

# Partie 2 : CSS 3





# Plan de la présentation

## Plan

Principe de base

Où écrit on le css?

Formatage du texte

La couleur du fond

Les bordures et les ombres

Les sélecteurs

Notion de class et id

Pseudo class et pseudo élément

Les modèle de boîtes

Flux et positionnement





# Principe de CSS

## Principe de CSS

- Le CSS est un langage permettant la mise en page d'une page web.



Attention à ne pas confondre :

- **Le HTML** : dit le **contenu** de ce qu'il y a sur la page

*Je veux un titre,  
Je veux une image,  
Je veux un paragraphe qui dit ça, ça, ça...*

- **Le CSS** : dit **comment afficher** ce qu'il y a sur la page

*Je veux que les titres soient centrés,  
Je veux qu'une image soit encadrée en vert,  
Je veux que le fond soit rouge...*



# Principe de CSS

## Pages personnelles de David NOËL

- [IRCCyN](#)
- [CNRS](#)



- [Accueil](#)
- [Curriculum Vitae](#)
- [Recherche](#)
- [Enseignement](#)
- [Contacts](#)

### Rechercher

Sur le site de l'IRCCyN

#### Formation

2010-2013 Doctorat au laboratoire de l'IRCCyN de Centrale Nantes : Identification du comportement dynamique de broches UGV et optimisation de leurs conditions d'exploitation  
2009-2010 Master II à l'Ecole Centrale de Nantes : [Conception de Systèmes et de Produits](#)

2008-2009 Préparation de l'Agrégation externe de Mécanique à l'Ecole Normale Supérieure de Cachan : admis le 25/06/09

2007-2008 Master I à l'Ecole Normale Supérieure de Cachan : [Mécanique et Ingénierie des Systèmes](#)

2006-2007 Licence III à l'Ecole Normale Supérieure de Cachan : [Science Mécanique et Technologie Industrielle](#)

#### Expérience professionnelle

2010 Stage de recherche à l'IRCCyN (6 mois) : Modélisation du comportement dynamique des roulements de broches UGV

2009-2010 Vacations à l'IUT de Nantes : encadrement de projets de conception en section SGM 2ième année

2010 Ingénieur Conception au sein de la start-up [MITIS](#) (travail à temps partiel)

2009 Dossier industriel : Analyse de défaillance de l'étanchéité du palier d'un moteur hydraulique pour [Podain Hydraulics](#) (dans le cadre du concours d'agrégation)

2008 Stage de Recherche à l'Université de Glasgow (4 mois) : Crack simulation with extended Finite Element Methods

*Page sans CSS*

## Pages personnelles de David NOËL

IRCCyN | CNRS



Accueil

Curriculum Vitae

Recherche

Enseignement

Contacts

Rechercher

Sur le site de l'IRCCyN

#### Formation

2010-2013 Doctorat au laboratoire de l'IRCCyN de Centrale Nantes : Identification du comportement dynamique de broches UGV et optimisation de leurs conditions d'exploitation

2009-2010 Master II à l'Ecole Centrale de Nantes : [Conception de Systèmes et de Produits](#)

2008-2009 Préparation de l'Agrégation externe de Mécanique à l'Ecole Normale Supérieure de Cachan : admis le 25/06/09

2007-2008 Master I à l'Ecole Normale Supérieure de Cachan : [Mécanique et Ingénierie des Systèmes](#)

2006-2007 Licence III à l'Ecole Normale Supérieure de Cachan : [Science Mécanique et Technologie Industrielle](#)

#### Expérience professionnelle

2010 Stage de recherche à l'IRCCyN (6 mois) : Modélisation du comportement dynamique des roulements de broches UGV

2009-2010 Vacations à l'IUT de Nantes : encadrement de projets de conception en section SGM 2ième année

2010 Ingénieur Conception au sein de la start-up [MITIS](#) (travail à temps partiel)

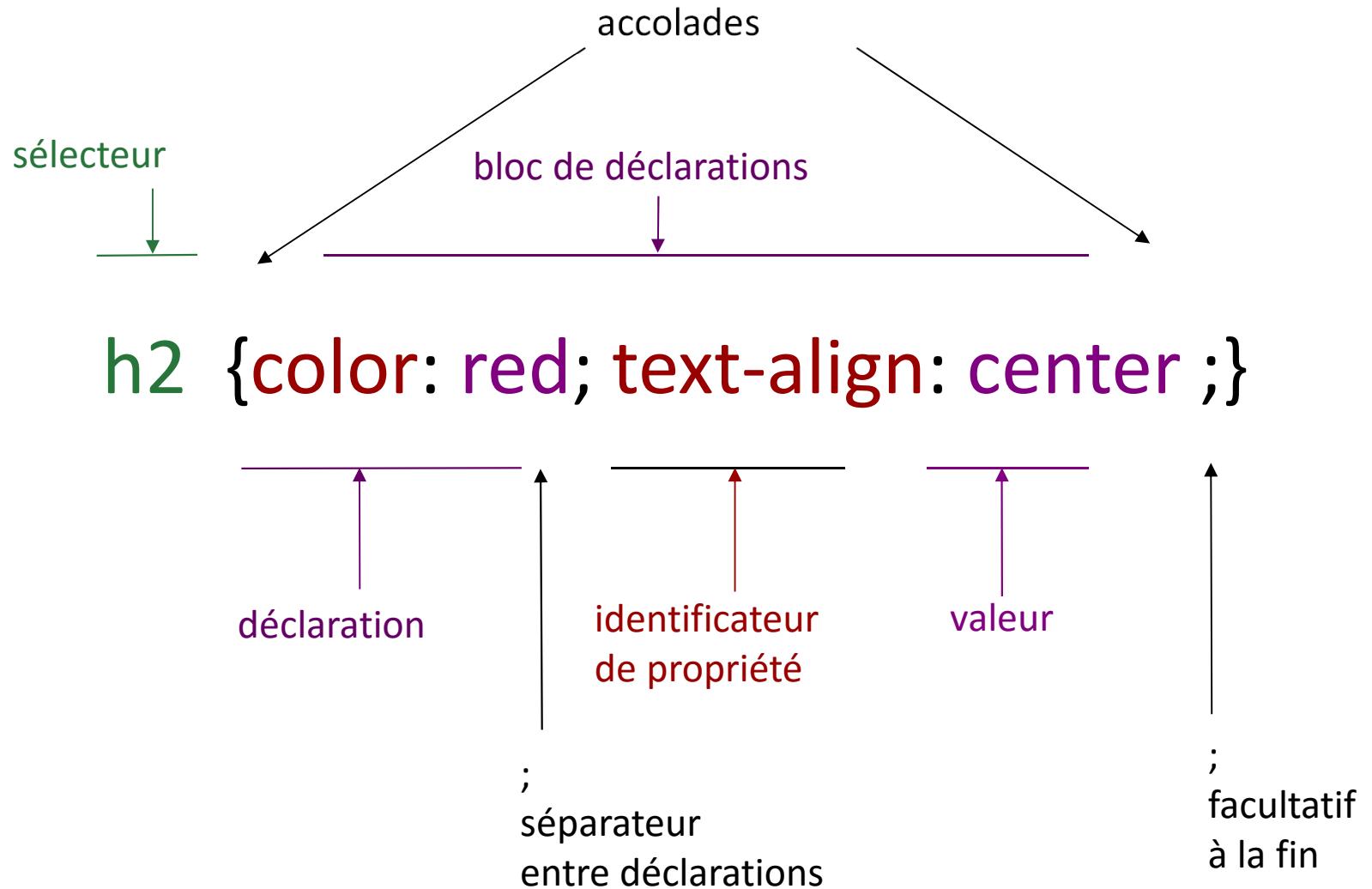
2009 Dossier industriel : Analyse de défaillance de l'étanchéité du palier d'un moteur hydraulique pour [Podain Hydraulics](#) (dans le cadre du concours d'agrégation)

2008 Stage de Recherche à l'Université de Glasgow (4 mois) : Crack simulation with extended Finite Element Methods

*Page avec CSS*



# Syntaxe





## Les sélecteurs

Les sélecteurs sont des caractères alphanumériques qui identifient la règle.

- Sélecteur HTML
- La classe
- L'ID



## Règle CSS : présentation du code

➤ Présentation possible :

```
h2 {color: red; text-align: center ; }
```

➤ Présentation conseillée (avec indentation et passage à la ligne) :

```
h2 {  
    color: red;  
    text-align: center ;  
}
```

Bonne  
Pratique



## Les commentaires

Comme en HTML, il est possible de mettre des **commentaires**.

Pour faire un commentaire, on tape :

`/*suivi du commentaire */`

Vos commentaires peuvent être sur une ou plusieurs lignes.

Par exemple :

`/* essai.css`

Par Flen

Fichier créé le jour j `* /`

```
p { color: blue; /* Les paragraphes seront bleus */  
}
```



## titre

On peut introduire un code en langage CSS de 3 façons :

- ✓ Dans un fichier CSS ;
- ✓ Dans l'en-tête du fichier HTML (<head>) ;
- ✓ Directement dans les balises via l'attribut style.



# Du CSS à travers un fichier CSS

- Pour qu'une page HTML puisse utiliser un fichier CSS il faut le référencer dans la partie `<head>` avec la balise `link`

```
<link rel="stylesheet" href="style.css" />
```



# Du CSS dans le <head> du fichier HTML

- On peut intégrer à l'aide de la balise head du code CSS sans passer par une page CSS

```
<head>
    <meta charset="utf-8" />
    <style>
        h1
        {
            color: red;
        }
    </style>
</head>
```



# Du CSS directement dans les balises

Où écrit on le CSS

- C'est la manière la moins recommandée

```
<html>
  <body>
    <p style="color:red;">Je ne suis pas recommandé</p>
  </body>
</html>
```



# Formatage du texte

- La taille
- La police
- Styles du texte
- L'alignement
- Les flottants



# La taille : font-size

## Balise { **font-size** : valeur }

- **taille absolue** : valeur en pixels, en centimètres ou millimètres.
  
- **taille relative** : valeur en pourcentage, « em » ou « ex »,

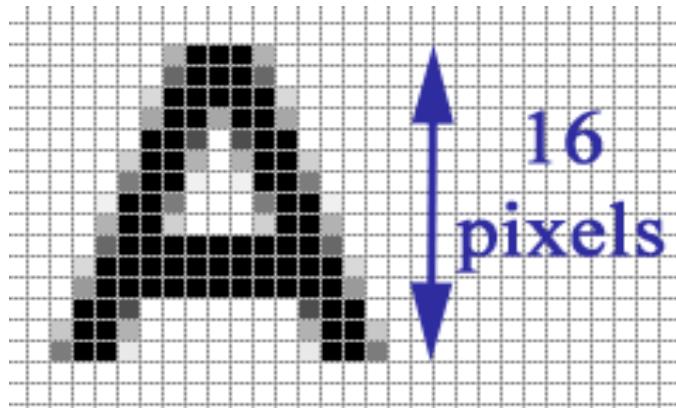


# Taille absolue

La taille

**Balise{ font-size : valeur px}**

Exemple : font-size: 16px;



Une lettre de 16 pixels de hauteur



# Une valeur relative

➤ C'est la méthode recommandée car le texte s'adapte alors plus facilement aux préférences de tous les visiteurs.

## valeur relative :

**xx-small** : minuscule ;

**x-small** : très petit ;

**small** : petit ;

**medium** : moyen ;

**large** : grand ;

**x-large** : très grand ;

**xx-large** : gigantesque.

Il existe que 7 tailles ?!!!



# Une valeur relative

il existe d'autres moyens :

- **em** :

1em, le texte a une taille normale.

`p { font-size: 0.8em; }`

- **ex** : qui fonctionne sur le même principe que le em mais qui est plus petit de base
- **le pourcentage** : (80%, 130%...).



# La police : font-family

## La police

### **balise { font-family: police; }**

➤ pour éviter les problèmes si l'internaute n'a pas la même police que vous, on précise en général plusieurs noms de police, séparés par des virgules :

### **balise { font-family: police1, police2, police3, police4; }**

➤ En général, on indique en tout dernier **serif** ou **sans-serif** (police par défaut)



# Liste des polices

- liste de polices qui fonctionnent bien sur la plupart des navigateurs

Arial ;

Arial Black ;

Comic Sans MS ;

Courier New ;

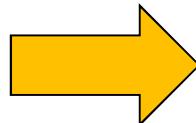
Georgia ;

Impact ;

Times New Roman ;

Trebuchet MS ;

Verdana.



*Texte en Arial*

**Texte en Arial Black**

*Texte en Comic Sans MS*

*Texte en Courier New*

*Texte en Georgia*

**Texte en Impact**

*Texte en Times New Roman*

*Texte en Trebuchet MS*

*Texte en Verdana*



## Exemple

La police

```
p { font-family: Impact, "Arial Black", Arial, Verdana, sans-serif; }
```

Entre " " car il y'a un  
espace

si rien n'a marché, mets une police  
standard **sans-serif**



# Italique, gras, souligné

## ➤ Mettre en italique

**font-style** : normal | oblique | italic

## ➤ Mettre en gras

**font-weight** : normal | bold

## ➤ Soulignement et autres décorations

**text-decoration** : underline | line-through |  
overline | none



# Alignement : à gauche, centré, à droite et justifié

## **text-align : left| center| right| justify**

➤ Valeur :

**left** : le texte sera aligné à gauche (par défaut).

**center** : le texte sera centré.

**right** : le texte sera aligné à droite.

**justify** : le texte sera « justifié ».

➤ Exemple :

```
h1 { text-align: center; }
```

```
p { text-align: justify; }
```



# Les flottants

- Le CSS nous permet de faire flotter un élément autour du texte.
- On dit aussi qu'on fait un « **habillage** ».

## Exemple :

une image flottante entourée par du texte :



  Lorem ipsum dolor sit amet,  
consectetuer adipiscing elit. Donec  
vitae lorem imperdiet lacus molestie  
molestie. Cum sociis natoque penatibus  
et magnis dis parturient montes, nascetur ridiculus  
mus. Donec eu purus. Phasellus metus lorem,  
blandit et, posuere quis, tincidunt vitae, ante.  
Vivamus consequat mauris a diam. Vivamus nibh  
erat, hendrerit nec, aliquet ut, hendrerit quis, nunc.  
Vestibulum et turpis et elit tempor euismod.



# Propriétés float et clear

➤ **Float** peut prendre deux valeurs très simples :

- left : l'élément flottera à gauche.
- right : l'élément flottera à droite

Exemple

```
img{  
    float : left  
}
```

➤ **clear**, une propriété qui permet de dire : « *Stop, ce texte doit être en-dessous du flottant et non plus à côté* ». Peut prendre trois valeurs :

- **left** : le texte se poursuit en-dessous après un **float: left**;
- **right** : le texte se poursuit en-dessous après un **float: right**;
- **both** : le texte se poursuit en-dessous, que ce soit après un **float: left**; ou après un **float:right**;



# La couleur et le fond

- La couleur du texte : *color*
- La couleur du fond : *background-color*
- Image du fond : *background-image*



# Les seize noms de couleurs utilisables en CSS

## La couleur et le fond

white	
silver	
gray	
black	
red	
maroon	
lime	
green	

yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	



# Color, background-color et background-image

- **Color** : définit la couleur du texte par exemple en hexadécimal

H3 {color: #000080}

- **background-color** : définit la couleur de l'arrière-plan

H1 {background-color: #000000}

- **background-image** : définit l'image de l'arrière-plan URL de l'image

Body {background-image: image.gif}



## Les bordures standards : border

➤ Border peut utiliser jusqu'à 3 valeurs pour modifier l'apparence de la bordure :

- **La largeur** : en pixels (comme 2px).
- **La couleur** : nom de couleur (black, red,...), soit une valeur hexadécimale (#FF0000), soit une valeur RGB (rgb(198, 212, 37)).
- **Le type de bordure** : (none | solid| dotted|double.....)

➤ Exemple :

```
h1 { border: 3px blue dashed; }
```



# Les différents types de bordure

`solid`

`dotted`

`dashed`

`double`

`groove`

`ridge`

`inset`

`outset`



# Exercice d'application

- Ecrire les propriétés CSS permettant de :
- ❖ Augmenter la taille des paragraphes à 150%
  - ❖ Encadrer les titres de niveau 1 avec un trait simple de couleur rouge
  - ❖ Souligner les titres de niveau 2 et changer la couleur du texte en vert. –



# border : en haut, à droite, à gauche, en bas

➤ Les quatre cotés :

**border-top** : bordure du haut ;

**border-bottom** : bordure du bas ;

**border-left** : bordure de gauche ;

**border-right** : bordure de droite.

➤ Exemple :

```
p { border-left: 2px solid black;  
    border-right: 2px solid black; }
```

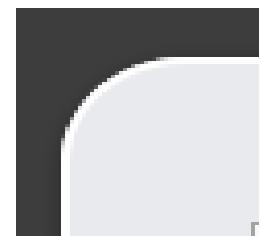
➤ On peut modifier les bordures d'une image



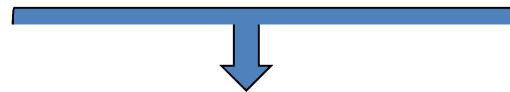
# Nouveauté de CSS3: Bordures arrondies

➤ **border-radius** : arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer la taille (« l'importance ») de l'arrondi en pixels :

```
p { border-radius: 10px; }
```



```
p { border-radius: 10px 5px 10px 5px; }
```



Les valeurs correspondent aux angles suivants dans cet ordre :en haut à gauche ;  
en haut à droite ;  
en bas à droite ;  
en bas à gauche;  
en haut à gauche.



# Nouveauté de CSS3 : Les ombres des boîtes

- La propriété **box-shadow** s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :
  - le décalage horizontal de l'ombre ;
  - le décalage vertical de l'ombre ;
  - l'adoucissement du dégradé ;
  - la couleur de l'ombre.
- Exemple

```
p { box-shadow: 6px 6px 0px black; }
```





# Nouveauté de CSS3 : L'ombre du texte

## ➤ **text-shadow** : l'ombre du texte

Avec **text-shadow**, vous pouvez ajouter une ombre directement sur les lettres de votre texte

Les valeurs fonctionnent exactement de la même façon que **box-shadow** : décalage, adoucissement et couleur.

## Code : CSS

```
p { text-shadow: 2px 2px 4px black; }
```

## Résultat :

file:///C:/Users/Mateo/Dropbox/Documents/Tutos/html-css/p2/ch

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.



# Les sélecteurs

## Les sélecteurs

Les principaux sélecteurs sont :

➤ Nom d'élément :

h1

➤ Sélecteur multiple (regroupement de sélecteurs) :

h1, h2, p

➤ Sélecteur contextuel (combinaison d'éléments imbriqués )

h1 a

➤ Identifiant :

p#intro1

#intro1

➤ Classe :

p.intro

.intro

➤ Pseudo-classe :

a:hover

➤ Sélecteur universel : \*



# Nom de l'élément

La règle css s'applique à toutes les balises HTML indiquées

Exemples :

```
h1 {  
    color: red;  
    font-size: 2em;  
}
```

```
h2 {  
    color: red;  
    font-size: 1.5em;  
}
```

```
p {  
    color: black;  
}
```



# Regroupement de sélecteurs du HTML (avec virgule)

```
h1 {color:red;  
    font-size:1,2em}  
  
h2 {color:red;}  
  
p  {color:red;}
```



```
h1, h2, p {color:red;}  
h1 {font-size:1,2em;}
```



# Sélecteur universel \*

\* remplace tout élément du document

Exemple :

\* {color: black}

Tous les éléments du document (h1, h2, p, div, ul...) auront une couleur de police noire.



# Les sélecteurs descendants

**A B : une balise contenue dans une autre**

Exemple : **h1 em**

h1 est un ancêtre de em, même lointain.

em est un descendant de h1, même lointain.

Dans la partie CSS

**h1 em {color: red;}**

Dans le body

```
<h1>Titre A1</h1>
<p>Texte A </p>
<h1>Titre B1 <em> Titre C1</em>Titre D1 </h1>
<p>Texte B <em> Texte C</em> Texte D </p>
```



# Les sélecteurs d'enfants

## Les sélecteurs

***A >B : une balise enfant direct de l'autre***

Exemple: **ol>li**

Li enfant direct de Ol

Dans la partie CSS

**ol>li {color :red}**

Dans le body

```
<ol>
  <li>titre1</li>
    <ul> <li>titre1.1</li>
      <li>titre1.2</li>
    </ul>
</ol>
```



# Sélecteur adjacent

***A + B : une première balise B qui suit A***

Exemple : **h3+p**

➤ Sélectionne la première balise **<p>** située après un titre **<h3>..**

Dans la partie CSS

**h3+p {color: red;}**

Dans le body

```
<h3>Titre</h3>
<p>Paragraphe1</p>
<p>Paragraphe2</p>
```



# Les sélecteurs d'attribut

**A[attribut] : une balise qui possède un attribut**

Exemple : **a[title]**

Sélectionne tous les liens <a> qui possèdent un attribut title.

Dans la partie CSS

**a[title]{color: red;}**

Dans le body

```
<a href="http://site1.com" title="Infobulle"> Lien1</a>
<br>
<a href="http://site2.com" > Lien2</a>
```



# Sélecteur de valeur d'attribut

***A[attribut="Valeur"] : une balise, un attribut et une valeur exacte***

Exemple : **a[title="Cliquez ici"]**

Idem, mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici »

Dans la partie CSS

**a[title="Cliquez ici"] {color: red;}**

Dans le body

**<a href="http://site.com" title="Infobulle " >Lien1</a>**

**<a href="http://site1.com" title="Cliquez ici">Lien2</a>**



# Sélecteur de valeur d'attribut

***A[attribut="Valeur"] : une balise, un attribut et une valeur***

Exemple : **a[title\*="ici"]**

Idem, l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position).

Dans la partie CSS

**a[title\*="ici"] {color: red;}**

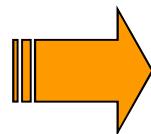
Dans le body

`<a href="http://site.com" title="Quelque part par ici">  
Lien</a>`



# L'identifiant id

- id est un attribut "générique" qui s'applique à toutes sortes d'éléments.
- Il sert à identifier une balise précise.
- Il doit être unique dans un document.



Elle s'ajoute dans une balise du document html :

```
<p id="intro">texte d'introduction</p>

```

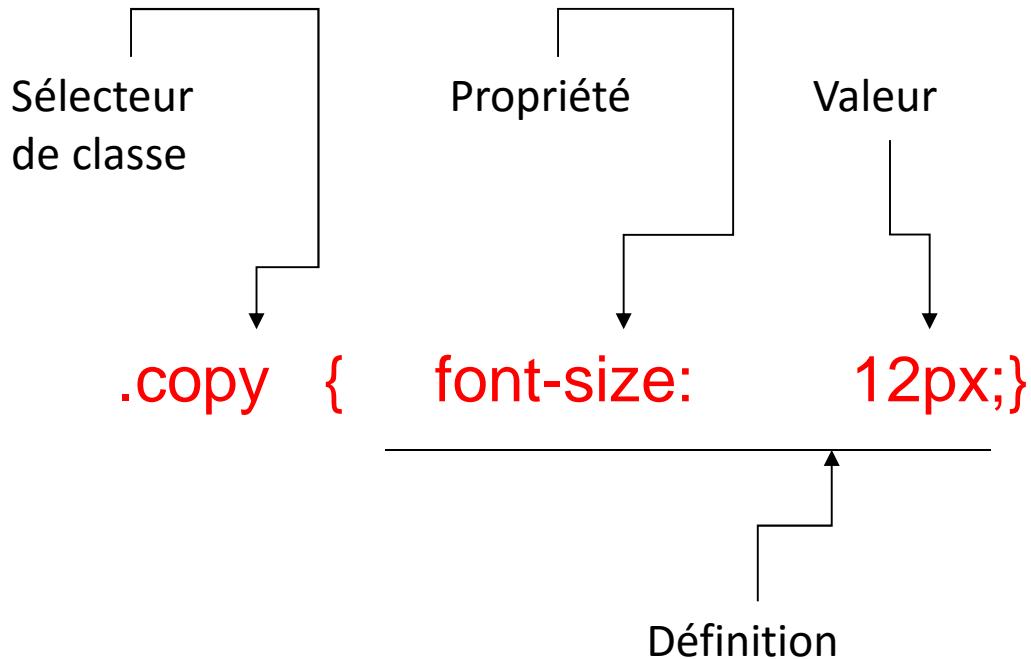
Utilisation dans une règle CSS :

```
#intro {font-style: italic; text-align: center;}
#logo {
    /* Mettez les propriétés CSS ici */
}
```



# Syntaxe générale d'une classe CSS

## Les sélecteurs

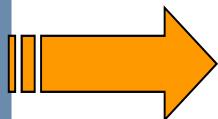


utilisation <p class="copy"> ... </p>



# La notion de class

- Class est un attribut "générique" qui s'applique à toutes sortes d'éléments.
- Une classe permet de définir un sous-ensemble.
- Elle peut s'appliquer à plusieurs éléments.



Elle s'ajoute dans une balise du document html :

```
<h1 class="intro" >Titre 1</p>
<p class="intro">Texte 2</p>
<p class="intro">Texte 3</p>
<img class="logo" src=image/img1.jpg>
```

Utilisation dans une règle CSS :

```
.intro {font-style: italic; color: brown}
h1.intro {text-align: center;}
.logo {border: 1px solid red ;}
```



# Les sélecteurs

## Les sélecteurs

.nav ul ul{color:red;}

```
<div class="nav">
  <ul>
    <li>nom
      <ul>
        <li>CHERIF</li>
        <li>Ben Saleh</li>
      </ul>
    </li>
    <li>prenom
      <ul>
        <li>Lina</li>
        <li>Mohamed</li>
      </ul>
    </li>
  </ul>
</div>
```



## Les sélecteurs

```
<BODY>
  <ul class="niveau1">
    <li>Exemple 1</li>
    <li>Liste simple</li>
  </ul>
  <div class="italic_rouge">
    Exemple 2
  </div>
</BODY>
```

.HTML

### Le Résultat

- Exemple 1
- Liste simple



#### Sous Class dépendante

.CSS

```
ul.niveau1
{
  list-style-type:square;
  color:blue;
}
```

#### Sous Class indépendante

.CSS

```
.italic_rouge
{
  color:red;
  font-style:italic;
}
```



## Les sélecteurs

# Bilan

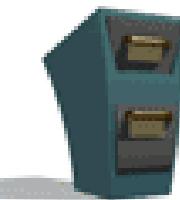
```
<BODY>
  <ul class="niveau1">
    <li>Exemple 1</li>
    <li>Liste simple</li>
  </ul>
  <div class="italic_rouge">
    Exemple 2
  </div>
</BODY>
```

.HTML

### Le Résultat

- Exemple 1
- Liste simple

**Exemple 2**



#### Sous Class dépendante

.CSS

```
ul.niveau1
{
list-style-type:square;
color:blue;
}
```

#### Sous Class indépendante

.CSS

```
.italic_rouge
{
  color:red;
  font-style:italic;
}
```



# DIV et SPAN

- **div** et **span** sont deux balises neutres.
- **div** et **span** sont en général associés à un id ou une class.
- **div** : balise de bloc
- **span** ("petite étendue") : balise en-ligne, délimite une partie de texte

```
<ul><li>.....</li>...</ul>
<div id="piedpage">
  <p>.....</p>
  <p>.....</p>
</div>
```

```
<p>Il prend un pot de <span
id="rouge">peinture rouge</span> et
un pinceau.</p>
```



# DIV : Exemple

## Les sélecteurs

```
<HTML>
  <HEAD>
    <STYLE >
      .zone{
        font-size: x-small;
      }
    </STYLE>
  </HEAD>
  <BODY>
    La balise DIV
    <DIV class=zone>
      <P>Commentaire :</P>
      <P>N'oubliez pas l'attribut class!</P>
    </DIV>
  </BODY>
</HTML>
```



# SPAN: Exemple

```
<!DOCTYPE Html >
<HTML>
  <HEAD>
    <TITLE>14 Janvier</TITLE>
    <STYLE>
      SPAN { float: left;
              font-family: Arial;
              font-size: 40pt }
    </STYLE>
  </HEAD>
  <BODY>
    <H2> Les Tunisiens fêtent le 14 Janvier sur fond de crise</H2>
    <SPAN>L</SPAN>e centre-ville de Tunis a été envahi par la foule,
    samedi, dès les premières heures de la journée. A chacun sa
    petite musique. .....
  </BODY>

</HTML>
```



# Les pseudo classes

➤ Règles affectant les liens :

**A:link { color: red }** /\*liens non visité\*/

**A:active { color: maroon }** /\*lien pendant que l'on click,dessus)

**A:visited { color: green }** /\*liens visités\*/

**A:hover {font-size:300%}** /\*Au survol\*/

**A:focus** /\*(état d'un élément qui a reçu «l'attention», par exemple un lien lorsqu'on y accède grâce au clavier, ou un champ texte d'un formulaire lorsqu'on clique dedans);\*/



# Les pseudo éléments

## Les sélecteurs

➤ Règles affectant le 1er caractère ou la 1ère ligne de chaque paragraphe :

`/*lettrine" taille 3x sup. au texte*/`

**P:first-letter**

```
{ font-size: 300% ;  
  float: left ;  
  color: green  
}
```

`/*1ère ligne en petites majuscules */`

**P:first-line { font-style: small-caps }**



## Les sélecteurs

# Exercice d'application

## Chapitre : CSS

- *partie 1 : Les sélecteurs*
  1. *Les selecteurs simples*
  2. *les selecteurs Combinés*
  3. *notion de class et id*
- *partie 2 : DIV et SPAN*
  1. La balise DIV
    - *Exemple*
  2. La balise Span
- *partie 3 : pseudo-classes*
  1. Règles affectant les liens
    - *Exemple*
  2. Règles affectant les paragraphes

**Donner le code HTML et CSS permettent d'obtenir ce résultat**



# Les styles du tableau

## Les styles du tableau

- Margin
- Padding
- Background-color , background-image
- Color,font-size, text-decoration,font-family....
- width, height
- Border
- Border-collapse
- Caption-side





## Les styles du tableau

# Ajouter les bordures : border

```
<table>
<tr> <th>Nom <th>Age<th>Pays
<tr>
  <td>Imen <td>33ans<td>Sfax
<tr>
  <td>Ahmed <td>26 ans<td>Mahdia
</table>
```



Nom	Age	Pays
Imen	33ans	Sfax
Ahmed	26 ans	Mahdia

```
table,td,tr {
  border: 1px solid black;
  /* auront une bordure de 1px */
}
```



# Coller les bordures : border-collapse

- Cette propriété peut prendre deux valeurs :
  - **collapse** : les bordures seront collées entre elles
  - **separate** : les bordures seront dissociées (valeur par défaut)

```
<table>
<tr> <th>Nom <th>Age<th>Pays
<tr>
  <td>Imen <td>33ans<td>Sfax
<tr>
  <td>Ahmed <td>26 ans<td>Mahdia
</table>
```

Nom	Age	Pays
Imen	33ans	Sfax
Ahmed	26 ans	Mahdia

```
th,td{
  border: 1px solid black;
}
table {
  border-collapse: collapse;
}
```



## Les styles du tableau

# Le style du titre du tableau: caption-side

- La propriété CSS **caption-side** peut prendre deux valeurs :
  - **top** : le titre sera placé au-dessus du tableau (par défaut)
  - **bottom** : le titre sera placé en dessous du tableau.

```
<table>
<caption> liste des enseignants</caption>
<tr> <th>Nom <th>Age<th>Pays
<tr>
  <td>Imen <td>33ans<td>Sfax
<tr>
  <td>Ahmed <td>26 ans<td>Mahdia
</table>
```

Nom	Age	Pays
Imen	33ans	Sfax
Ahmed	26 ans	Mahdia

liste des enseignants

```
caption{ caption-side :bottom;}
```



# Exercice d'application

## Les sélecteurs

- Ecrire les codes HTML et CSS permettant d'obtenir le résultat suivant:



**Titre de niveau 1**

**Titre de niveau 2**

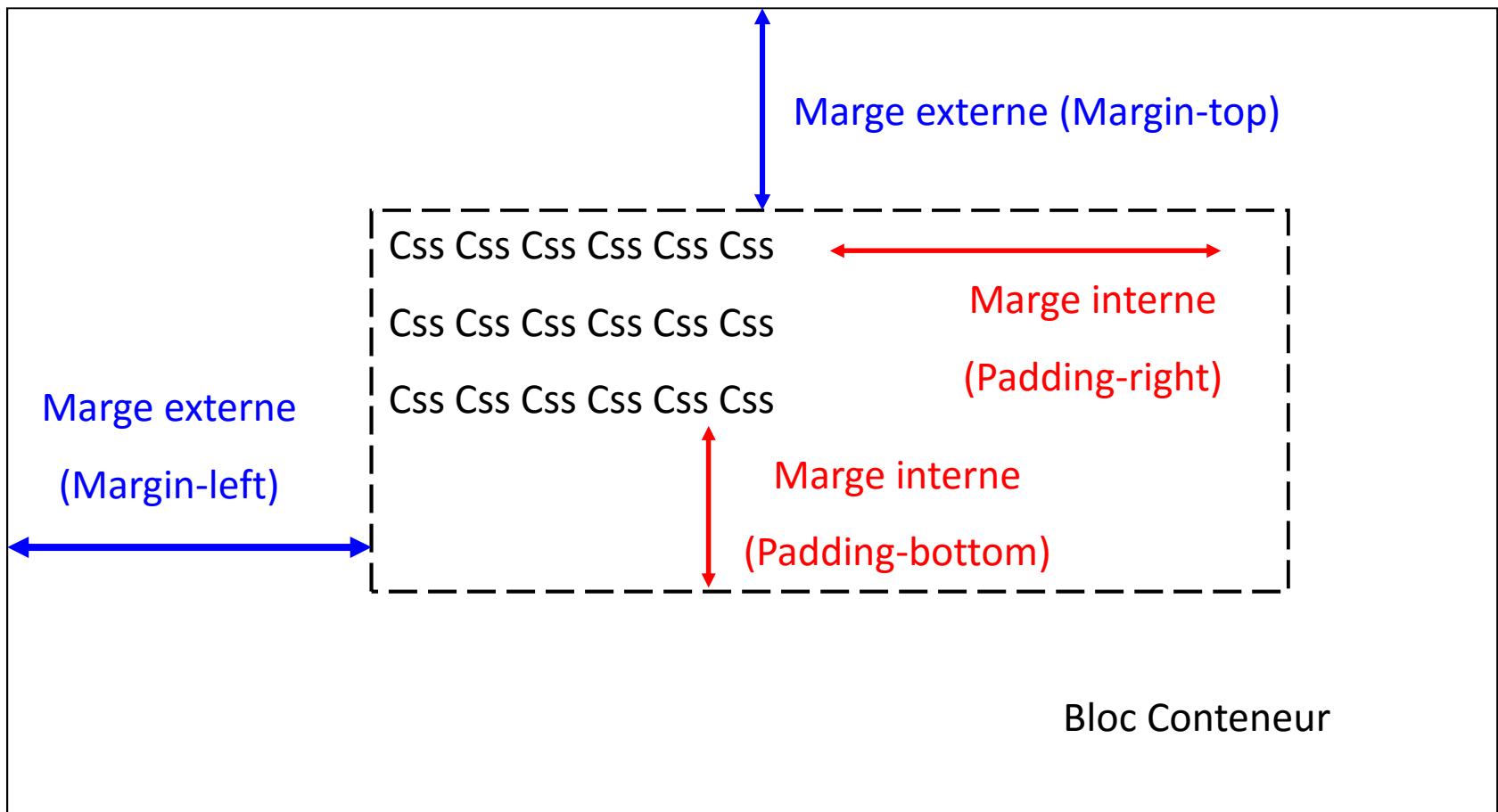
- La couleur par défaut du texte est blue navy
- Les titres sont en police "comic sans MS" ou "sans serif"
- Le texte est en police est "Arial" ou "sans serif"
- La page contient l'image "fond\_spirale.gif" (image à répéter horizontalement)
- Pour ménager de la place à gauche pour la spirale on a défini une marge gauche de 50 pixels.
- Les liens pas encore visités sont colorés en rouge et les liens déjà visités en marron.
- Le texte en TT en couleur bleu et police monospace

Couleur de fond "silver"	
Texte en police Arial narrow et couleur du texte green	Texte en police Arial narrow et couleur du texte green



# Les propriétés margin et padding

Marge interne et marge externe



The CSS logo consists of the letters 'CSS' in a bold, black, sans-serif font. A red speech bubble icon containing the number '2' is positioned to the right of the letter 'S'. Below the letters, a red pen lies horizontally.

## Example (1/6)

belle dans mon miroir je suis la plus belle dans mon miroir

三

miroir

**<H2>**

---

**je suis la plus belle dans mon miroir**

六

je suis la plus belle dans mon miroir je suis la plus belle dans mon miroir  
je suis la plus belle dans mon miroir je suis la plus belle dans mon miroir je suis la plus  
belle dans mon miroir je suis la plus belle dans mon miroir je suis la plus belle dans mon



## Marge interne et marge externe

# Exemple (2/6)

```
h1 {  
margin: 30px ;color:#000000;  
border-right: 5px solid black ;  
border-top: 5px solid #000000 ;  
padding-left: 40%;  
margin-right: 10px ;  
}
```

belle dans mon miroir je suis la plus belle dans mon miroir je suis la plus belle dans mon miroir je suis la plus belle dans mon miroir  
je suis la plus belle dans mon miroir

je suis la plus belle dans mon  
miroir



## Marge interne et marge externe

## Exemple (3/6)

```
h2 {  
padding-left: 25px ;  
border-top: 1px solid #ffffff;  
font-size: 1.4em ;  
border-bottom: 2px solid #ffffff ;  
}
```

**je suis la plus belle dans mon miroir**



## Marge interne et marge externe

### Exemple (4/6)

```
h3 {  
margin-left: 15px ;  
padding-left: 5px ;  
border-bottom: 1px solid #9b2 ;  
border-left: 3px solid #9b2 ;  
color: #9b2 ;  
}
```

mon miroir

je suis la plus belle dans mon miroir



## Exemple (5/6)

### Marge interne et marge externe

margin: 50px 100px ;

The screenshot shows a Mozilla Firefox browser window with a red background. It displays three examples of text layout:

- Top Example:** A large block of white text: "je suis la plus belle dans mon miroir je suis la plus belle dans mon miroir". A green arrow points to the top margin, and a blue arrow points to the bottom margin.
- Middle Example:** A bolded text block: "je suis la plus belle dans mon miroir". This block has a black border and a white background.
- Bottom Example:** Another large block of white text: "je suis la plus belle dans mon miroir je suis la plus belle dans mon miroir".

Dotted lines indicate the boundaries of the margin areas around the text blocks.



## Exemple (6/6)

### Marge interne et marge externe

padding-right: 200px ;

The screenshot shows a Mozilla browser window with a red background. It displays three examples of text layout:

- Top Example:** A large block of text "je suis la plus belle dans mon miroir" repeated multiple times, enclosed in a dotted border. A green arrow points from the right edge of the text area towards the right margin of the page.
- Middle Example:** A bold, centered text block: "je suis la plus belle dans mon miroir". This block is enclosed in a solid black border.
- Bottom Example:** Another block of the same repeating text, enclosed in a dotted border.

The overall layout demonstrates how internal padding (the black border) and external padding (the red background and the green arrow pointing to the right margin) affect the visual presentation of the text.



# Image de fond en CSS

➤ **background-image** : définit l'image de l'arrière-plan URL de l'image

```
Body {background-image: url (image.gif)}
```



## Image de fond en CSS

```
<html>
  <head>
    <style >
      html {
        background-color: black;
        background-image: url('roger_rabbit.jpg');
        color: white;
      }
    </style>
  </head>
  <body>
    <p>Roger Rabbit: Yeah ...</p>
  </body>
</html>
```

# Exemple





# background-repeat

- Permet de définir si l'image doit être répété ou pas
  - repeat, repeat-x,  
repeat-y, no-repeat

...

```
body {  
    background-color: black;  
    background-image: url(roger_rabbit.jpg);  
    background-repeat: no-repeat;  
    color:white;  
}  
...
```





# Positionner une image de fond: **background-position**

- Permet de définir la position de l'image dans l'élément
  - <pourcentage x> <pourcentage y>
  - <unité x> <unité y> (ex: 30px 50px)
  - top | center | bottom et  
left | center | right (ex: top left)
- ...

```
body {  
    background-color: black;  
    background-image: url(roger_rabbit.jpg);  
background-repeat: no-repeat;  
    background-position: 30px 150px;  
    color:blue;  
}  
...
```





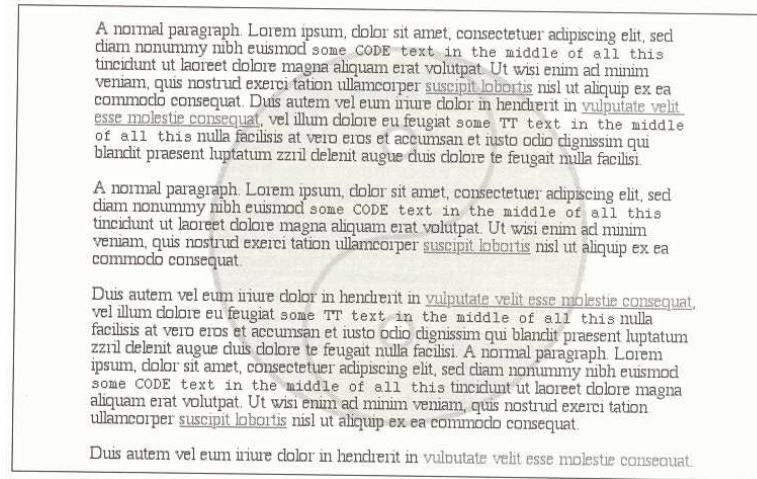
# Positionner une image de fond: **background-position**

- Si l'on veut qu'elle se place en haut à droite, on rajoutera le code :  
**background-position: right top;**  
ou  
**background-position:100% 0%;**
- Pour une position au centre de la page, cela donnera :  
**background-position:center center;**  
ou  
**background-position: 50% 50%;**
- Pour une position en bas à droite :  
**background-position:right bottom;**  
ou  
**background-position:100% 100%;**



# Fond de page fixe : background-attachment

- Pour que le fond de page devienne fixe, on rajoutera dans le code :  
**background-attachment: fixed;**



→ L'image demeure au centre même lorsque l'on déroule le texte

- Valeur par défaut  
**background-attachment: scroll**



# Background

- On peut aussi cumuler ces différentes informations dans un unique background.

- Exemple :

```
body  
{  
  background: white url(image_de_fond.png) no-repeat  
  right bottom;  
}
```



## Nouveauté en **CSS3** : Multiple background

- La propriété **background-image** n'acceptait, en **CSS 2.1**, qu'une seule image.
- Les **CSS3** permettent des background multiples
- Exemple :  
**Une image aux quatre coins d'une page**

```
Body {  
    background: url(images/houx.gif) no-repeat left top,  
               url(images/houx.gif) no-repeat right top,  
               url(images/houx.gif) no-repeat left bottom,  
               url(images/houx.gif) no-repeat right bottom;  
}
```



# Changement d'image de fond au survol du lien

```
#menu a:hover
```

```
{ background-image:url(images/gris_anim.gif);  
background-repeat:no-repeat;  
}
```

/\*Pour pouvoir combiner une image de fond (ici gris\_anim.gif) et un caractère (ici un guillemet), on peut utiliser l'attribut content et le pseudo élément before\*/

```
#menu a:hover:before
```

```
{ content:" » "; }
```





# Positionnement et flux

- Flux : ordre d'affichage des éléments
- 4 méthodes de positionnement
  - normal
  - float
  - relatif
  - absolu
- Quels sont les différences ?

# Positionnement et flux



## Le flux normal

```
<BODY>
<div class="conteneur">
    <div class="normalA">
        Bloc A
    </div>
    <div class="normalB">
        Bloc B
    </div>
</div>
</BODY>
```

```
.normalA {
width:150px;
height:150px;
background-color:red;
border:1px solid black;
}
```

```
.normalB {
width:250px ;
height:100px ;
background-color:green;
border:1px solid black;
}
```

The screenshot shows a window titled "Internet Explorer" with the address bar containing "Le CSS c'est pratique !!!!". Inside the browser, there is a large black rectangular container labeled "Bloc conteneur" at the bottom. Inside this container, there are two other blocks: a red one labeled "Bloc A" and a green one labeled "Bloc B". A large red "X" is positioned in the top-left corner of the browser window. The bottom right corner of the slide contains the number "79".



## Positionnement et flux

# Le flux normal en bloc

```
<BODY>
<div class="conteneur">
    <div class="normalA">
        Bloc A
    </div>
    <div class="normalB">
        Bloc B
    </div>
</div>
</BODY>

.normalA {
width:150px;
height:150px;
background-color:red;
border:1px solid black;
}

.normalB {
width:250px ;
height:100px ;
background-color:green;
border:1px solid black;
}
```

Bloc conteneur

Flux normal en bloc :  
Succession verticale



## Positionnement et flux

# Le flux normal en ligne

```
<BODY>
<div class="conteneur">
    <span class="normalC">
        Bloc C
    </span>
    <span class="normalD">
        Bloc D
    </span>
</div>
</BODY>
```

```
.normalC {
width:150px;
height:150px;
background-color:red;
border:1px solid black;
}
```

```
.normalD {
width:250px ;
height:100px ;
background-color:green;
border:1px solid black;
}
```

Internet Explorer

Précédente ▶ 🔍 Rechercher Favoris Outils ?

OK Liens >

Adresse : Le CSS c'est pratique !!!!

Bloc conteneur

**Flux normal en-ligne :**

**Succession horizontale**

Bloc D

C



## Noter bien

➤ CSS:

```
.bloc_relatif  
{position:relative;left:30px;}
```

➤ HTML:

Sous le pont Mirabeau

```
<div class="bloc_relatif"> div class="bloc_relatif" </div>  
La joie venait toujours après la peine  
</div>
```

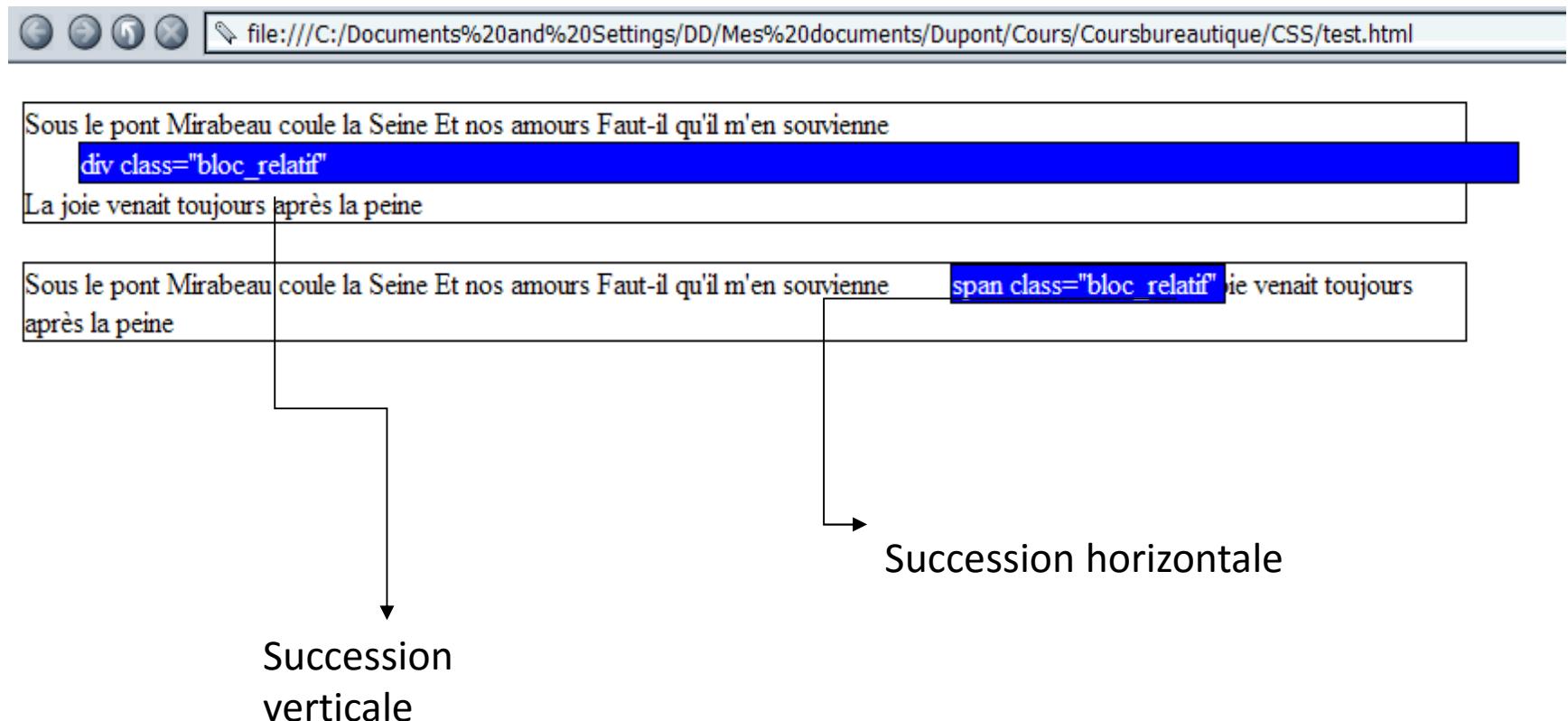
Sous le pont Mirabeau

```
<span class="bloc_relatif"> span class="bloc_relatif" </span>  
La joie venait toujours après la peine  
</div>
```



## Positionnement et flux

# Noter bien la différence : inline et bloc





## Positionnement et flux

Noter bien la différence : inline et bloc

## Bienvenue a mon site.

Blog, cv, info,...

- [Accueil](#)
  - [Blog](#)
  - [CV](#)
  - [Cours](#)
  - [Download](#)
  - [Liens utiles](#)
  - [Contact](#)

## À propos de l'auteur

Je suis Anas tarek, je suis né un 23 octobre 2011.

**Je suis un grand voyageur.**

Copyright Anas Tarek -2012- Tous droits réservés - [Me contacter](#).



## Positionnement et flux

# Noter bien la différence : inline et bloc

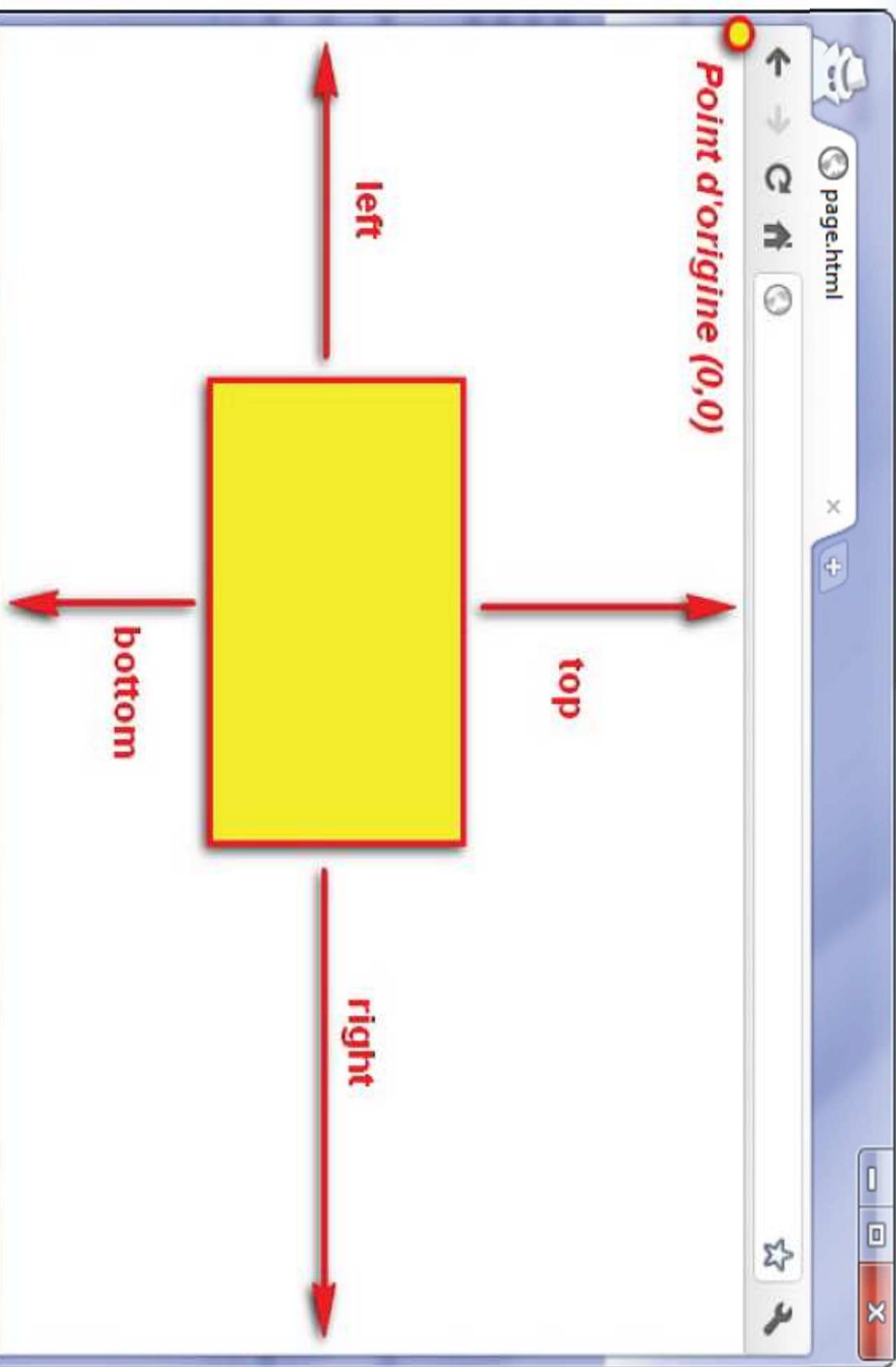
```
<html>
<head>
<title>Site Web</title>
<style>
header{
border : 5px inset red;
padding : 5px 5px;}
nav{
float: left;
width: 150px;
border: 3px solid black;}
section{
margin-left: 170px;
border: 3px solid blue;
padding : 5px 5px 5px 5px;}
footer{
margin-left: 170px;
border: 3px outset green;
text-align: center;}
</style>
</head>
```

```
<body>
<header> <h1>Bienvenue a mon site.</h1>
</header>
<nav>
<ul>
<li><a href="#">Accueil</a></li>
<li><a href="#">Blog</a></li>
<li><a href="#">CV</a></li>
<li><a href="#">Cours</a></li>
<li><a href="#">Download</a></li>
<li><a href="#">Liens utiles</a></li>
<li><a href="#">Contact</a></li>
</ul></nav>
<section>
<aside>
<h1>À propos de l'auteur</h1><p>Je suis ...</p>
</aside>
<article>
<h1>Je suis un grand voyageur.</h1><p>...</p>
</article> </section>
<footer><p>Copyright Le proprio- Tous droits réservés - <a href="#">Me contacter.</a></p>
</footer></body></html>
```

# Positionnement et flux



Exemple



Positionnement absolu de l'élément sur la page



## Noter bien la différence : inline et bloc

- ❖ **Le positionnement absolu** permet de placer un élément (réellement) n'importe où sur la page.
- ❖ Pour effectuer un positionnement absolu, on doit écrire :  
**Element{ position: absolute;}**
- ❖ On a dit qu'on voulait un positionnement absolu, mais encore faut-il dire où l'on veut que le bloc soit positionné sur la page.
- ❖ Pour ce faire, on va utiliser quatre propriétés CSS :
  - ✓ **left : position par rapport à la gauche de la page ;**
  - ✓ **right : position par rapport à la droite de la page ;**
  - ✓ **top : position par rapport au haut de la page ;**
  - ✓ **bottom : position par rapport au bas de la page.**
- ❖ On peut leur donner une valeur en pixels, comme 14px, ou bien une valeur en pourcentage, comme 50%.



# Le positionnement absolu

- Le bloc doit être positionné tout en bas à droite (0 pixel par rapport à la droite de la page, 0 par rapport au bas de la page).

```
nav{  
    position: absolute;  
    right: 0px;  
    bottom: 0px; }
```



## Positionnement fixe

➤ **Le positionnement fixe:** Le principe est exactement le même que pour le positionnement absolu sauf que, cette fois, le bloc reste fixe à sa position, même si on descend plus bas dans la page.

```
element{  
    position: fixed;  
    right: 0px;  
    bottom: 0px; }
```



## Le positionnement absolu

- Le positionnement relatif permet d'effectuer des «ajustements» : l'élément est décalé par rapport à sa position initiale.
- Par exemple un texte important, situé entre deux balises `<strong>`.
- Pour commencer, je le mets sur fond rouge pour qu'on puisse mieux le repérer :
  - `strong{ background-color: red; /* Fond rouge */`
  - `color: yellow; /* Texte de couleur jaune */ }`

**Point d'origine(0,0)**

C'est moi, **bonjour les amis!** Je suis né un 23 novembre 2005.

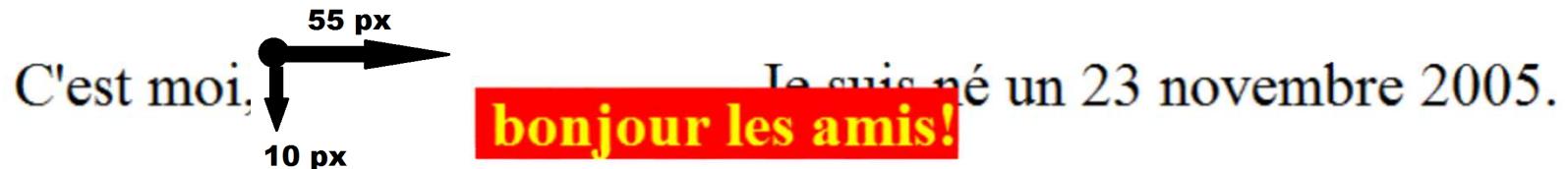
C'est moi, **bonjour les amis!** Je suis né un 23 novembre 2005.



## Le positionnement relatif

- Si vous faites un **position: relative;** le texte sur fond rouge va se déplacer par rapport à la position où il se trouve.
- Je veux que mon texte se décale de 55 pixels vers la droite et de 10 pixels vers le bas.
- Je vais donc demander à ce qu'il soit décalé de 55 pixels par rapport au « bord gauche » et de 10 pixels par rapport au « bord haut ».

```
Strong{background-color: red;  
color: yellow;  
position: relative;  
left: 55px;  
top: 10px; }
```





## Positionnement et flux

```
<BODY>
<div class="conteneur">
    <div class="flotteA">
        Boîte A
    </div>
    <p>
        Texte...blabla ...
    </p>
</div>
</BODY>
```

```
.flotteA {
float:left;
width:500px;
background-color:yellow;
border:1px solid black;
}
```

# Le flux flottant

The screenshot shows a window titled "Internet Explorer" with the address bar containing "Le CSS c'est pratique !!!!". A red 'X' mark is placed over the top-left corner of the main content area. The content itself consists of a large yellow rectangular box labeled "Boîte A – float:left" at the bottom, and below it, a smaller black rectangular box labeled "Bloc conteneur".

Bloc conteneur

Boîte A – float:left

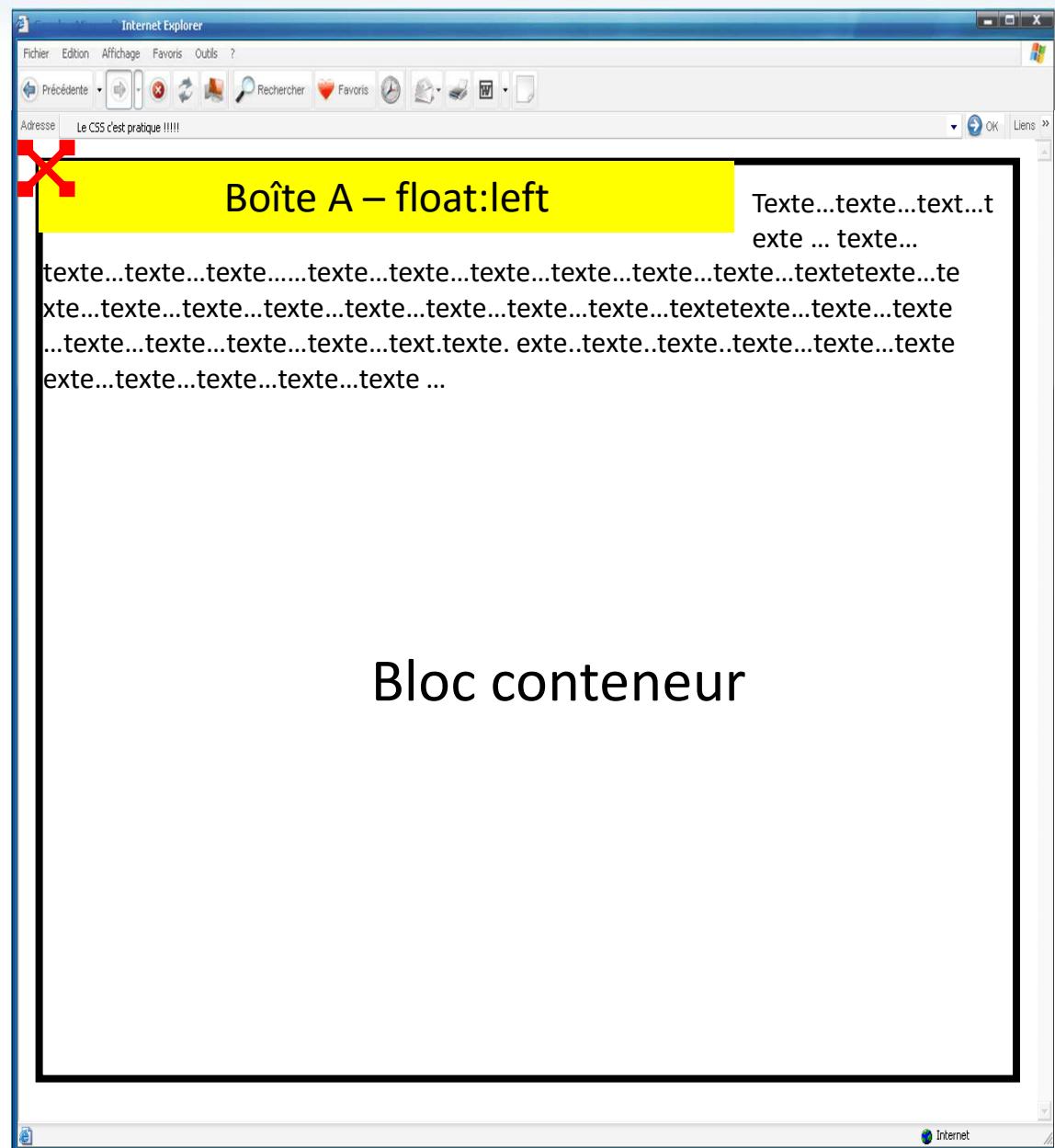


## Positionnement et flux

```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <p>
    Texte...blabla ...
  </p>
</div>
</BODY>
```

```
.flotteA {  
float:left;  
width:500px;  
background-color:yellow;  
border:1px solid black;  
}
```

# Le flux flottant





## Positionnement et flux

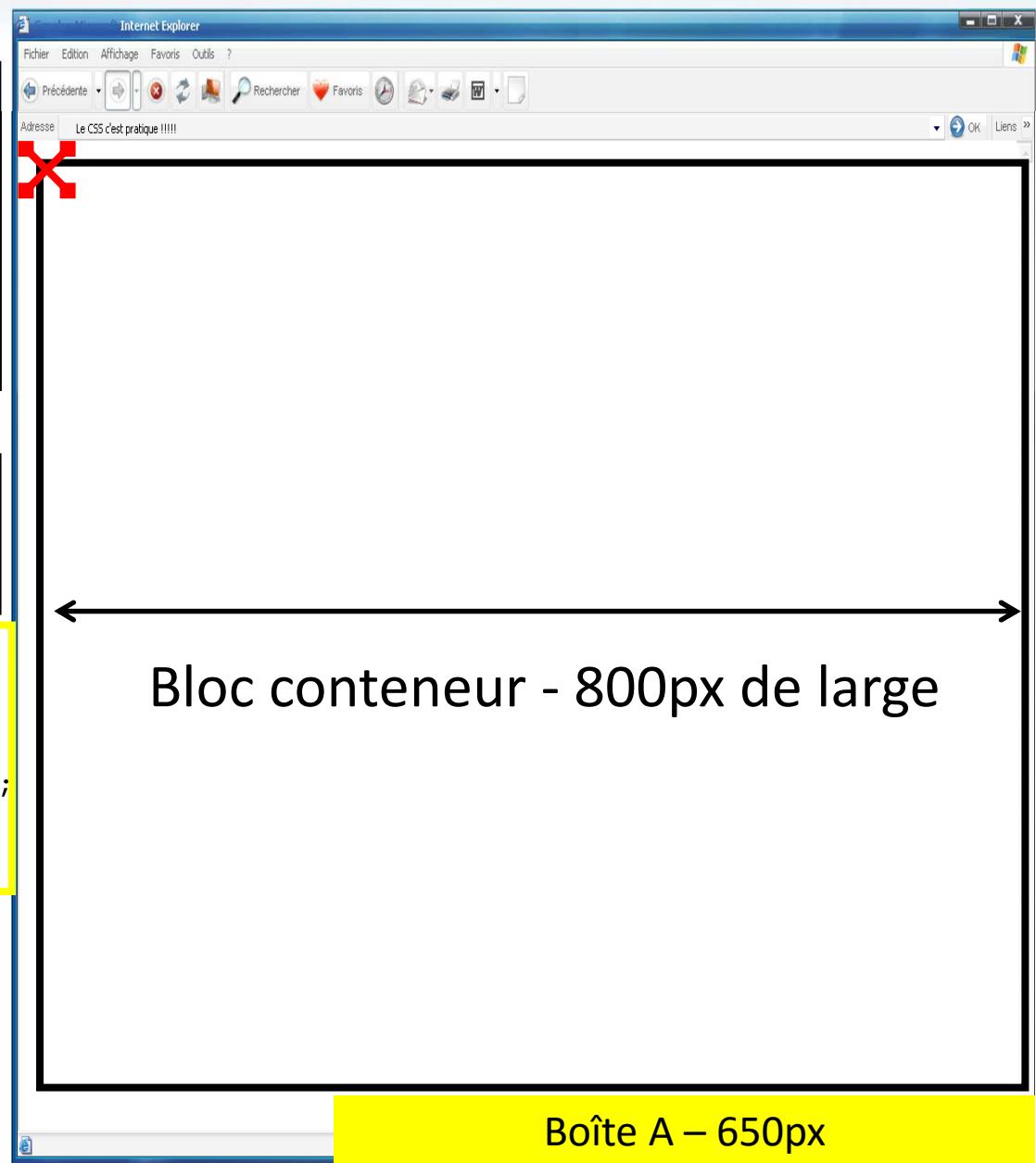
# Le flux flottant

```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <div class="flotteB">
    Boîte B
  </div>
</div>
</BODY>

.conteneur {
width:800px;
border:1px solid black;
}

.flotteA {
float:left;
width:650px;
background-color:yellow;
border:1px solid black;
}

.flotteB {
float:left;
width:100px;
background-color:blue;
border:1px solid black;
}
```



## Positionnement et flux



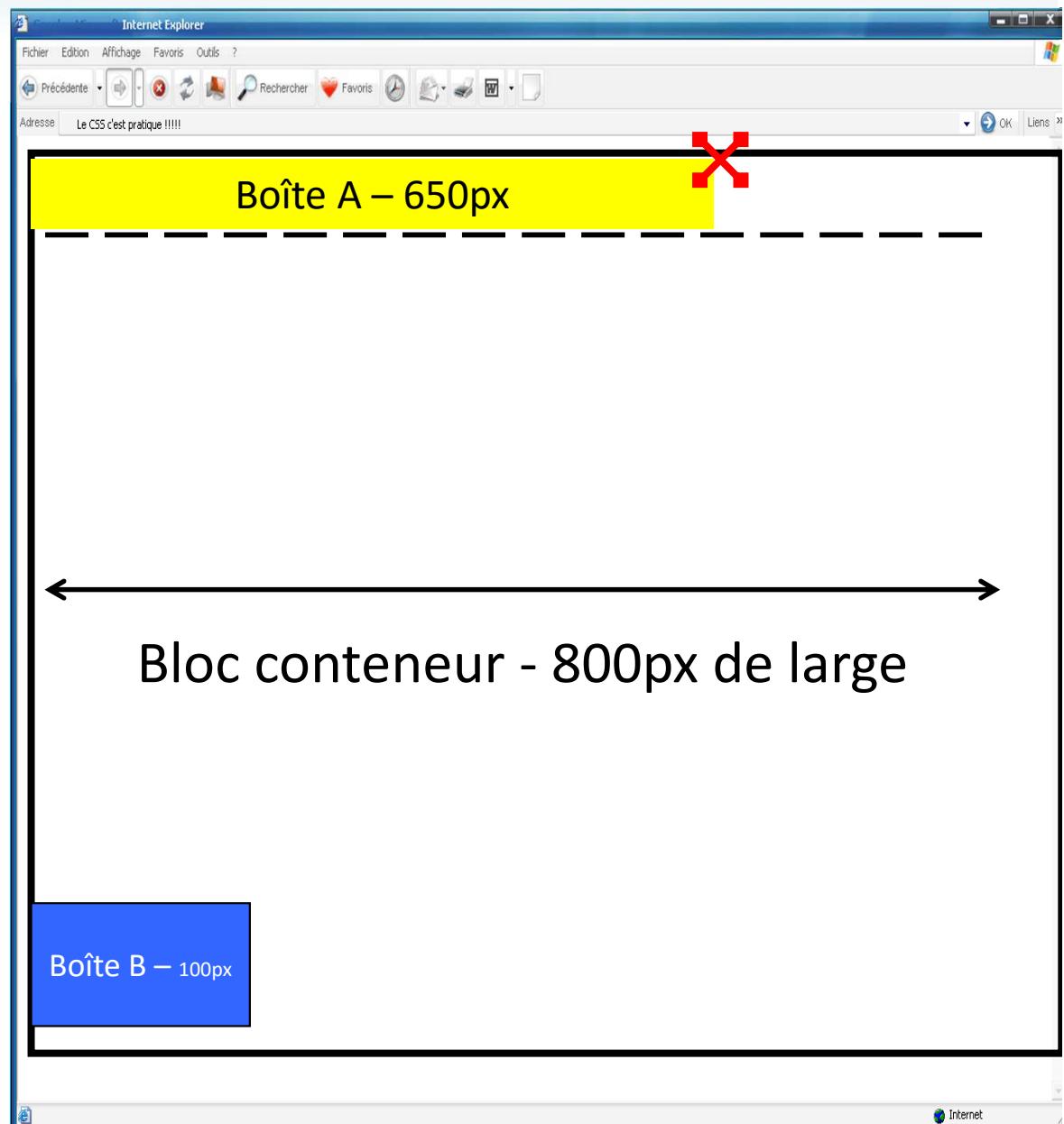
```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <div class="flotteB">
    Boîte B
  </div>
</div>
</BODY>

.conteneur {
width:800px;
border:1px solid black;
}

.flotteA {
float:left;
width:650px;
background-color:yellow;
border:1px solid black;
}

.flotteB {
float:left;
width:100px;
background-color:blue;
border:1px solid black;
}
```

## Le flux flottant





## Positionnement et flux

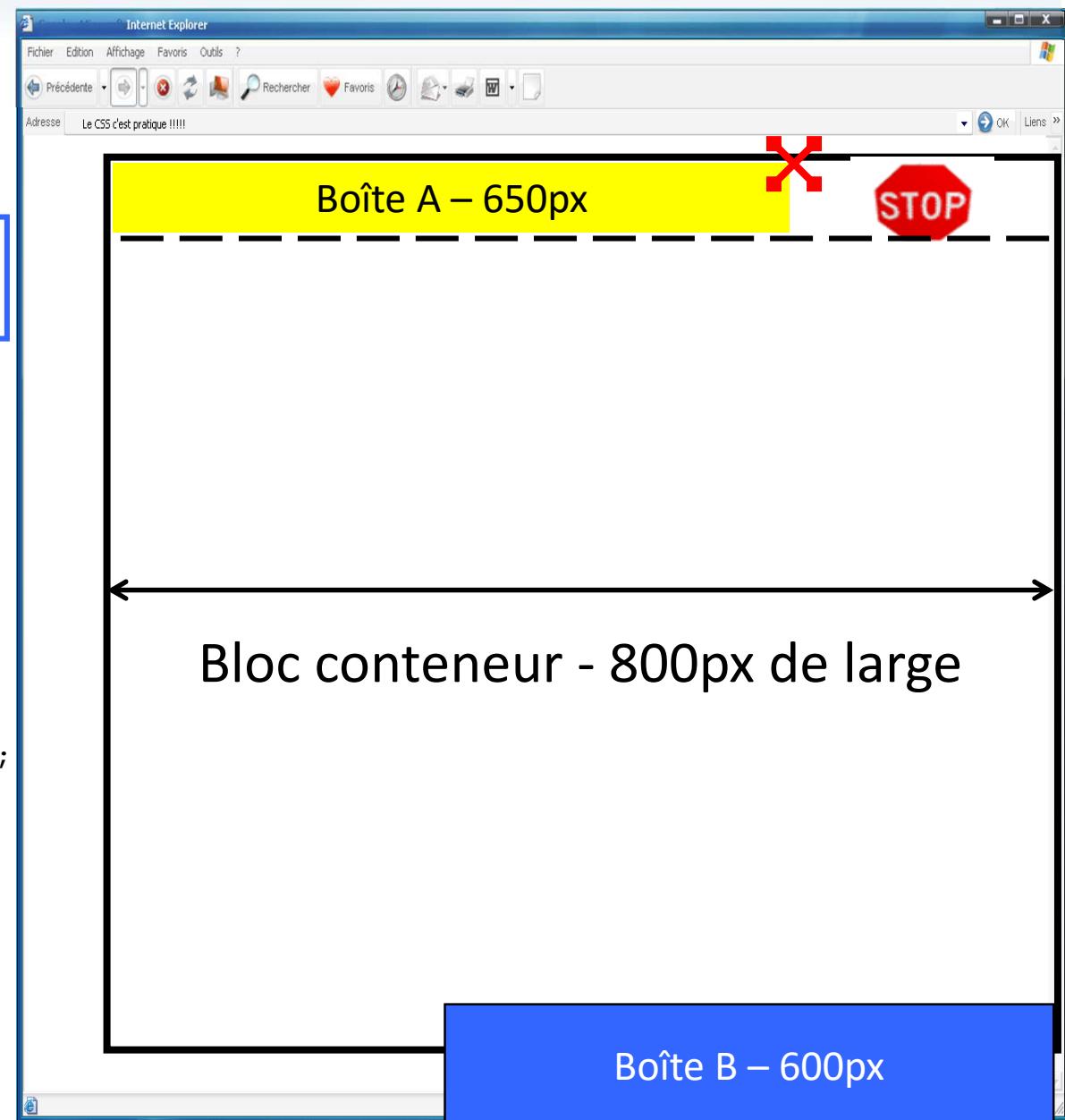
```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <div class="flotteB">
    Boîte B
  </div>
</div>
</BODY>

.conteneur {
width:800px;
border:1px solid black;
}

.flotteA {
float:left;
width:650px;
background-color:yellow;
border:1px solid black;
}

.flotteB {
float:left;
width:600px;
background-color:blue;
border:1px solid black;
}
```

# Le flux flottant



## Positionnement et flux



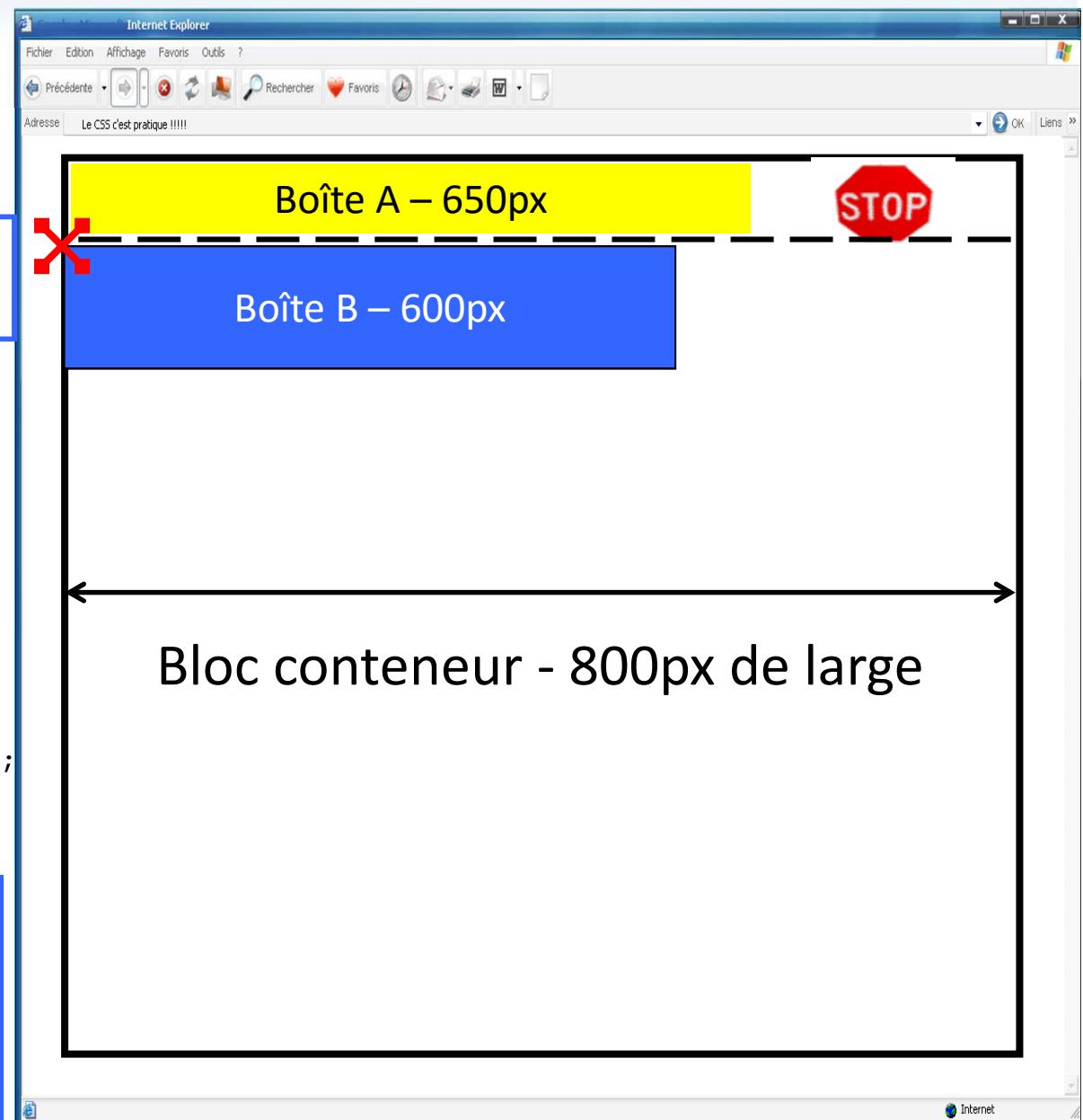
```
<BODY>
<div class="conteneur">
  <div class="flotteA">
    Boîte A
  </div>
  <div class="flotteB">
    Boîte B
  </div>
</div>
</BODY>

.conteneur {
width:800px;
border:1px solid black;
}

.flotteA {
float:left;
width:650px;
background-color:yellow;
border:1px solid black;
}

.flotteB {
float:left;
width:600px;
background-color:blue;
border:1px solid black;
}
```

# Le flux flottant





## Positionnement et flux

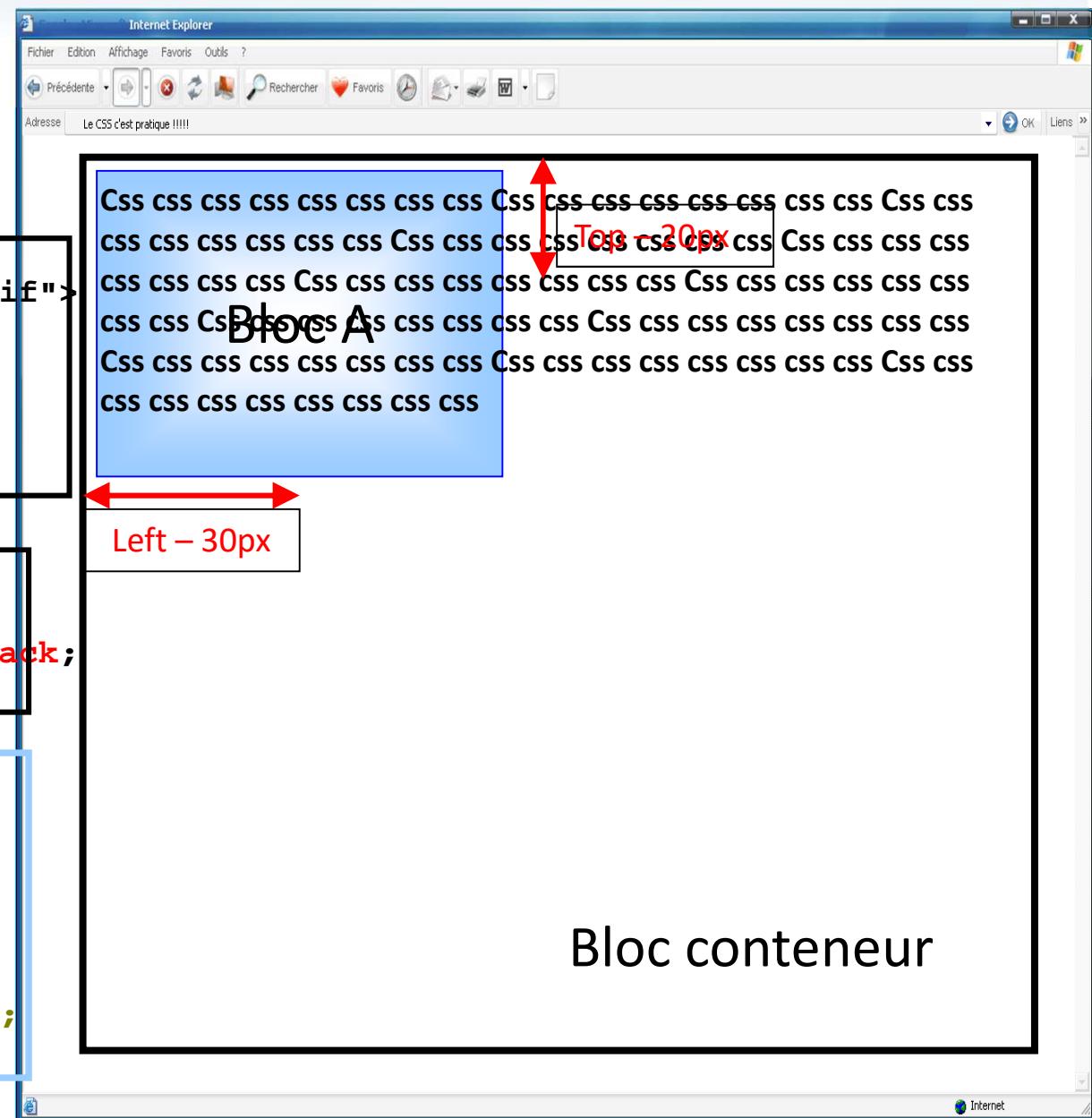
# Le flux relatif

```
<BODY>

<div class="conteneur">
    <div class="bloc_relatif">
        Bloc A
    </div>
    CSS CSS CSS CSS...
</div>
</BODY>
```

```
.conteneur {  
    width:800px;  
    border:1px solid black;  
}
```

```
.bloc_relatif {  
position: relative;  
width: 300px;  
margin-top: 20px;  
margin-left: 30px;  
border: 1px solid black;  
}
```





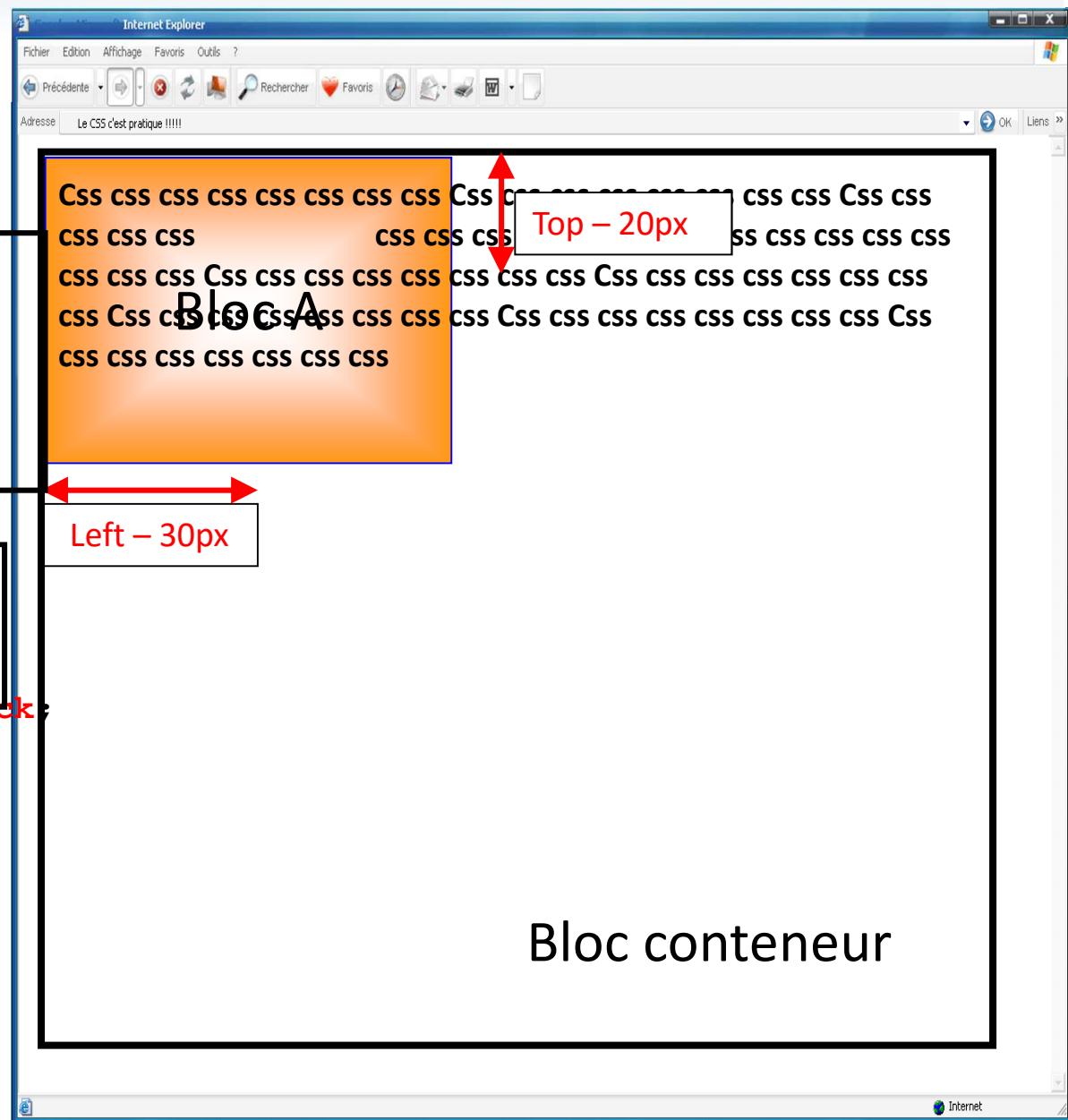
# Flux absolu

## Positionnement et flux

```
<BODY>
<div class="conteneur">
  <div class="bloc_absolu">
    Bloc A
  </div>
  CSS CSS CSS ...
</div>
</BODY>

.conteneur {
  width:800px;
  border:1px solid black;
}

.bloc_absolu {
  position:absolute;
  width:300px;
  margin-top:20px;
  margin-left:30px;
  border:1px solid black;
}
```





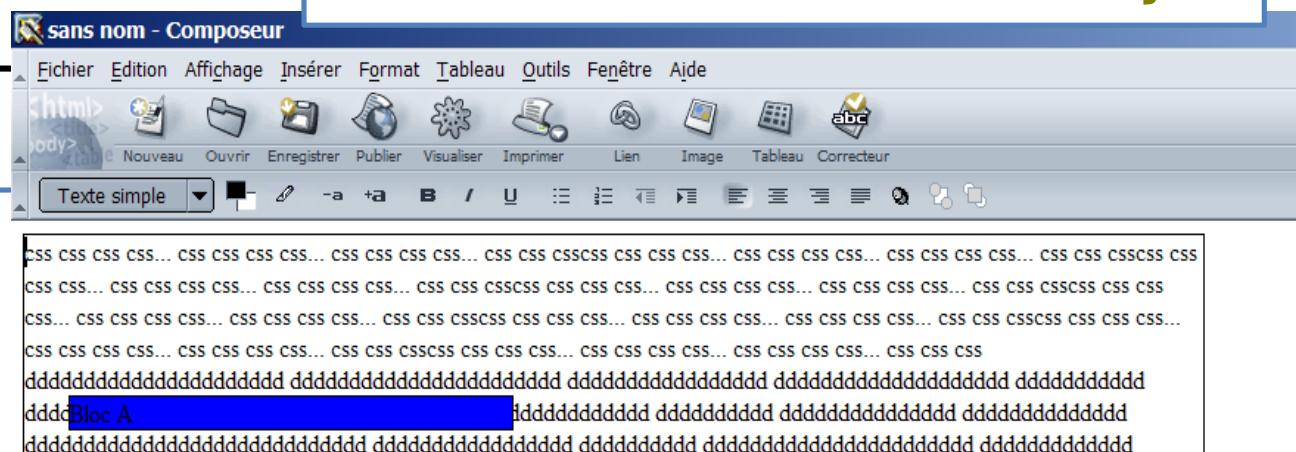
# Flux absolu

## Positionnement et flux

```
<BODY>
<div
  class="conteneur">
CSS CSS CSS CSS...
<div class="bloc_absolu">
  Bloc Aaaaaaaa
</div>
  dddddd....ddd
</div>
</BODY>
```

```
.conteneur {
  width: 800px;
  border: 1px solid black;
}

.bloc_absolu {
  position: absolute;
  width: 300px;
  margin-top: 20px;
  margin-left: 30px;
  border: 1px solid black;}
```



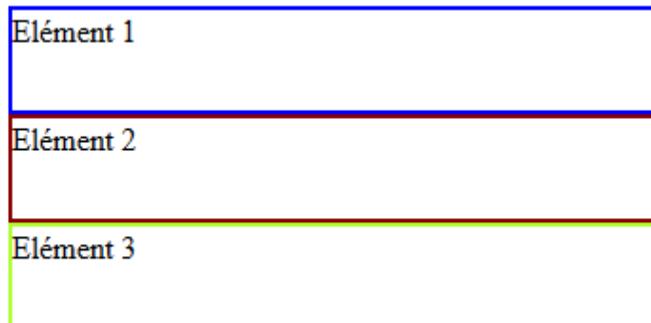


# Positionnement

# Flexbox

Principe : conteneur + éléments

Sans Flexbox



Avec Flexbox



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="css/flextest.css" rel="stylesheet">
    <title>Illustration de la mise en page avec flex</title>
  </head>
  <body>
    <div id="conteneur">
      <div class="element1">Elément 1</div>
      <div class="element2">Elément 2</div>
      <div class="element3">Elément 3</div>
    </div>
  </body>
</html>
```

```
#conteneur
{
  display: flex;
}
.element1
{
  width:25%;
  height:50px;
  border: solid 2px blue;
}
.element2
{
  width:25%;
  height:50px;
  border: solid 2px darkred;
}
.element3
{
  width:25%;
  height:50px;
  border: solid 2px greenyellow;
}
```



# Flexbox

Flex offre une flexibilité dans l'ordonnancement des éléments via la propriété **flex-direction** qui peut prendre les valeurs suivantes:

- **row** : organisés sur une ligne. C'est la valeur par défaut
- **column** : organisés sur une colonne
- **row-reverse** : sur une ligne, mais en ordre inversé
- **column-reverse** : sur une colonne, mais en ordre inversé



# Positionnement : Direction

## Flexbox



The screenshot shows the Chrome DevTools Elements tab with the following details:

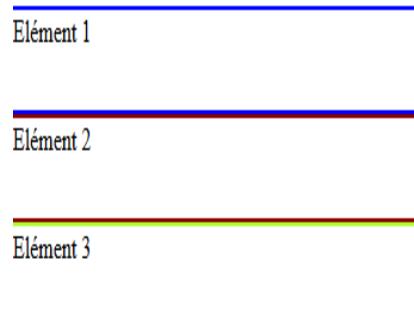
- DOM Tree:**

```
<!DOCTYPE html>
<html hola_ext_inject="disabled">
  <head>...</head>
  <body>
    <div id="conteneur">
      <div class="element1">Elément 1</div>
      <div class="element2">Elément 2</div>
      <div class="element3">Elément 3</div>
    </div>
  </body>
</html>
```
- Elements Panel:** Shows the three elements: "Elément 1", "Elément 2", and "Elément 3".
- Properties Panel:**
  - Style:** The "column" property is selected, highlighted with a red border. Other properties shown include "display: flex;" and "flex-direction: row-reverse;".
  - Event Listener:** A list of event listeners is shown.
  - Filter:** A dropdown menu is open, showing options like "column", "inherit", "initial", and "row-reverse".
  - Properties:** A list of CSS properties for the element, including "margin", "border", and "padding".



# Flexbox

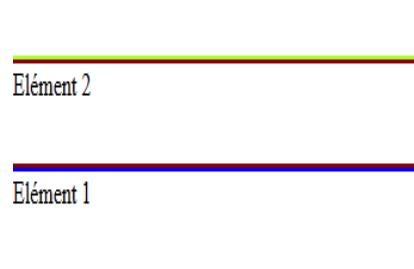
## Positionnement : Direction



column



Column-reverse



row



row



row-reverse

row-reverse



# Flexbox

## WRAP

Le traitement par défaut des blocs consiste à toujours essayer de ce placer sur la même ligne.

La propriété `flex-wrap` permet de gérer l'alignement des blocs au sein même de ma ligne.

- `nowrap` : pas de retour à la ligne c'est la valeur par défaut
- `wrap` : les éléments vont à la ligne lorsqu'il n'y a plus la place
- `wrap-reverse` : les éléments vont à la ligne lorsqu'il n'y a plus la place en sens inverse



# Flexbox

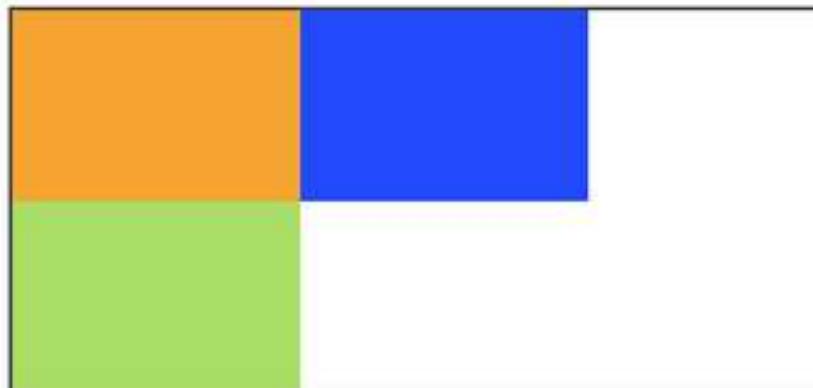
## WRAP

nowrap



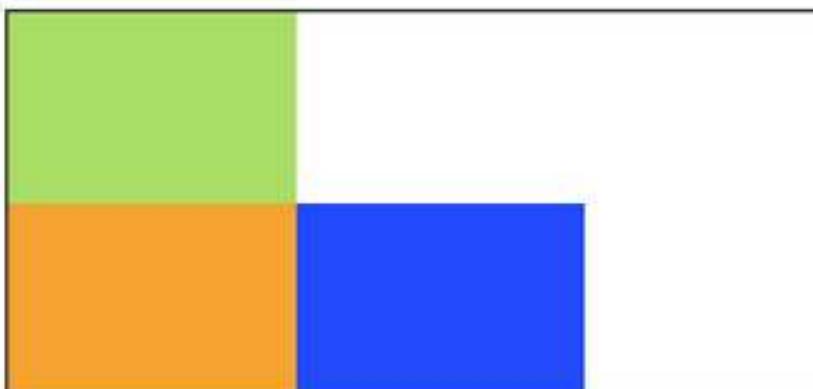
Les éléments se resserrent tant qu'ils peuvent

wrap



Les éléments passent à la ligne

wrap-reverse



Les éléments passent à la ligne à l'envers



# Flexbox

## Alignement

Les blocs sont alignés sur deux axes : **axe principale** (défini par `flex-direction`) et **l'axe secondaire**.

L'**axe secondaire** représente l'axe opposé l'axe que vous utilisez initialement.

- Si vos éléments sont organisés horizontalement, l'axe secondaire est l'axe vertical.
  
- Si vos éléments sont organisés verticalement, l'axe secondaire est l'axe horizontal.



# Flexbox

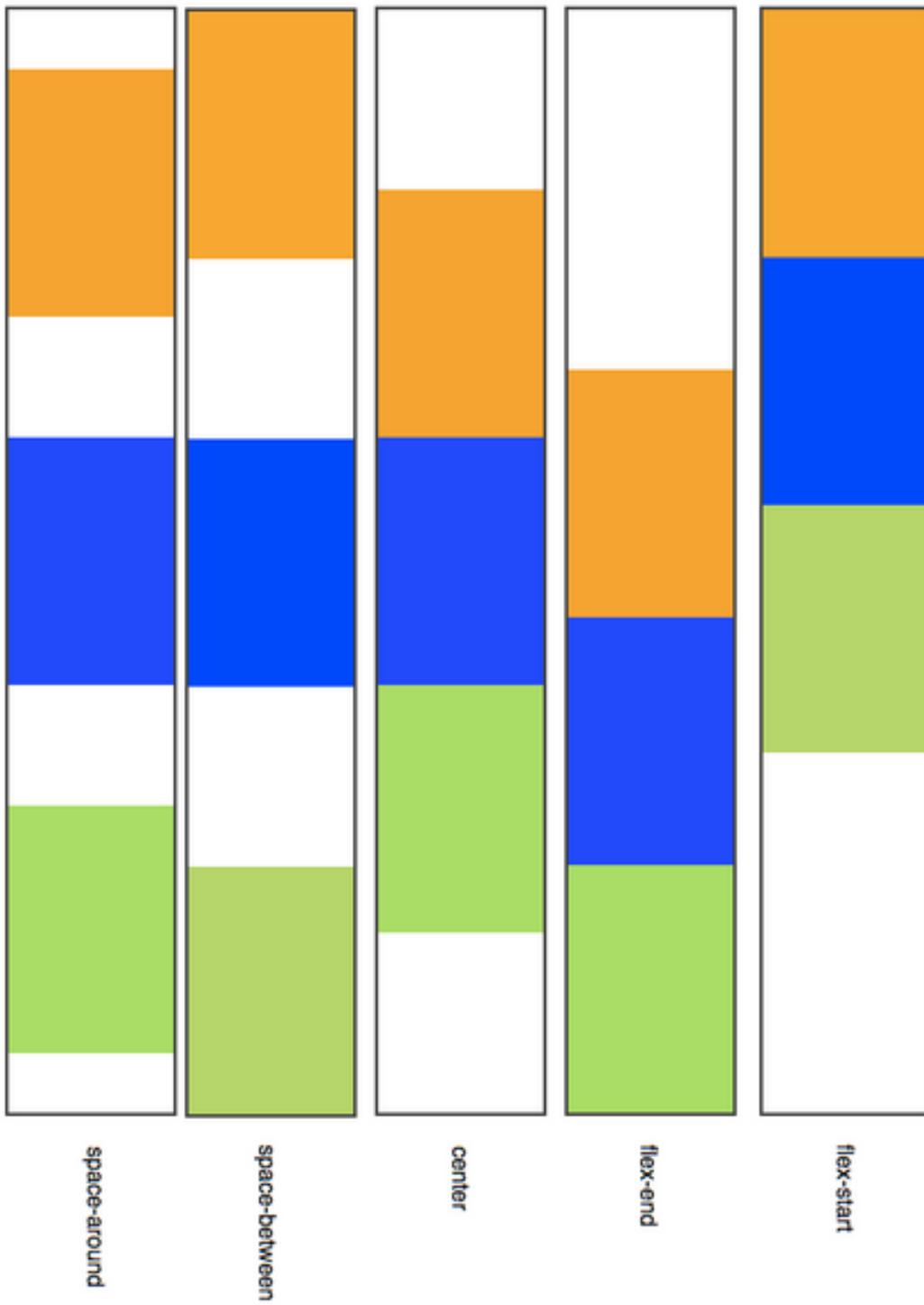
L'alignement sur l'axe principal se fait avec la propriété `justify-content`

- `flex-start` : alignés au début qui est la valeur par défaut
- `flex-end` : alignés à la fin
- `center` : alignés au centre
- `space-between` : les éléments sont étirés sur tout l'axe sans laisser d'espace entre eux
- `space-around` : idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités

# Alignment sur l'axe principal



Flexbox





# Flexbox

L'alignement sur l'axe secondaire se fait avec la propriété `align-items`

- `stretch` : les éléments sont étirés sur tout l'axe ce qui représente la valeur par défaut
- `flex-start` : alignés au début
- `flex-end` : alignés à la fin
- `center` : alignés au centre
- `baseline` : alignés sur la ligne de base ce qui est semblable à `flex-start`



# Flexbox

## Alignement sur l'axe secondaire d'un seul élément

Elément 1



Elément 3



Elément 2



```
#conteneur1
{
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    align-items: center;
}

.element1
{
    width:25%;
    height:50px;
    border: solid 2px blue;
}

.element2
{
    width:25%;
    height:50px;
    border: solid 2px darkred;
    align-self: flex-end;
}

.element3
{
    width:25%;
    height:50px;
    border: solid 2px greenyellow;
}
```



# Flexbox

## Taille des éléments

La propriété `flex` permet de pondérer la taille des éléments.

Si un élément possède la propriété `flex` et la valeur `1`, il prendra tout l'espace restant libre

Sinon si plusieurs éléments possèdent cette propriété alors la valeur représentera le poids de la taille de cet élément

