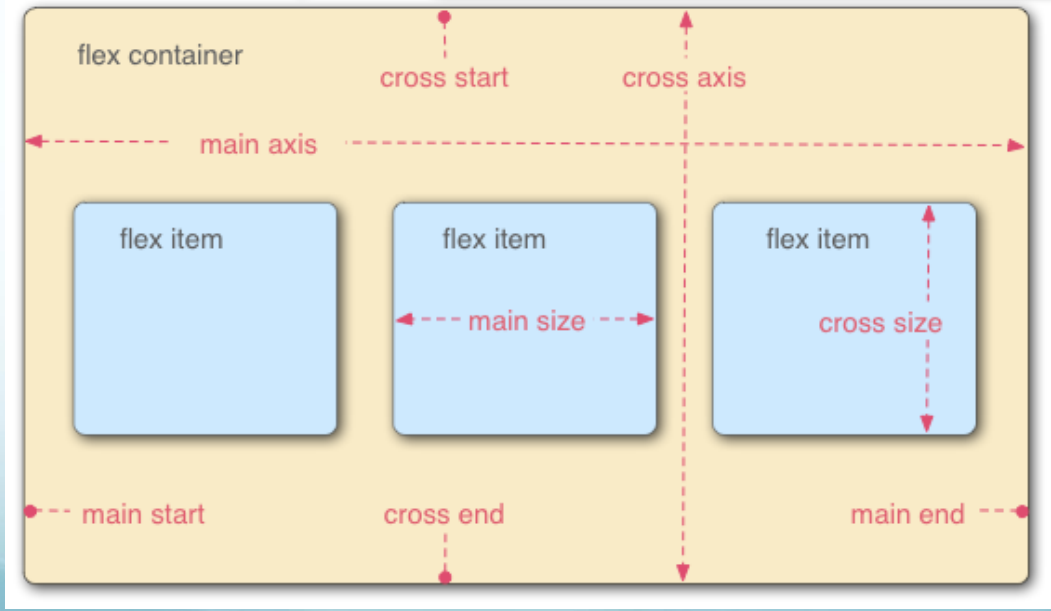
```
<body>
  <div id="externe">
    <div id="inter">
        <div id="interne"></div>
    </div>
  </div>
</body>
```



Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

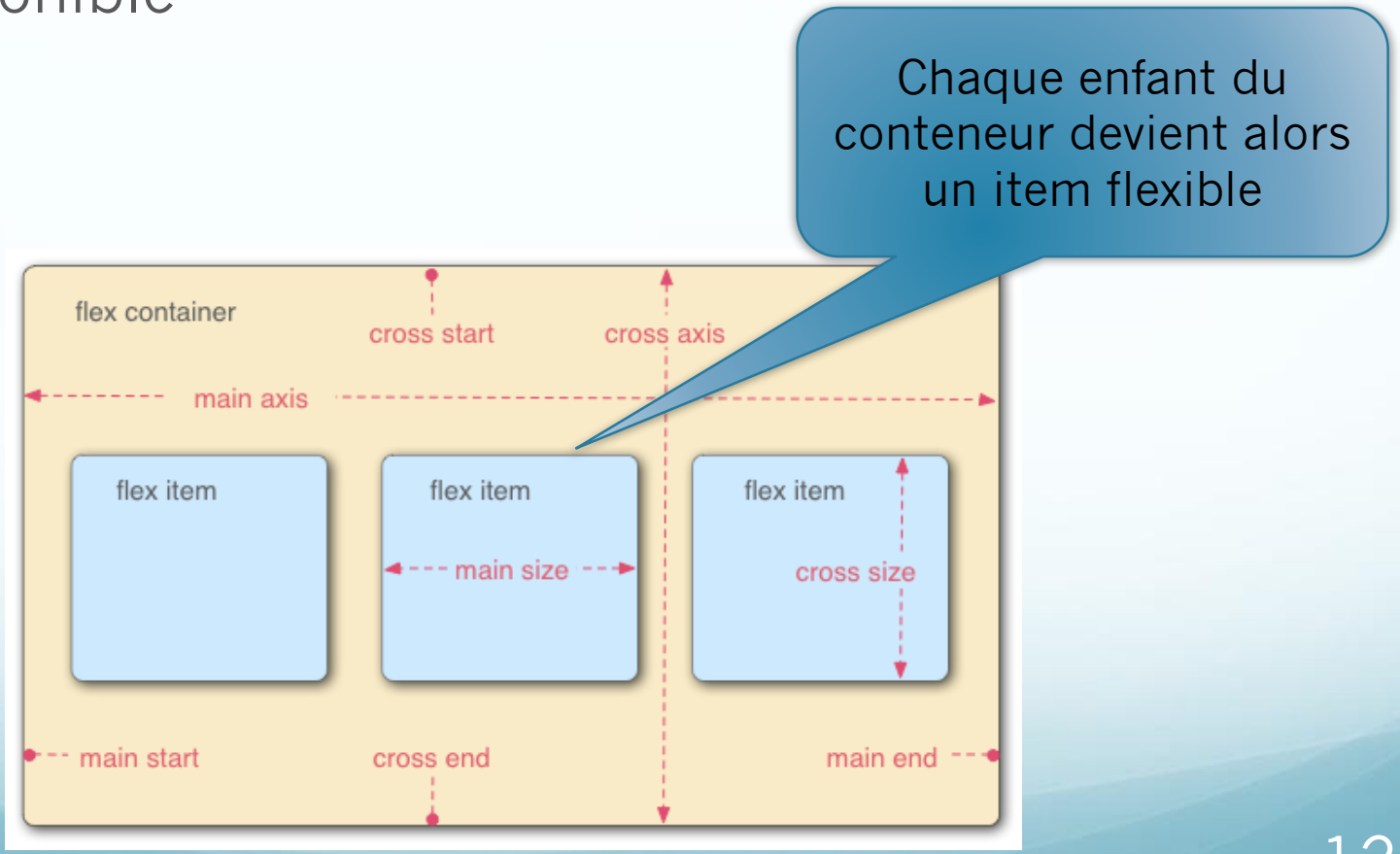
Un conteneur flexible est défini par la valeur **flex** à la propriété **display**



Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

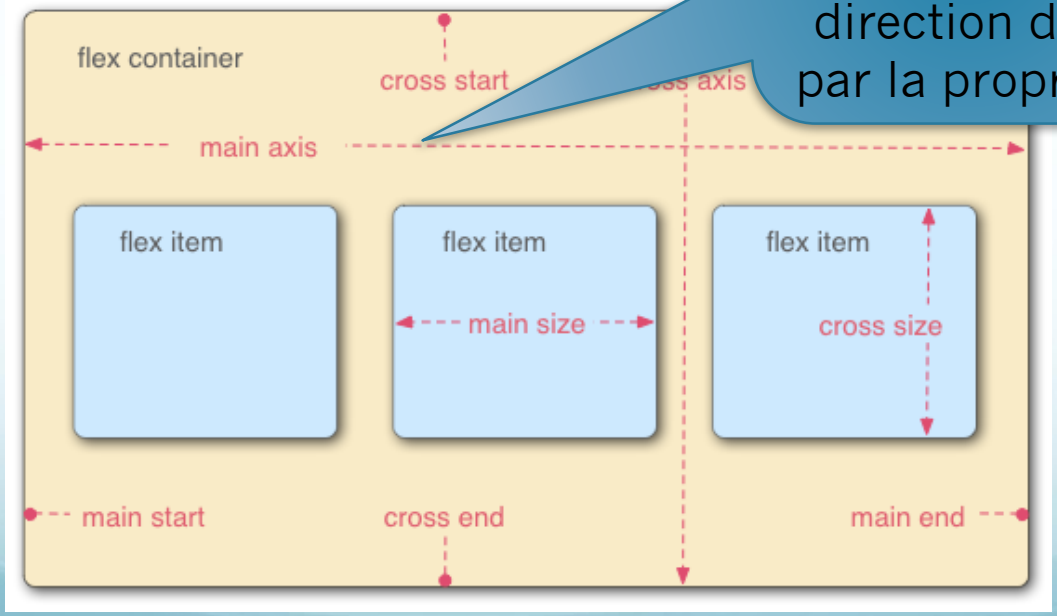


Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

Un axe principal détermine dans quelle direction les item flexibles seront placés les uns à la suite des autres. La direction de l'axe est établie par la propriété **flex-direction**

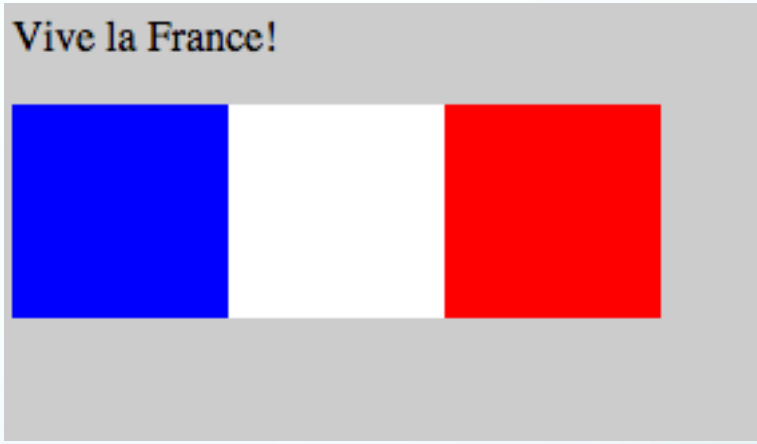


Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

Boîte flexible (CSS3)

```
body { background-color: #ccc; }
.flex {
   ...
   display: flex;
   flex-direction: row;
}

.flex > div {width: 100px; height: 100px}
.flex > div:nth-child(1){ color: white; background : blue; }
.flex > div:nth-child(2){ background : white; }
.flex > div:nth-child(3){ color: white; background : red; }
```

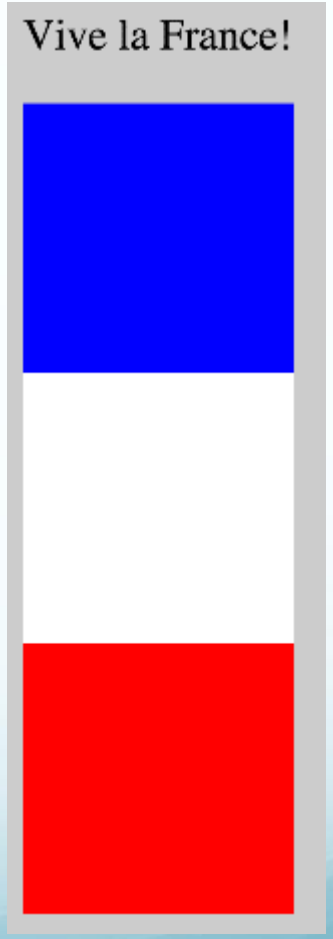
```
<body>
  <p>Vive la France!</p>
  <div class="flex">
          <div></div>
           <div></div>
           <div></div>
  </div>
</body>
```



Boîte flexible (CSS3)

```
body { background-color: #ccc; }
.flex {
   display: flex;
   flex-direction: column;
}

.flex > div {width: 100px; height: 100px}
.flex > div:nth-child(1){ color: white; background : blue; }
.flex > div:nth-child(2){ background : white; }
.flex > div:nth-child(3){ color: white; background : red; }
```

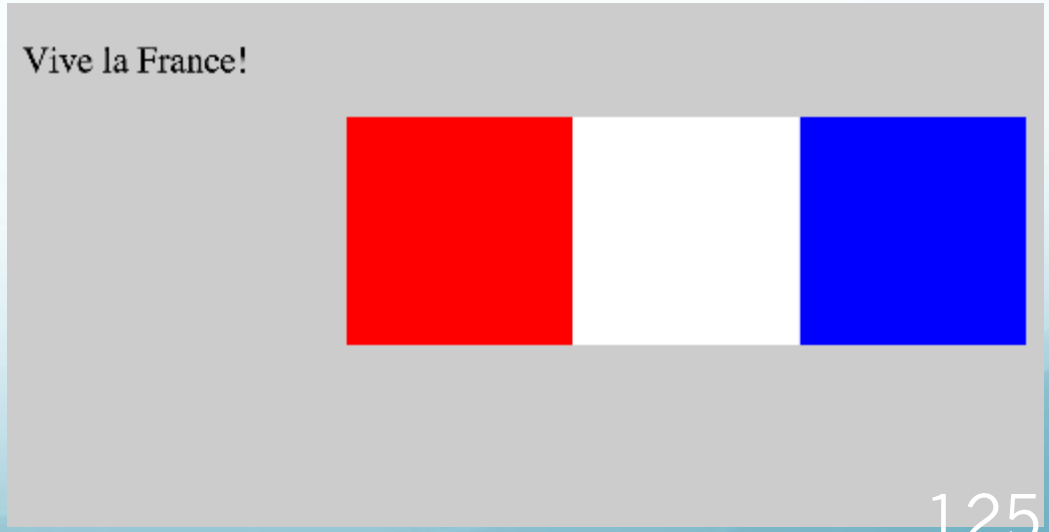
```
<body>
  <p>Vive la France!</p>
  <div class="flex">
          <div></div>
           <div></div>
           <div></div>
  </div>
</body>
```



Boîte flexible (CSS3)

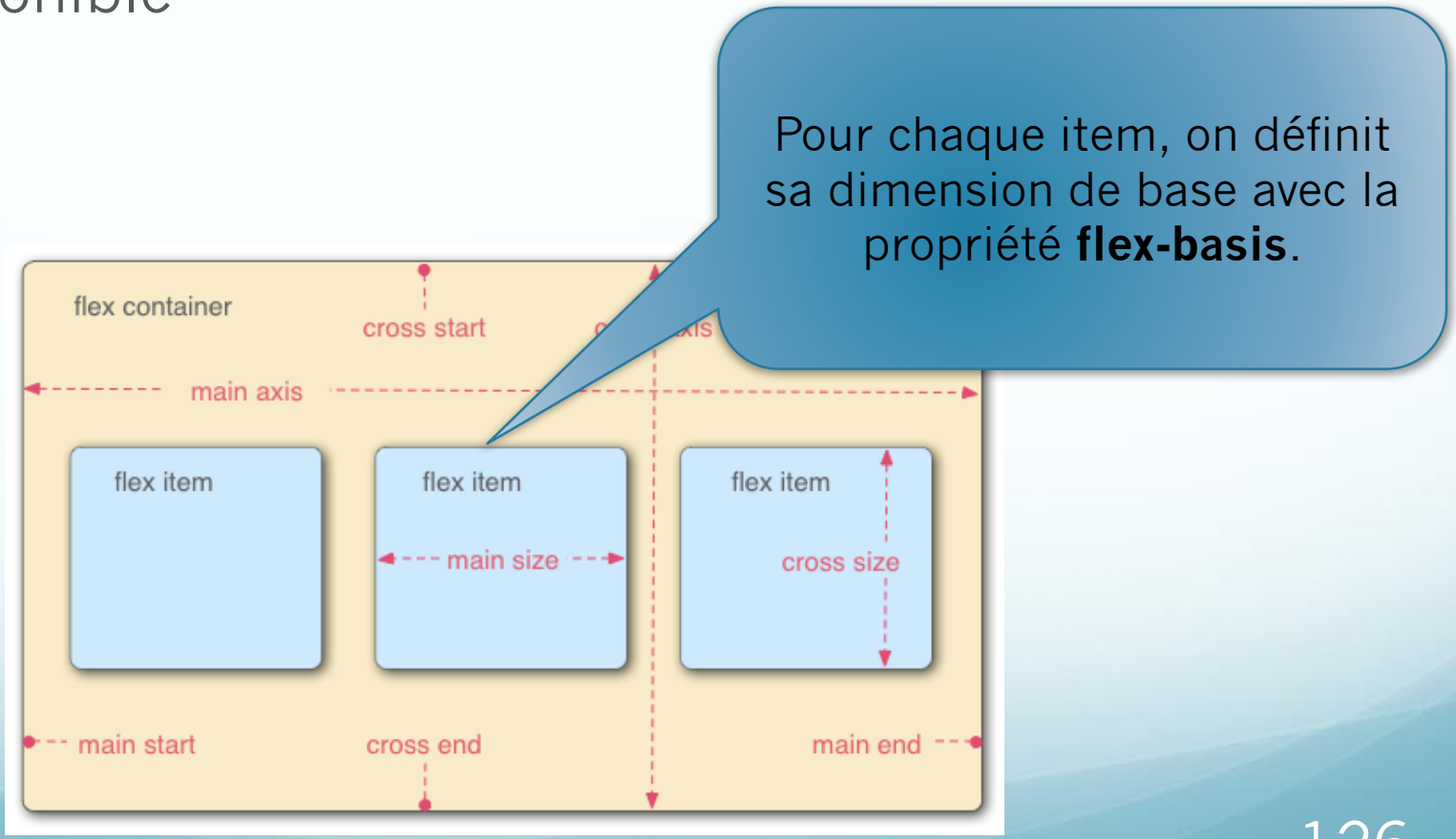
```
body { background-color: #ccc; }
.flex {
   ...
   display: flex;
   flex-direction: row-reverse;
}

.flex > div:nth-child(1){ color: white; background : blue; }
.flex > div:nth-child(2){ background : white; }
.flex > div:nth-child(3){ color: white; background : red; }
```

```
<body>
  <p>Vive la France!</p>
  <div class="flex">
      <div></div>
      <div></div>
      <div></div>
  </div>
</body>
```



Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

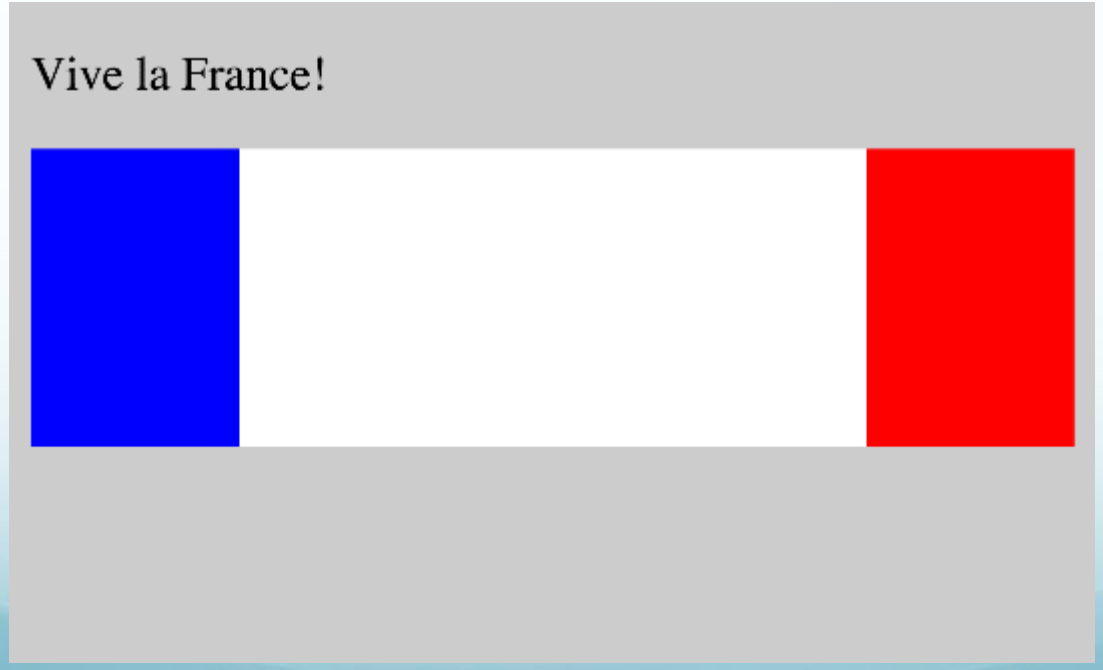


Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

Boîte flexible (CSS3)

....

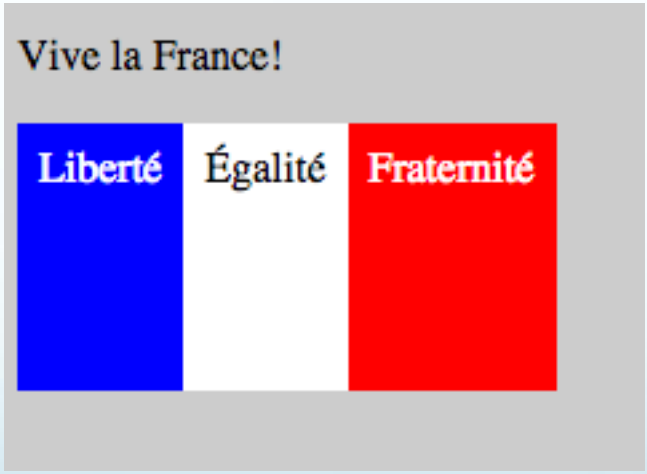
```
.flex > div:nth-child(1){flex-basis: 20%;}
.flex > div:nth-child(2){flex-basis: 60%;}
.flex > div:nth-child(3){flex-basis: 20%;}
```



Boîte flexible (CSS3)

```
….

. .flex > div {
        flex-basis: auto;
        padding: 0.5rem;
    }
```

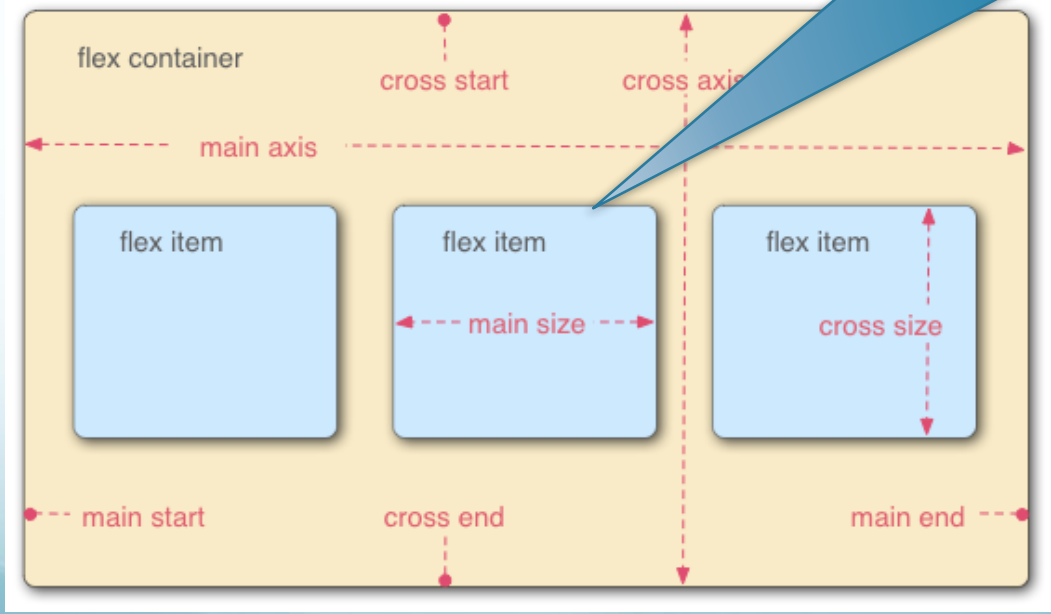
La valeur par défaut est auto: la boîte prend la taille de son contenu



Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

On peut changer l'ordre des items avec la propriété **order**.

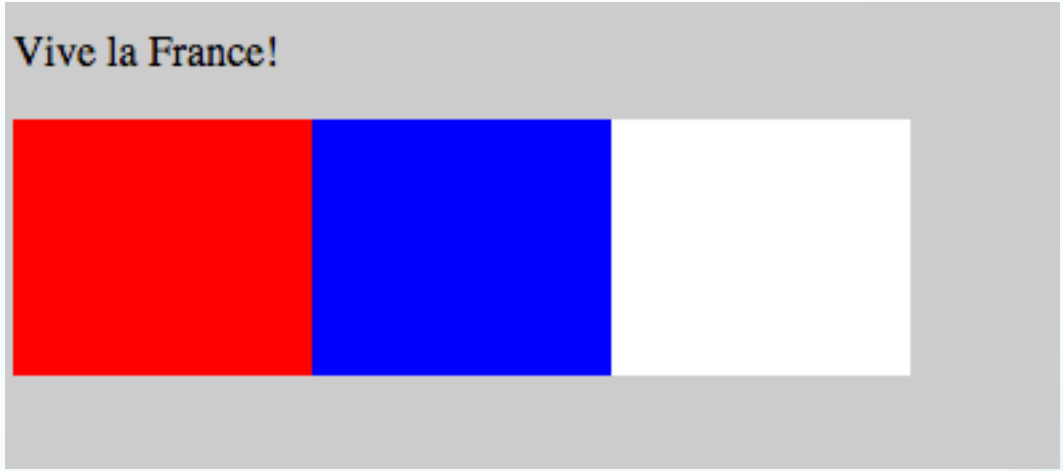


Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

Boîte flexible (CSS3)

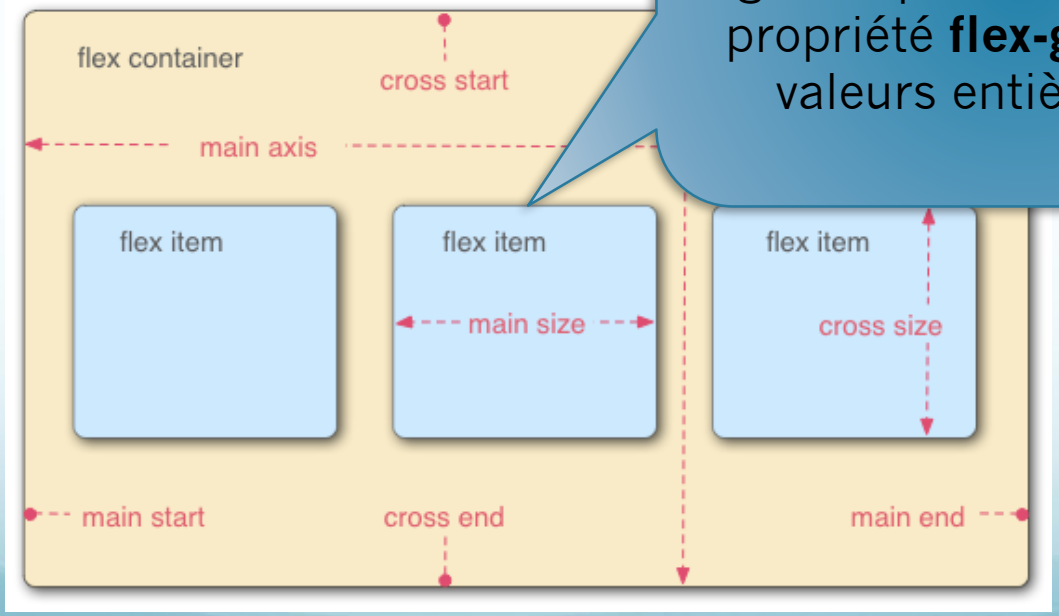
```
body { background-color: #ccc; }
....

.flex > div:nth-child(1){ order: 2;}
.flex > div:nth-child(2){ order: 3; }
.flex > div:nth-child(3){ order: 1; }
```

```
<body>
  <p>Vive la France!</p>
  <div class="flex">
      <div></div>
      <div></div>
      <div></div>
  </div>
</body>
```



Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

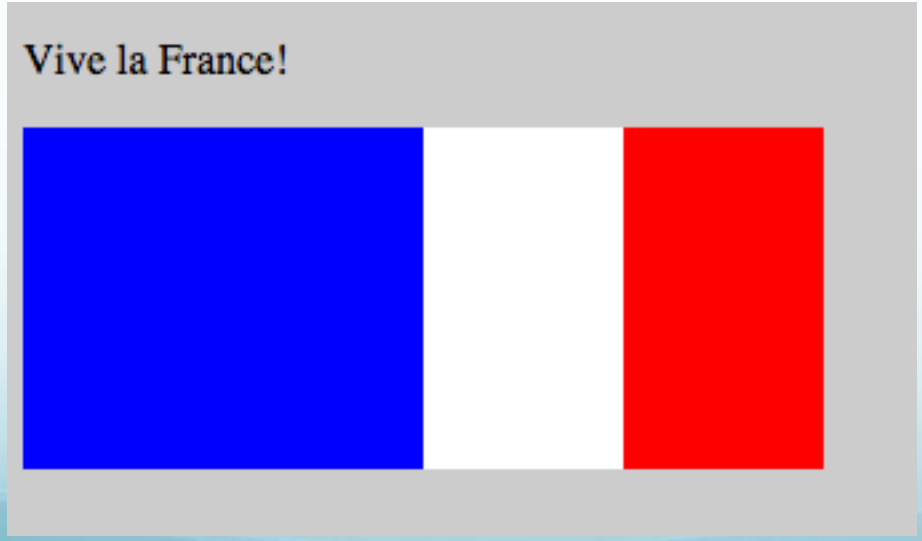


La proportion que chaque item occupe si le conteneur est agrandi peut être spécifiée par la propriété **flex-grow** . Il s'agit de valeurs entières sans unité.

Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

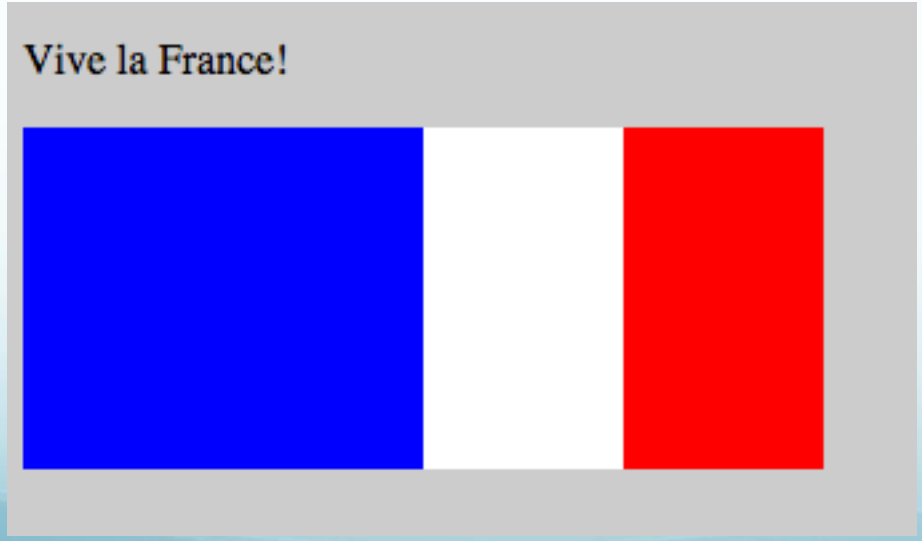
Boîte flexible (CSS3)

```
.flex {
   ...
   display: flex;
   flex-direction: row;
}

.flex > div:nth-child(1){ flex-grow: 2;}
.flex > div:nth-child(2){ flex-grow: 1;}
.flex > div:nth-child(3){ flex-grow: 1;}
```



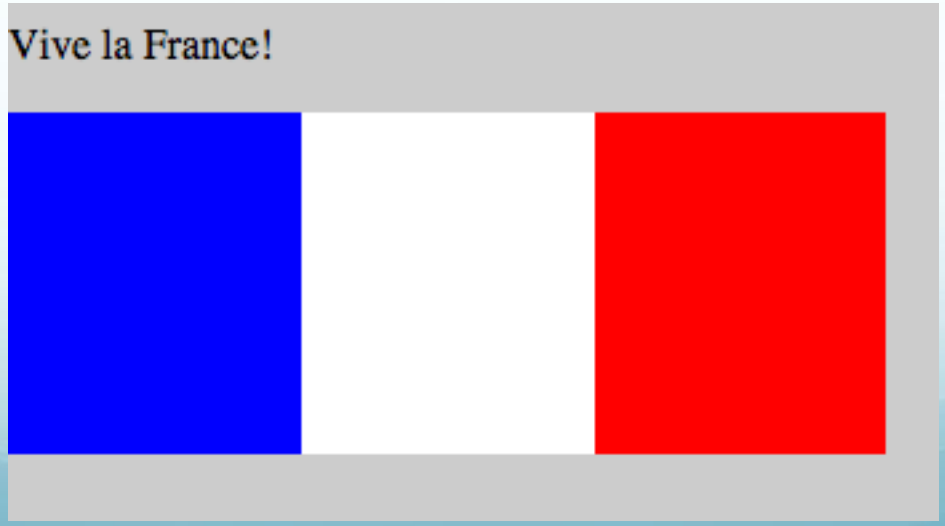
Boîte flexible (CSS3)

```
.flex {
   ...
   display: flex;
   flex-direction: row;
}

.flex > div:nth-child(1){ flex-grow: 2;}
.flex > div:nth-child(2){ flex-grow: 1;}
.flex > div:nth-child(3){ flex-grow: 1;}
```



Boîte flexible (CSS3)

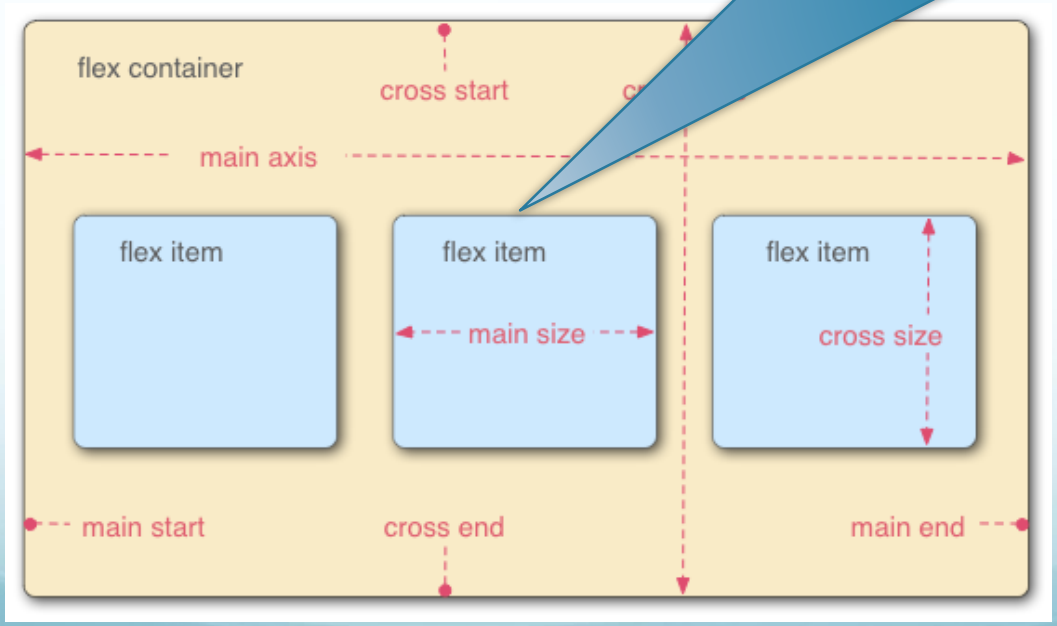
```
.flex {
   ...
   display: flex;
   flex-direction: row;
}

.flex > div:nth-child(1){ color: white; background : blue; }
.flex > div:nth-child(2){ background : white; }
.flex > div:nth-child(3){ color: white; background : red; }
.flex > div:hover {  flex-grow: 2 }
```



Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

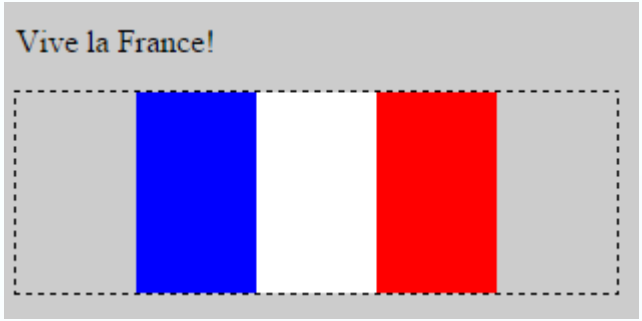
Avec la propriété **justify-content**, on peut déterminer l'alignement des items selon l'axe principal.



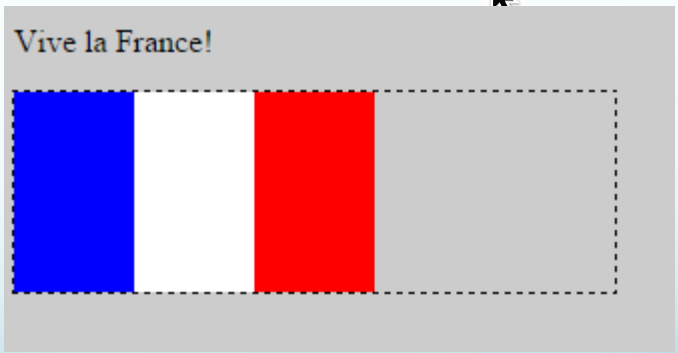
Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

Boîte flexible (CSS3)

```
.flex {
   ...
   display: flex;
   flex-direction: row;
   width: 300px;
   justify-content: center;
}
 .flex > div {
          flex-basis: 60px;
 }
```

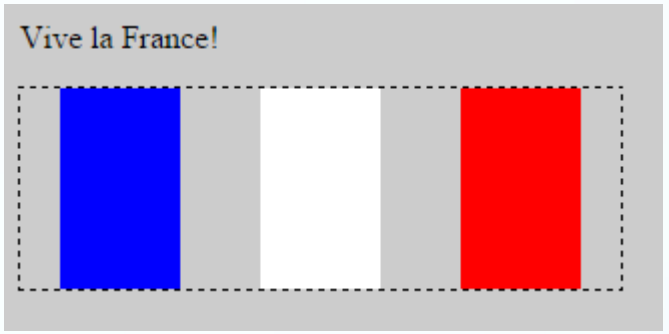



```
  .flex {
      ...
      display: flex;
      flex-direction: row;
      width: 300px;
      justify-content: flex-start;
  }
   .flex > div {
            flex-basis: 60px;
   }
```



Boîte flexible (CSS3)

```
.flex {
   ...
   display: flex;
   flex-direction: row;
   width: 300px;
   justify-content: space-around;
}
 .flex > div {
          flex-basis: 60px;
 }
```

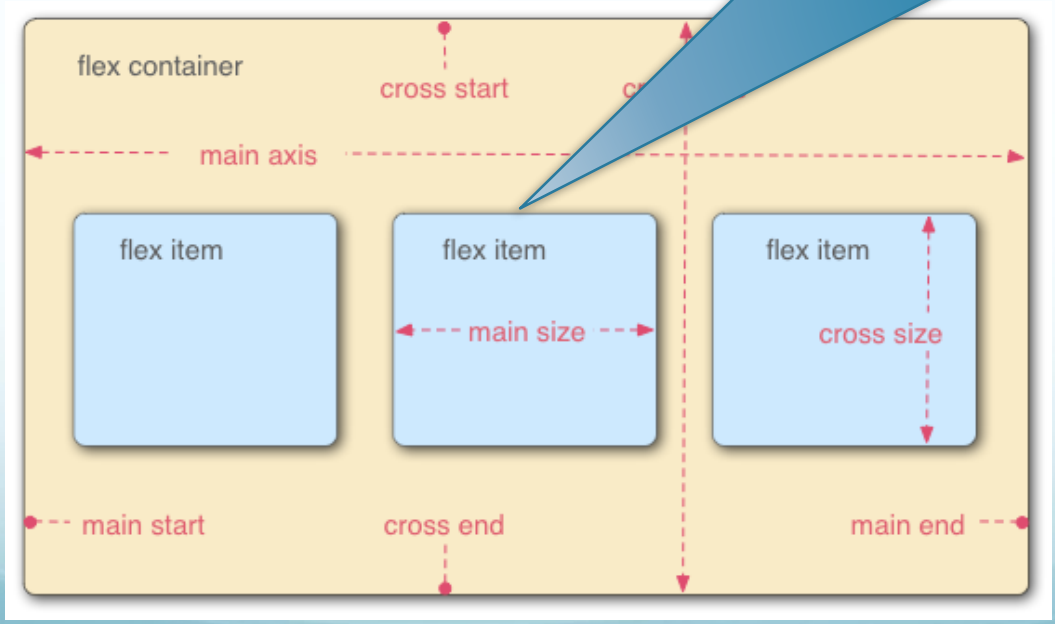



```
  .flex {
     ...
     display: flex;
     flex-direction: row;
     width: 300px;
     justify-content: space-between;
  }
   .flex > div {
            flex-basis: 60px;
   }
```

Boîte flexible (CSS3)

- Il s'agit d'une boîte qui s'adapte à l'espace disponible

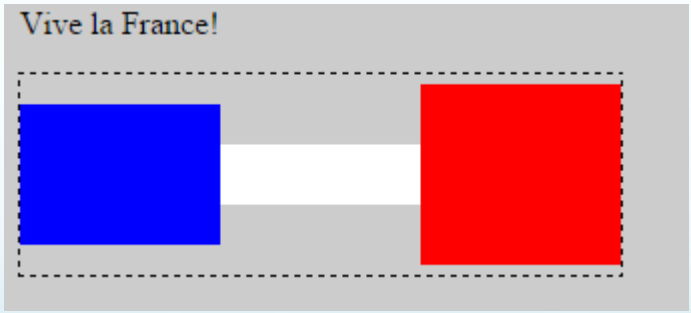
Avec la propriété **align-items**, on peut déterminer l'alignement des items selon l'axe perpendiculaire à l'axe principal (*cross axis*).



Tiré de https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

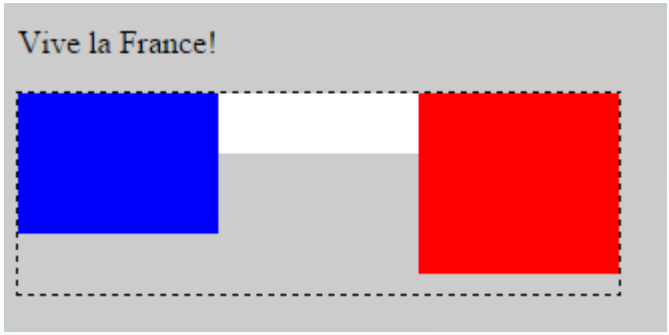
Boîte flexible (CSS3)

```
.flex {
   ...
   display: flex;
   flex-direction: row;
  align-items: center;
}
```

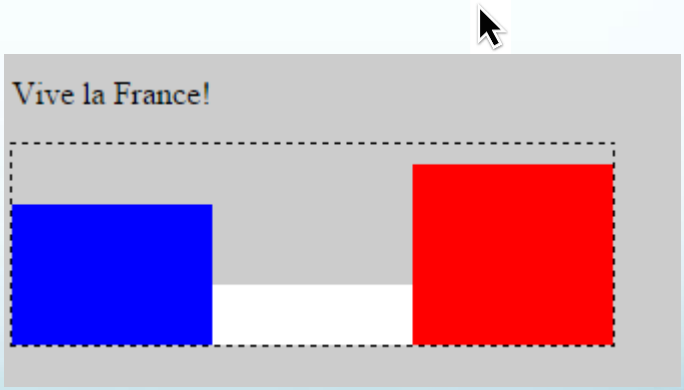


Boîte flexible (CSS3)

```
.flex {
   ...
   display: flex;
   flex-direction: row;
   align-items: flex-start;
}
```




```
.flex {
   ...
   display: flex;
   flex-direction: row;
   align-items: flex-end;
}
```



Grille (CSS3)

- Mise en forme du contenu sous forme de grille
- `display: grid`
- On définit ensuite le nombre et la tailles des colonnes et rangées
- Pour chaque bloc positionné, on spécifie la rangée et la colonne de départ, ainsi que le nombre de colonnes et rangées couvertes par le bloc

Grille (CSS3)

- Ce cours se concentre sur les boîtes flexibles.
- Si vous voulez en apprendre plus sur les Grilles, voici quelques ressources intéressantes :
  - Grid By Example : recueil de ressources maintenu par Rachel Andrew, experte dans le domaine de CSS
  - A Complete Guide to Grid : un autre tutoriel très intéressant sur le sujet

Propriété **overflow**

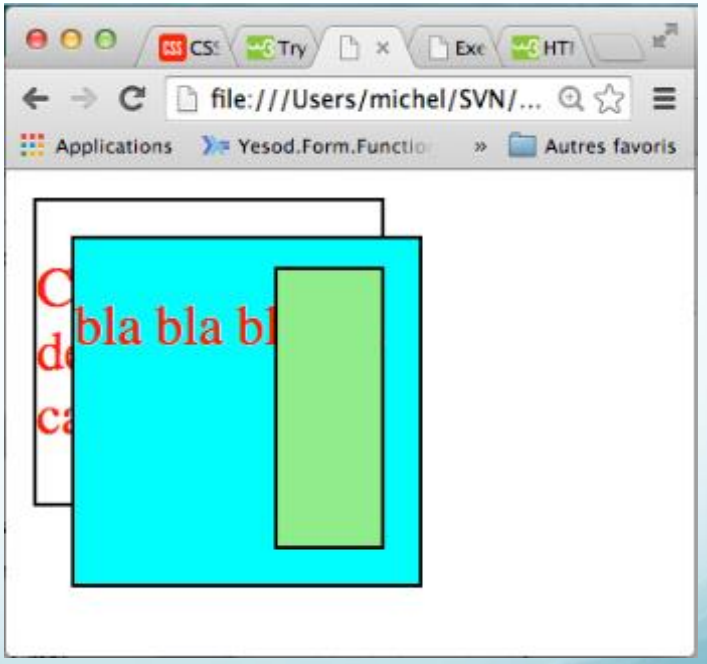
- Indique ce qui doit être fait avec le contenu d'une boîte lorsque celle-ci dépasse la largeur de la boîte englobante
- Valeurs possibles:
  - **visible** (valeur par défaut): le contenu est affiché en dépassant la largeur de la boîte englobante
  - **hidden** : le contenu qui dépasse sera caché
  - **scroll** : une barre de défilement sera ajoutée
  - **auto** : comportement non spécifié par CSS3

Propriété **inherit**

- Cette valeur peut être utilisée pour toutes les propriétés
- Elle indique que la valeur utilisée doit être celle de l'élément parent
- Évidemment, cela n'est pas très utile pour une propriété qui est déjà héritée selon les spécifications de CSS (**color** par exemple)

Propriété **inherit**

```
#div1 {
    position: relative;
    width: 100px;
    border: solid 1px;
    color: red;
}
#div2 {
    border: inherit;
    position: absolute;
    background-color: aqua;
    width: 100px;
    height: 100px;
    top: 10px;
    left: 10px;
}
#div3 {
    border: inherit;
    position: absolute;
    background-color:
        lightGreen;
    width: 30px;
    height: 80px;
    bottom: 10px;
    right: 10px;
}

<body>
    <div id="div1">
      <div id="div2">
        <p>bla bla bla bla</p>
        <div id="div3”></div>
      </div>
      <p> Ce texte devrait
            être caché </p>
    </div>
  </body>
</html>
```



*La propriété **color** est héritée par défaut*
La propriété border n’est pas héritée par défaut

Propriété opacity

- La valeur varie entre 0 (complètement transparent) à 1 (complètement opaque)
- S'applique à l'élément et ses descendants
- Cette propriété n'est pas héritée: si un des descendants se retrouve par dessus un autre, par défaut cet autre élément sera caché par ce descendant

Propriété opacity

```
#div1 {
    position: relative;
    width: 100px;
}
#div2 {
    position: absolute;
    background-color: aqua;
    opacity: 0.75;
   ...
}
#div3 {
    position: absolute;
    background-color: lightGreen;
    width: 30px;
    height: 80px;
    bottom: 10px;
    right: 10px;
}
```

```
<body>
    <div id="div1">
      <div id="div2">
        <p><br>bla bla bla bla</p>
        <div id="div3"></div>
      </div>
      <p> Ce texte ne devrait pas être caché </p>
    </div>
 </body>
```

Règles spéciales

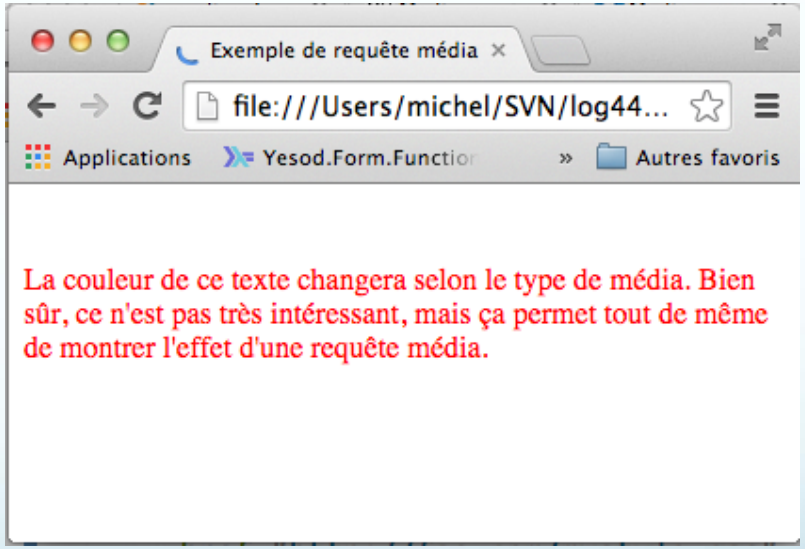
- **@charset** : pour déterminer le jeu de caractères
- **@import** : pour importer une feuille de style
- **@media** : pour définir le type de média
  - Valeurs possibles: all, braille, embossed, handheld, print, projection, screen, speech, tty, tv

Requêtes média

- Consiste en un type de média, et au moins une expression limitant la portée des déclarations CSS
- Permet d'adapter la présentation aux divers types d'appareils
- Utilisent les opérateurs logiques **not**, **and** et **only**
- Plusieurs requêtes peuvent être combinées dans une liste séparée par des virgules
- Exemple: **@media (min-width: 700px) and (orientation: landscape) {…}**
- Attributs pouvant être utilisés: **`color`**, **`height`**, **`width`**, **`orientation`**, **`resolution`**, etc.

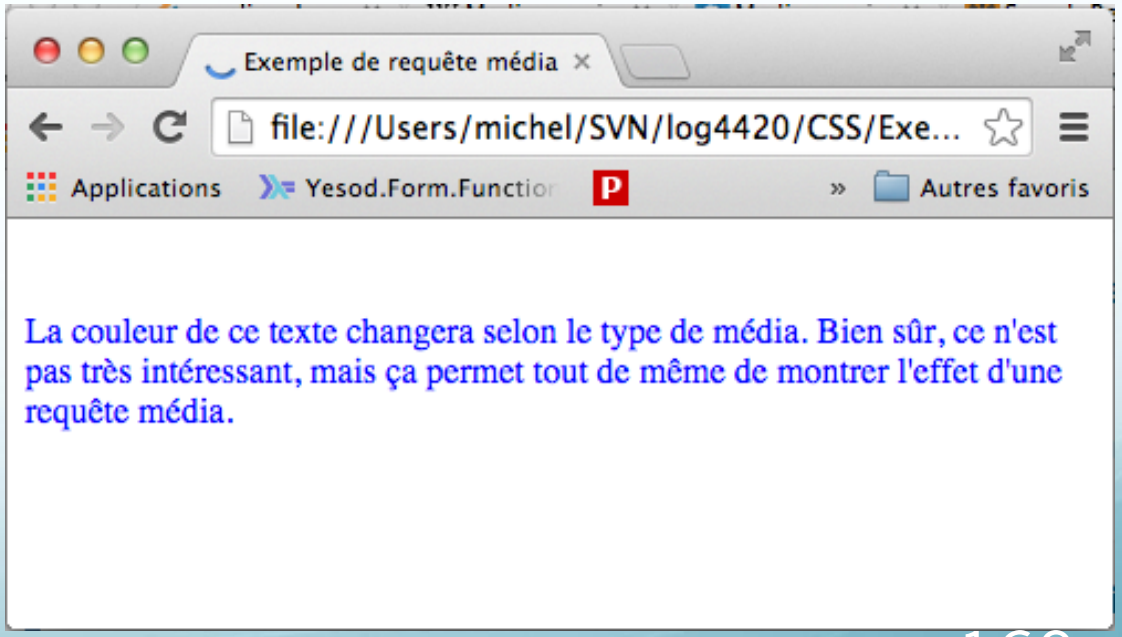
Requête média

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Exemple de requête média</title>
<style>
@media screen and (max-width: 500px) {

p {color: red;}

}

@media screen and (min-width: 500px) {

p {color: blue;}

}
</style>
</head>
<body>
  <p> La couleur ...média.
</body>
</html>
```



Requête média

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Exemple de requête média</title>
<style>
@media (max-width: 500px) {

p {color: red;}

}

@media (min-width: 500px) {

p {color: blue;}

}
</style>
</head>
<body>
  <p> La couleur ...média.
</body>
</html>
```



Requête média

Les boîtes flexibles peuvent être utilisées aussi pour avoir une disposition flexible de la mise en page du site. Pour un écran d'ordinateur de table:

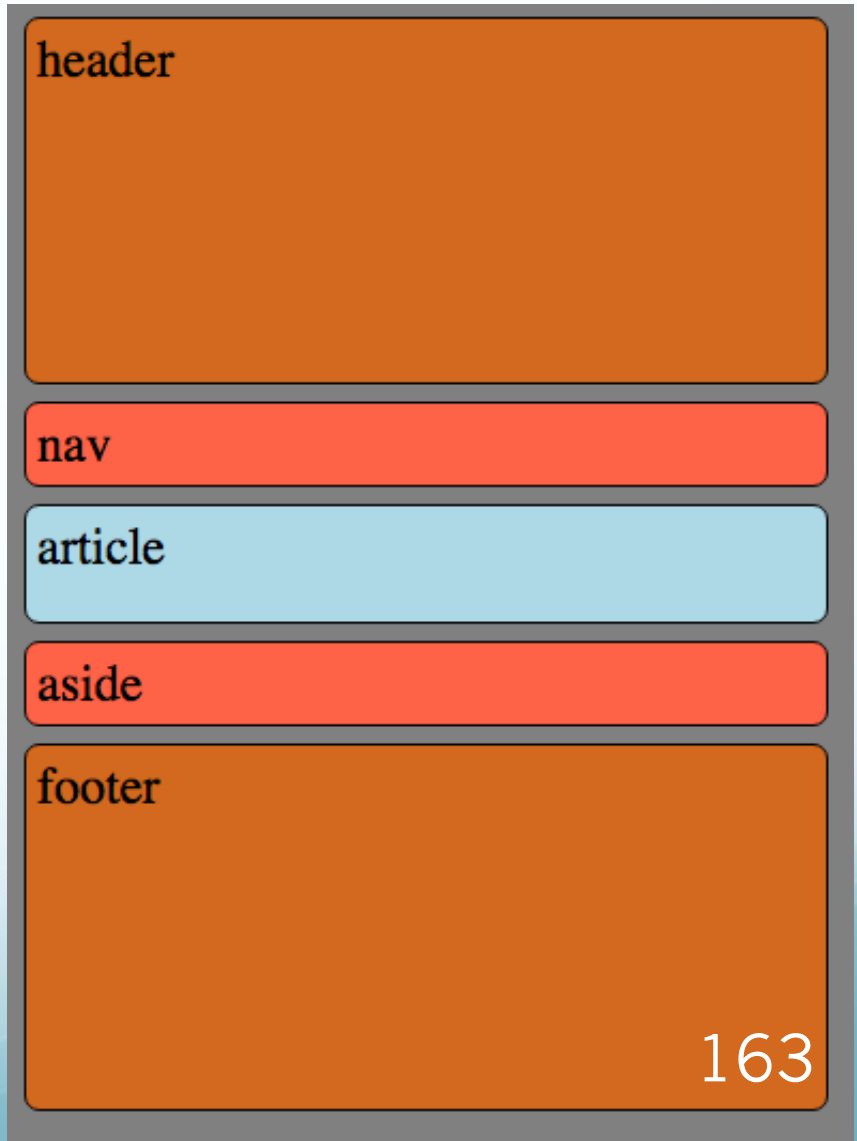
Requête média

Les boîtes flexibles peuvent être utilisées aussi pour avoir une disposition flexible de la mise en page du site. Pour un écran d'ordinateur de table:

```
/* Suffisamment large pour afficher  3 colonnes */
 @media screen  and (min-width: 640px) {

  #main {
       flex-direction: row;
  }
  #main > article {
      flex-basis: 60%;
  }
  #main > nav,
  #main > aside {
       flex-basis: 20%;
     }
}
```

Requête média

Les boîtes flexibles peuvent être utilisées aussi pour avoir une disposition flexible de la mise en page du site. Pour un écran de téléphone:



Requête média

Les boîtes flexibles peuvent être utilisées aussi pour avoir une disposition flexible de la mise en page du site. Pour un écran de téléphone:

```
/* Trop étroit pour afficher 3 colonnes */
 @media screen  and (max-width: 640px) {

  #main {
          flex-direction: column;
  }
  #main > article {
   height: 50rem;
  }
  nav,  aside, header, footer {
   height: 10rem;
  }
 }
```