

A faded, sepia-toned photograph of a mountainous landscape. In the foreground, a large evergreen tree stands on a snowy slope. Behind it, several small, rustic houses with dark roofs are visible. The background features more mountains and a hazy sky. The text "Cascade Style Sheets 2" is overlaid in a bold, orange font.

# Cascade Style Sheets 2



# Problématique

# HTML 3.2

- Syntaxe peu rigoureuse
  - Rendu différent selon les navigateurs
- Mélange des concepts :
  - Balises sémantiques (structure du document) :
    - `<h>`, `<p>` etc
  - Balises de présentation :
    - `<b>`, `<i>` etc

# Accessibilité

- Facilité l'accès des sites
  - Prise en compte du handicap
    - Malvoyance
    - Déficience auditive
    - etc
  - Obligation pour les services publics
- Nécessité pour les navigateurs de réinterpréter la page.

# Exemple trivial

- Malvoyant :
  - Synthèse vocale (ex : Opera pour Windows)
- `<IMG>`
  - Attribut alt => le navigateur lit le texte
  - Sinon ????

# Exemple trivial

- `<i>` (italique)
- Pourquoi avoir utilisé cette balise ?
  - Pour faire joli ?
    - Ne rien faire de plus
  - Pour mettre en valeur un mot ?
    - Dire le mot plus fort par exemple
    - Utiliser une balise sémantique `<em>`

# XHTML 1.0

- Nécessité de séparer contenu et présentation
- Nettoyage de HTML 3.2
- HTML 4.0 (XHTML 1.0)
- Mise en place d'une syntaxe plus rigoureuse
  - Toute balise ouverte doit être fermée
  - Obligation de mettre des valeurs pour les attributs
  - Les valeurs doivent être entre " ou « »
  - Balises en minuscule

# XHTML 1.0

- Suppression des balises de présentation :
  - `<i>`
  - `<font>`
  - `<b>`
  - `<center>`, `<h1 align= « center »>`
  - etc
- On ne garde que la structure du document
  - `<h>`, `<p>`, `<div>`, ....



# Entête

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv= « Content-Type »  
        content= « text/html; charset=utf8 » />  
  </head>  
  
</html>
```



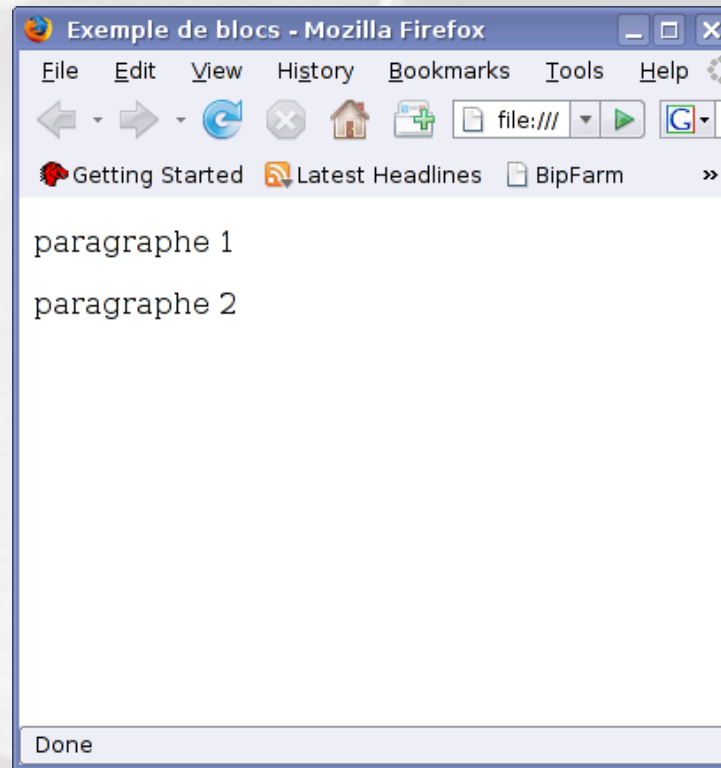
# Principe de base

# Structure

- Deux types d'éléments :
  - Bloc
    - Partie entière du site
    - Ex : `<h1>`, `<p>`, etc.
  - En ligne
    - Mise en valeur d'un morceau de ligne
    - Ex : `<strong>`

# Blocs

- Par défaut : affichage vertical
  - `<p>paragraphe 1</p><p>paragraphe 2</p>`



# Blocs (suite)

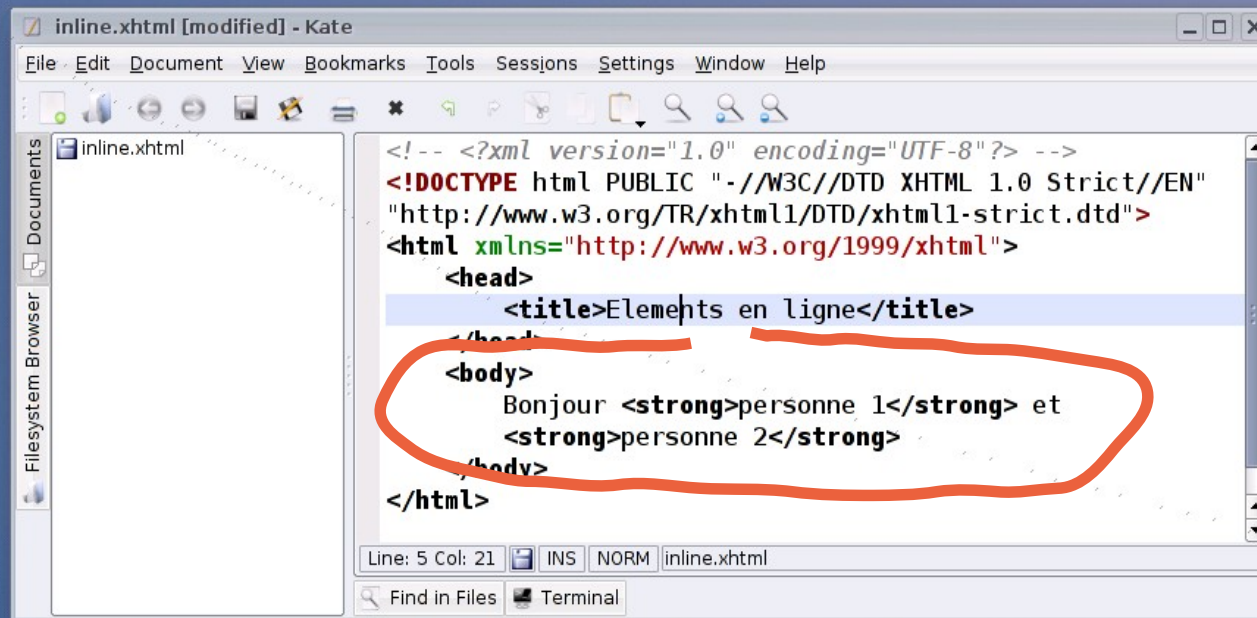
- Quelques exemples de balises blocs

Balise	Commentaire
<p>	paragraphe
<h1> <h2> ...	titres
<ol>	liste ordonnée
<ul>	liste non ordonnée
<table>	tableau
<form>	formulaire
<blockquote>	citations longues
<dl>	liste générique
<div>	bloc générique (pas de présentation ni marge)

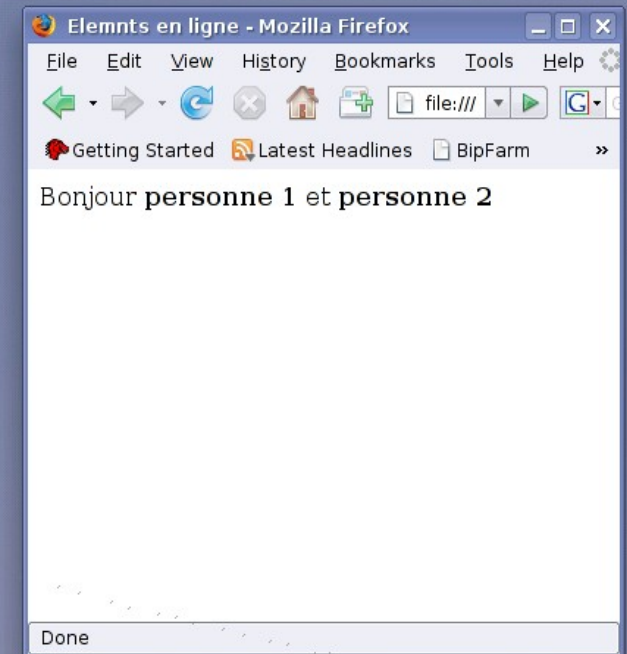
- Un bloc peut contenir des blocs
- Exceptions :
  - <p>, <h1>, <h2> etc.

# Elements en ligne

- Par défaut, affichage « au fil » de la ligne



```
<!-- <?xml version="1.0" encoding="UTF-8" ?> -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Elements en ligne</title>
  </head>
  <body>
    Bonjour <strong>personne 1</strong> et
    <strong>personne 2</strong>
  </body>
</html>
```



# Eléments en ligne (suite)

- Quelques exemples de balises en ligne

Balise	Commentaire
<a>	lien hypertext
<img>	image
<strong>	renforcement
<em>	emphase
<q>	citation courte
<span>	balise générique

- Un élément en ligne peut contenir d'autres éléments en ligne (mais pas de bloc)

# Principe

- Rappel :
  - Séparer la présentation du contenu
  - Choix des balises HTML :
    - Structure du document
    - Pas sa mise en page
- CSS : association balise HTML ↔ présentation
- 2 possibilités :
  - Directement dans le fichier html
  - Dans un fichier séparé



# Principe (suite)

- CSS directement dans le fichier HTML
  - Affaiblit à la séparation contenu présentation
  - Site : présentation homogène
  - Duplication des commandes CSS dans l'ensemble des pages
  - Difficulté de mise à jour

# Séparation HTML - CSS

- Référencer la feuille CSS
  - Dans la partie « <head> » du fichier HTML
  - Deux possibilités :

```
<head>
....
<link rel= 'stylesheet' type= 'text/css'
      href='style.css'/>
...
</head>
```

```
<head>
....
<style type= 'text/css' >
    @import url(style.css);
</style>
....
</head>
```

# Syntaxe : principe de base

```
balise  
{  
propriété: valeur;  
...  
}
```

- Exemple :

```
<head>  
  <style type="text/css">  
    @import url('style.css');  
  </style>  
</head>  
<body>  
  <h1> titre 1 </h1>  
  <h2> titre 2 </h2>  
  <h3> titre 3 </h3>  
</body>
```

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

style.css



# *Sélection de balise (suite)*

- Sélection de base : le type de balise
- Problème : pas assez précis
- Exemple :
  - Un article =
    - 1 résumé
    - le corps de l'article
  - Fichier HTML
    - un paragraphe pour chaque

# Article

```
<body>
  <p>
    Résumé : c'est un article vraiment très intéressant. Je vous conseille de le lire.
    Le message principal est blabla blabla
    blabla blabla blabla blabla blabla blabla blabla
    blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla
    blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla
  </p>

  <p>
    Pourquoi le blabla dans notre société ? plusieurs raisons à cela. Tout d'abord
    blabla blabla blabla blabla blabla blabla blabla blabla
    blabla blabla blabla blabla blabla blabla blabla
    blabla blabla blabla blabla blabla blabla blabla blabla
  </p>
</body>
```

# Classes

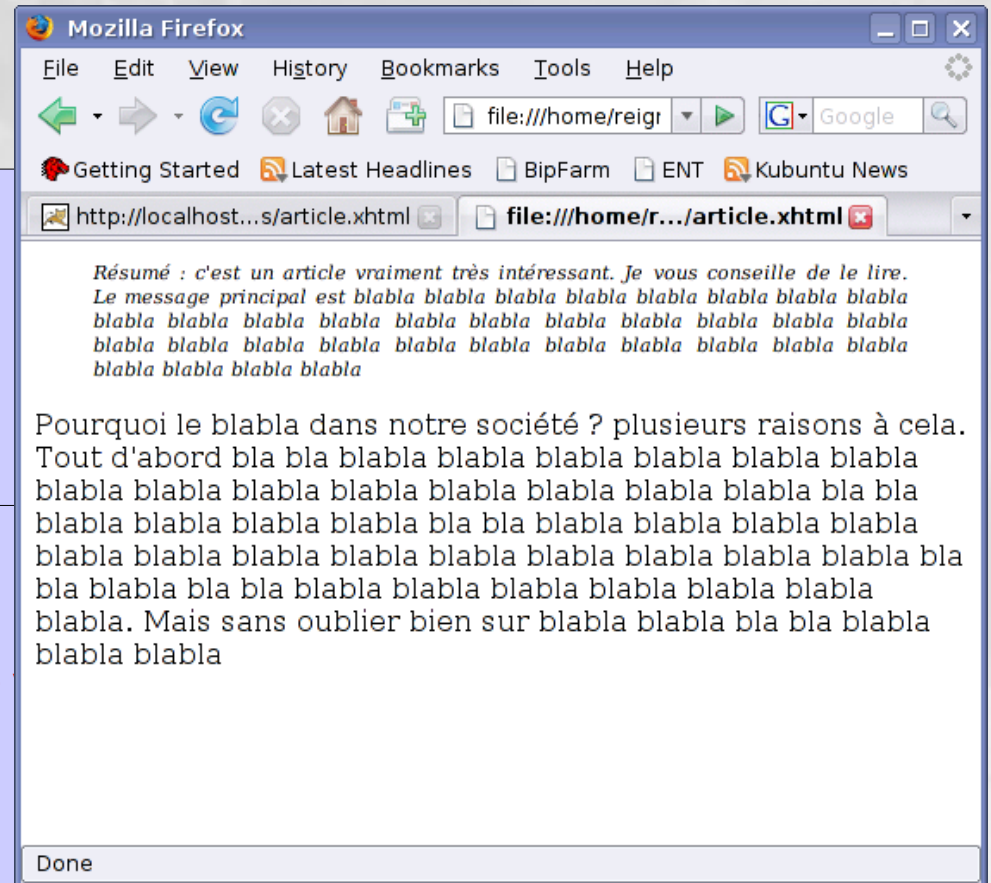
- Associer un nom à un comportement
  - classes
- Associer une ou plusieurs classes aux balises HTML concernées.
- Syntaxe CSS :
  - Le nom de la classe est précédé de « . »

# Article + style

```
.resume
{
  font-style: italic;
  font-size : 0.7em ;
  text-align: justify;
  padding-left: 3em;
  padding-right: 3em ;
}
```

```
<body>
  <p class="resume">
    Résumé : c'est un article
  </p>

  <p>
    Pourquoi le blabla ....
  </p>
</body>
```



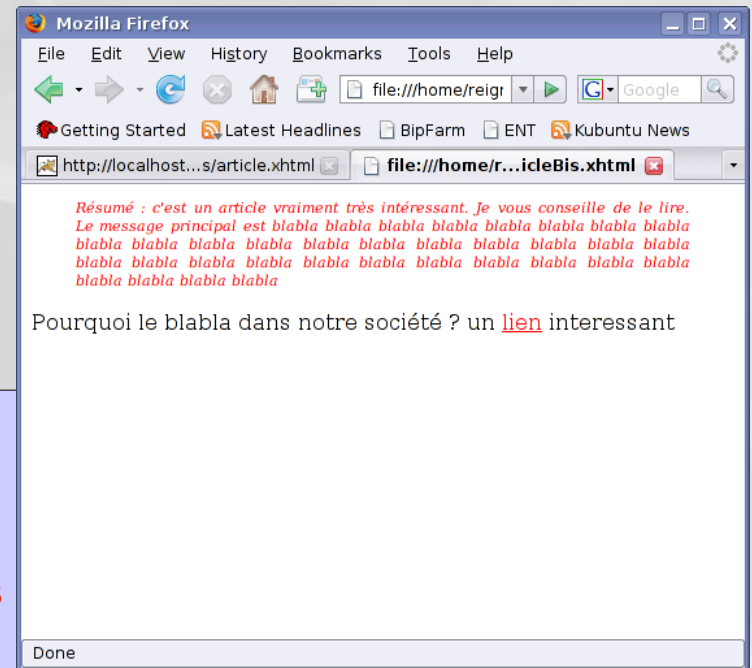
# Association multiple

```
.resume
{
  font-style: italic;
  font-size : 0.7em ;
  text-align: justify;
  padding-left: 3em;
  padding-right: 3em ;
}
```

```
.reinforce
{
  color: red
}
```

```
<body>
  <p class="resume renforce">
    Résumé : c'est un article vraiment très
  </p>

  <p>
    Pourquoi le blabla dans notre société ?
    un <a class="renforce" href="http://labas.fr">lien</a> interessant
  </p>
</body>
```





# Nom de classe

- Reflète la sémantique
- Ne reflète pas la présentation choisie
- Exemple précédent :
  - Classe « renforcer » : ok
  - Classe « rouge » : NON
    - Incohérence si on décide de renforcer avec du vert au lieu du rouge.

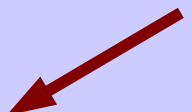
# Instance

- Comportement destiné à une seule balise
- Evite de créer une classe pour un seul élément
- Identification de la balise
  - `<tag id= 'id'>`
  - id doit être unique dans le fichier
- CSS
  - `#id { .... }`

# Article

- Renforcement :
  - Utilisable pour plusieurs balises
  - Classe
- Résumé
  - Un seul résumé par article
  - Utilisation d'une instance au lieu d'une classe

# Article : instance



```
<body>
  <p id="resume" class="renforce">
    Résumé : c'est un article vraiment très intéressant. ...
  </p>

  <p>
    Pourquoi le blabla dans notre société ?
    un <a class="renforce" href="http://labas.fr">lien
  </p>
</body>
```

```
#resume
{
  font-style: italic;
  font-size : 0.7em ;
  text-align: justify;
  padding-left: 3em;
  padding-right: 3em ;
}

.renforce
{
  color: red
}
```

# *Pseudo classes et éléments*

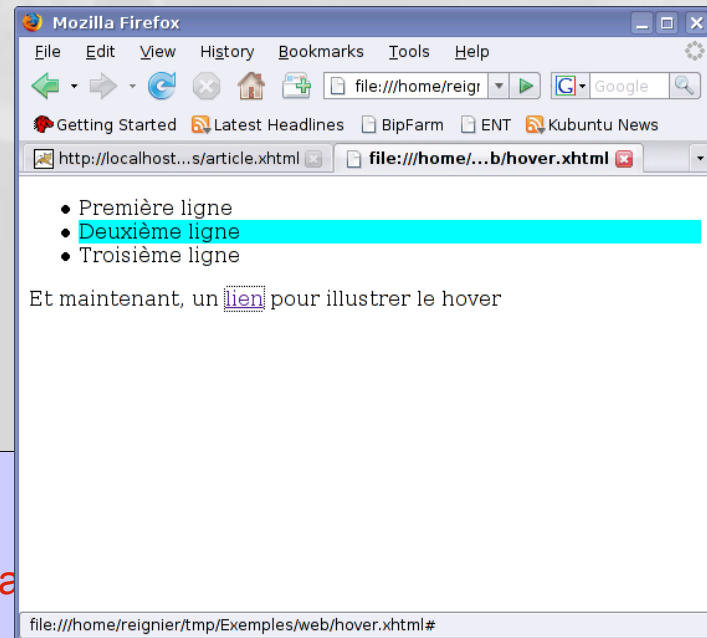
- Pseudo éléments
  - Contrôle d'éléments non décrits en HTML
  - ex : première ligne d'un paragraphe, première lettre d'un mot etc.
- Pseudo classe
  - Présentation fonction de l'état d'un élément
    - Passage au dessus d'une balise, lien visité, etc.
- Syntaxe :
  - Précédé de « : »

# Exemple

```
<body>
  <ul>
    <li>Première ligne</li>
    <li>Deuxième ligne</li>
    <li>Troisième ligne</li>
  </ul>
  Et maintenant, un <a href="#">lien</a>
  pour illustrer le hover
</body>
```

```
li:hover
{
  background-color : a
}
```

```
a:hover
{
  text-decoration : none;
  color: red;
}
```



# Sélection

- Rappel
  - Présentation par type de balise
  - Sélection plus fine :
    - classes + association balise – classe
    - instance
- Approche complémentation :
  - Sélection basée sur la structure

# Contrainte classe - balise

- Rappel :
  - une même classe peut être associée à plusieurs balises
  - `<p class='renforce'>`
  - `<a class='renforce'>`
- Restreindre une classe à un type de balise:
  - préfixer le nom de la classe par le nom de la balise



## Exemple : l'article

<p class="renforce">

Résumé : c'est un article vraiment très intéressant...

</p>

<p>

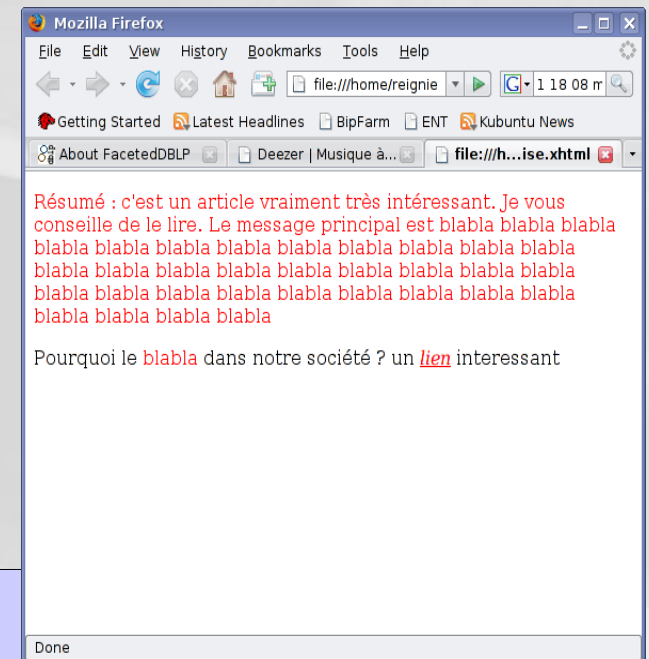
Pourquoi le **blabla** dans notre société ? un

`<a class="renforce" href="http://labas.fr">lien</a>`  
intéressant

</p>

```
.reinforce {
  color: red
}
```

```
a.renforce {
  font-style: oblique;
}
```



# *Hiérarchie de balises*

- Sélection en fonction d'une relation parent - descendant :
  - instance
  - balise
  - classe
- Syntaxe
  - éléments séparés par un espace

# Exemple : le menu

```
<body>
  <p id="menu">
    <dl>
      <dt><a href="#">menu 1</a></dt>
      <dt><a href="#">menu 2</a></dt>
      <dt><a href="#">menu 3</a></dt>
      <dt><a href="#">menu 4</a></dt>
    </dl>
  </p>
</body>
```

- Problème :
  - comment spécifier la présentation des liens du menu

# Solution 1

- Créer une classe correspondante

```
<body>
  <p id="menu">
    <dl>
      <dt><a class='lienMenu' href="#">menu 1</a></dt>
      <dt><a class='lienMenu' href="#">menu 2</a></dt>
      <dt><a class='lienMenu' href="#">menu 3</a></dt>
      <dt><a class='lienMenu' href="#">menu 4</a></dt>
    </dl>
  </p>
</body>
```

```
.lienMenu
{
  color: green;
}
```

# Solution 2

- Utilisation de la structure : évite de créer une classe supplémentaire

```
<body>
  <p id="menu">
    <dl>
      <dt><a href="#">menu 1</a></dt>
      <dt><a href="#">menu 2</a></dt>
      <dt><a href="#">menu 3</a></dt>
      <dt><a href="#">menu 4</a></dt>
    </dl>
  </p>
</body>
```

```
#menu dl dt a
{
  color: green;
}
```

ou

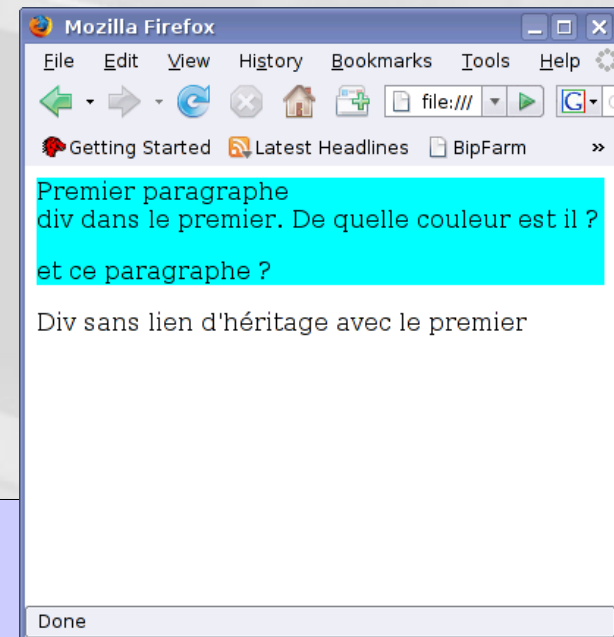
```
#menu a
{
  color: green;
}
```

# Cascade Stylesheets...

- Propagation des propriétés aux enfants

```
<div class="azure">  
  Premier paragraphe  
  <div>  
    div dans le premier. De quelle couleur est il ?  
  </div>  
  <p>  
    et ce paragraphe ?  
  </p>  
</div>  
<div>  
  Div sans lien d'héritage avec le premier  
</div>
```

```
.azure  
{  
  background-color : aqua;  
}
```





# Dimensions

# Dimensions

- Éléments concernés :
  - polices, images, bordures, boîtes etc..
- Unités :
  - Tailles fixes
  - Tailles relatives



# Tailles Fixes

- Le point (pt)
  - 1 pt = 0.35mm
- Le pica (pc)
  - 1pc = 4.22 mm
- Le centimètre (cm)
- Le millimètre (mm)
- Le pouce (in)
  - 1 in = 2.54 cm
- Le pixel (px)

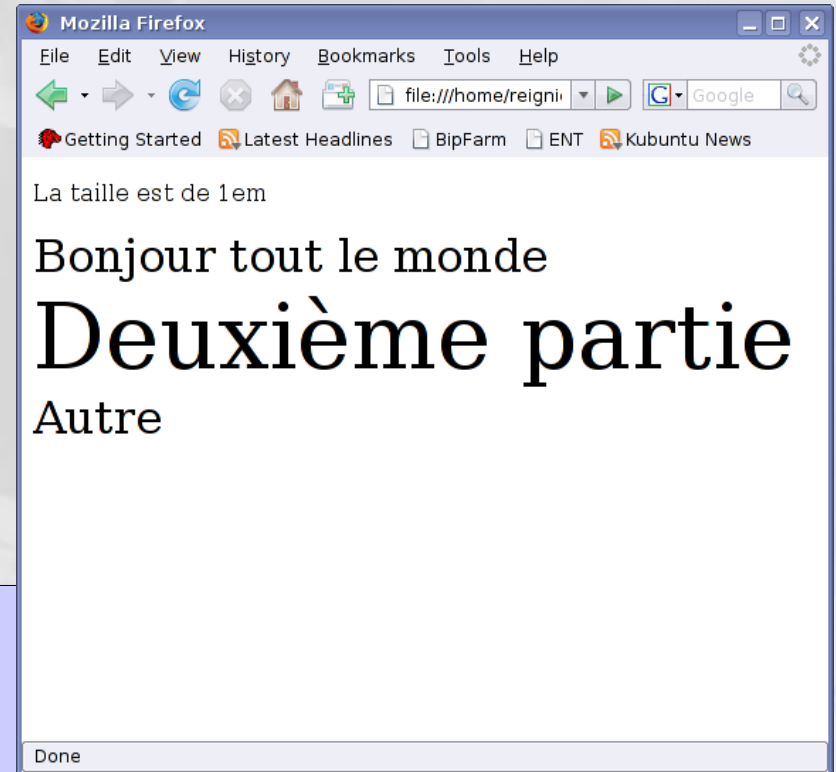
# Tailles relatives

- Le cadratin : em
  - Taille de la police de référence
  - Police de référence
    - Sommet de la hiérarchie : taille de la police par défaut du navigateur
    - Police de référence d'un conteneur (ex : div) = taille de la police du conteneur parent.

# Exemple

```
<body>
  <p>
    La taille est de 1em
  </p>
  <div id="div1">
    Bonjour tout le monde
    <div id="div2">
      Deuxième partie
    </div>
    <div>
      Autre
    </div>
  </div>
</body>
```

```
#div1, #div2
{
  font-size : 2em;
}
```



# Exemple 2

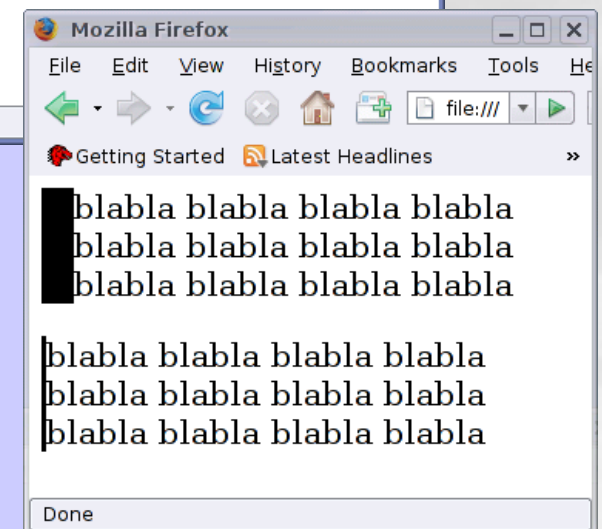
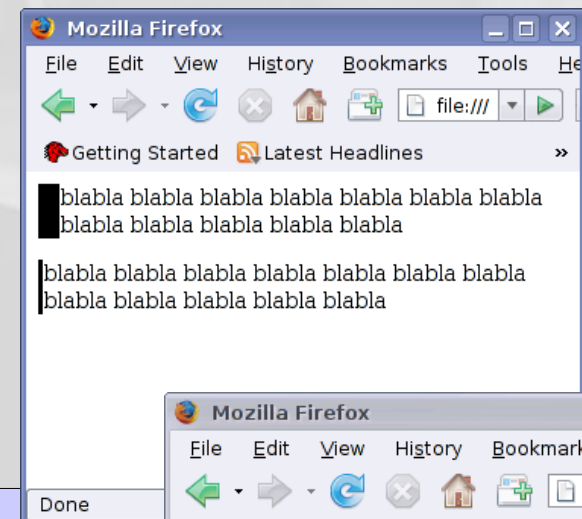
```
<body>
  <div id="bordureRelative">
    blabla blabla blabla blabla blabla blabla
    blabla blabla blabla blabla blabla blabla
  </div>
```

```

  <p/>
  <div id="bordureFixe">
    blabla blabla blabla blabla blabla blabla
    blabla blabla blabla blabla blabla blabla
  </div>
</body>
```

```
#bordureFixe
{
  border-left-style: solid;
  border-left-width: 3px;
}
```

```
#bordureRelative
{
  border-left-style: solid;
  border-left-width: 1em;
}
```



# Tailles relatives

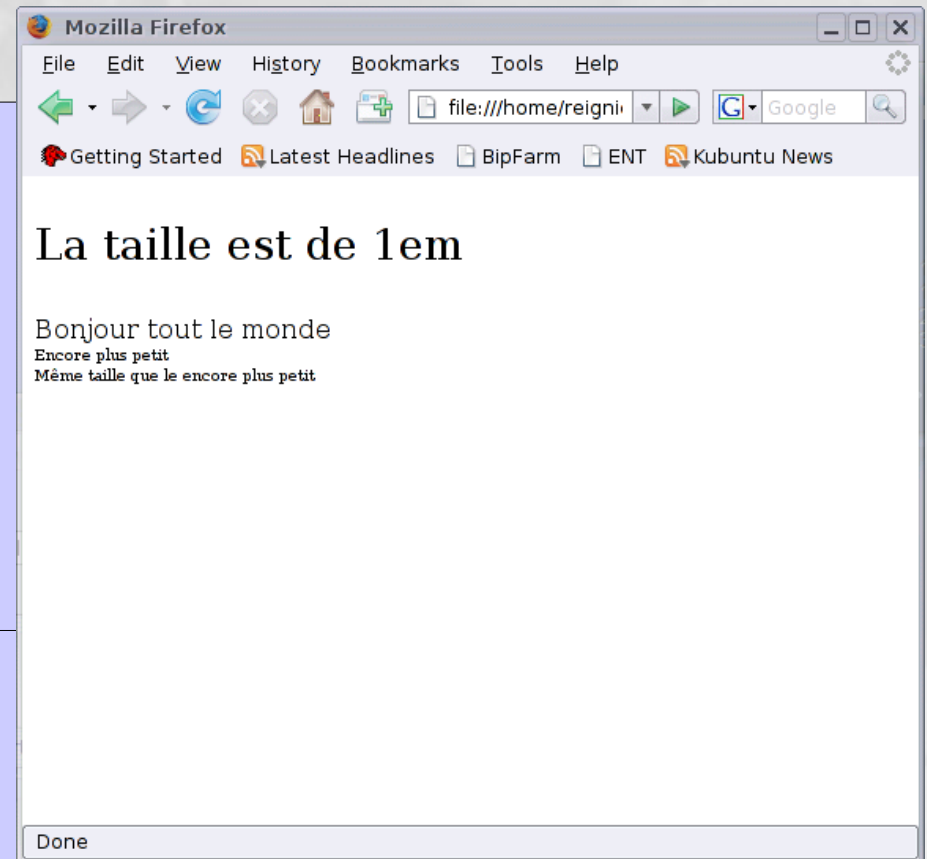
- Le pourcentage (%)
  - Disponible uniquement pour les **polices**
  - Même principe que le cadratin :
    - Pourcentage de la taille de référence

# Exemple

```
<body>
  <p>
    La taille est de 1em
  </p>
  <div id="div1">
    Bonjour tout le monde
    <div id="div2">
      Encore plus petit
    </div>
    <div id="divEm">
      Même taille que le encore plus petit
    </div>
  </div>
</body>
```

```
#div1, #div2
{
  font-size : 60% ;
}
```

```
#divEm
{
  font-size : 0.6em ;
}
```



# Tailles relatives

- Mots clés réservés :
  - Uniquement pour les **polices**
  - xx-small, x-small, small
  - medium
  - large, x-large, xx-large



# Les caractères



# *Propriétés des caractères*

- Police
- Taille (cf chapitre précédent)
- Couleur
- Gras
- Casse
- Style
- Decoration

# Police

- Deux grandes familles :
  - Sérif :
    - Forme « complexe » des caractères (i, h r etc)
    - Eviter pour les présentations écran (aliasing etc)
  - Sans Sérif :
    - Forme simple des caractères
    - Pour les présentations écran

# *Police (suite)*

- Choix de la police :
  - font-family
- Attention :
  - La police choisie doit être disponible sur l'ordinateur où sera visualisé le document
    - Eviter les polices exotiques
- Sans Sérif « classique » :
  - Arial, Verdana

# Couleur

- Choix de la couleur
  - color
  - Mots clés réservés : red, blue, black etc
  - Codage hexadécimal :
    - #rrvvbb
    - Exemple : rouge = #ff0000
- Couleur de fond :
  - background-color

# Graisse

- Epaisseur de la police
  - font-weight
- Mots clés réservés :
  - normal, bold, lighter, bolder

# Casse

- text-transform:
  - capitalize : première lettre en majuscule
  - lowercase : tout le texte en minuscule
  - uppercase : tout le texte en majuscule
  - none : aucune modification du texte. Permet d'annuler une modification positionnée par une balise parent (cf cascade).

# Style

- font-style
  - normal : police droite
  - italic.
  - oblique (type de police particulier)

# Décoration

- text-decoration
  - underline : souligné
  - overline : surligné
  - line-through : barré
  - blink : clignotant
- Plusieurs valeurs peuvent être spécifiée :
  - text-decoration: underline overline;



# *Positionnement des caractères*

- Interlignage
- crénage
- espace entre les mots
- justification

# *Interlignage*

- Espace entre deux lignes (paragraphe)
- line-height
  - Généralement exprimée en cadratin (1.2 em)
  - Utilisation du mot clé : normal

# *Crénage et espace entre mots*

- Distance entre deux caractères
  - letter-spacing
  - positive ou négative (rapproche les lettres)
- Distance entre les mots
  - word-spacing
  - positive ou négative

# *Justification*

- text-align
  - left : alignement à gauche
  - right : alignement à droite
  - center : centré
  - justify : justifié



# Bordures

# Bordures

- Possibilité d'entourer
  - Les éléments de type bloc (div, h, p etc)
  - Certains éléments en ligne (img)
- Caractéristiques
  - style
  - épaisseur
  - couleur
- Appliquable à tout le bord ou à chaque côté

# Style

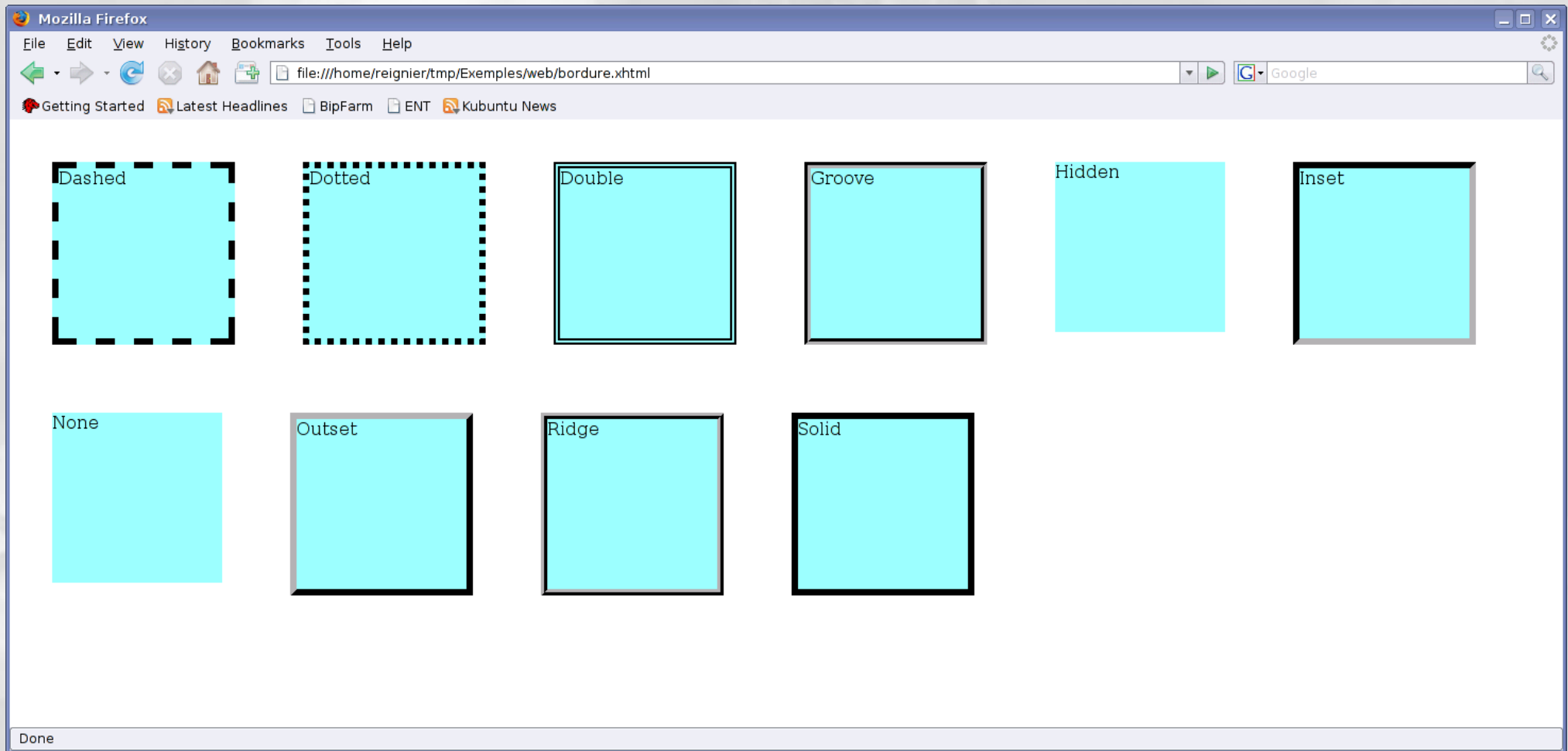
- border-style: <val> (tous les côtés)
- border-style: <val1> <val2>
  - val1 = côtés horizontaux
  - val2 = côtés verticaux

# Style (suite)

- border-style: <val1> <val2> <val3>
  - val1 = côté horizontal haut
  - val2 = côtés verticaux
  - val3 = côté horizontal bas
- Possibilité de spécifier chaque côté
  - border-left-style, border-right-style
  - border-top-style, border-bottom-style
- Valeurs possibles:
  - dashed, dotted, groove, hidden, inset, none, outset, ridge, solid



# Style



# *Epaisseur*

- border-width (cf left, right, bottom, top)
- Valeur numérique
- Mots clés :
  - thin : mince
  - medium
  - thick

# Couleur

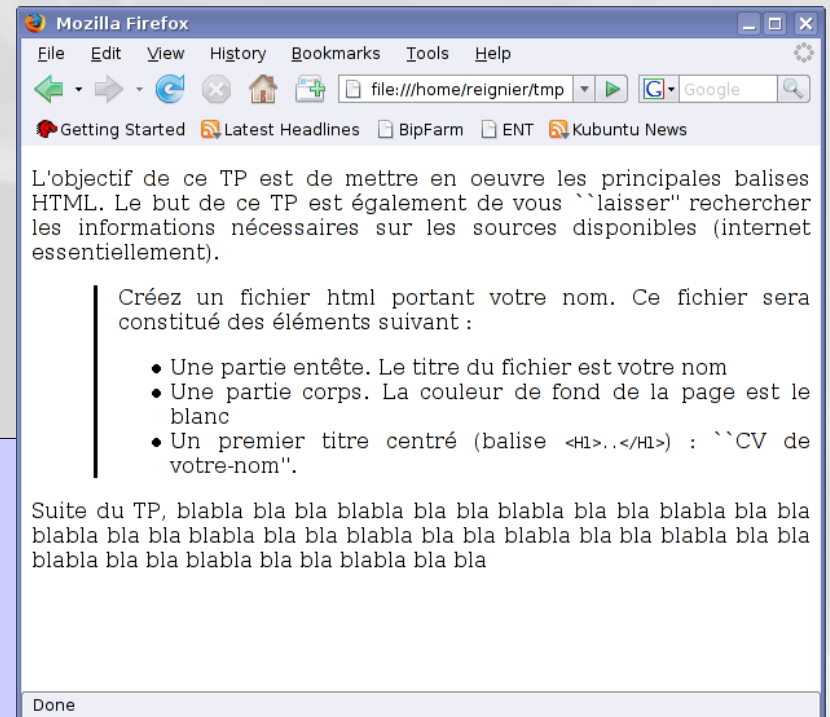
- border-color (cf left, right, width, height)
- Valeur
  - valeur hexadécimale
  - mot clé

# Exemple : un trait vertical

```
<p>
  L'objectif de ce TP est de mettre ....
</p>
<p class="question">
  Créez un fichier html portant votre nom. ...
</p>
<p>
  Suite du TP, blabla bla bla blabla p
</p>
```

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

```
.question
{
  border-left-style : solid ;
  border-left-width : 3px ;
  margin-left : 3em ;
  padding-left : 1em ;
}
```





# Image, couleur de fond

# Définition

- Association :
  - image ou couleur de fond – élément HTML
- Propriétés
  - URL ou couleur
  - Positionnement
  - Répétition
  - Attachement

# URL et couleur

- URL :
  - background-image: url(<URL>);
  - Attention : pas de ' autour de <URL>
- Couleur de fond
  - background-color: <valeur>

# Positionnement

- background-position:
- Valeur :
  - position horizontal / bord gauche
    - Mot clé (right, left, center) ou pourcentage
  - Si seconde valeur, position verticale / sommet
    - Mot clé (top, bottom, center) ou pourcentage



# Répétition

- background-repeat:
- Valeurs:
  - repeat-x : répétition horizontale
  - repeat-y : répétition verticale
  - repeat : répétition sur les deux axes (valeur par défaut)
  - no-repeat : pas de répétition.

# Attachement

- Par défaut :
  - Image attachée à l'élément
  - Se déplace avec l'élément lors d'un scroll
  - `background-attachement: scroll;`
- Attacher une image à la fenêtre :
  - `background-attachement: fixed;`

# Exemple

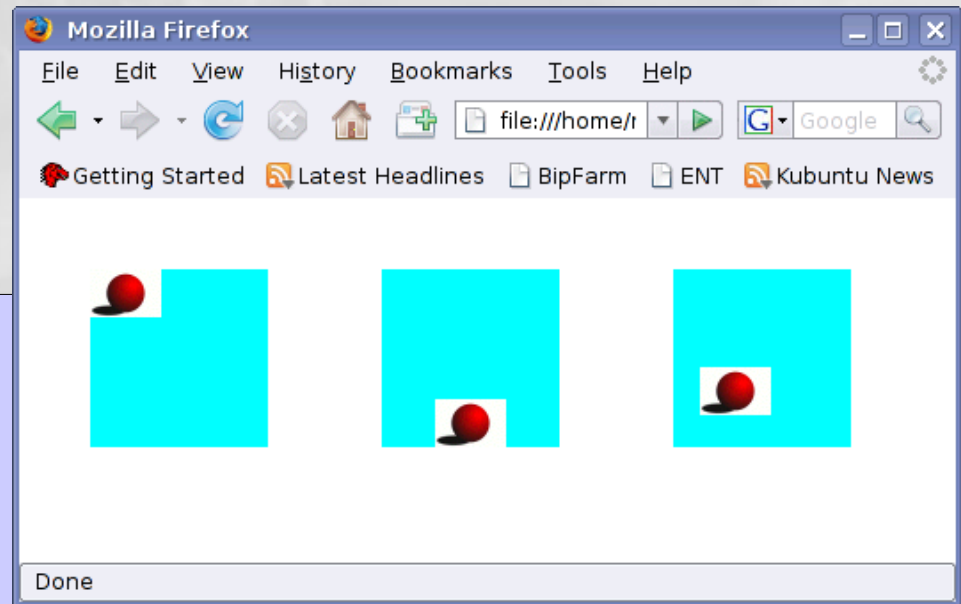
```
<body>
  <div id="div1"/>
  <div id="div2"/>
  <div id="div3"/>
</body>
```

```
div
{
  background-color : aqua;
  width: 100px;
  height: 100px;
  background-image : url(fond.gif) ;
  float: left;
  background-repeat: no-repeat;
}

#div1 {}

#div2 { background-position : center bottom ; }

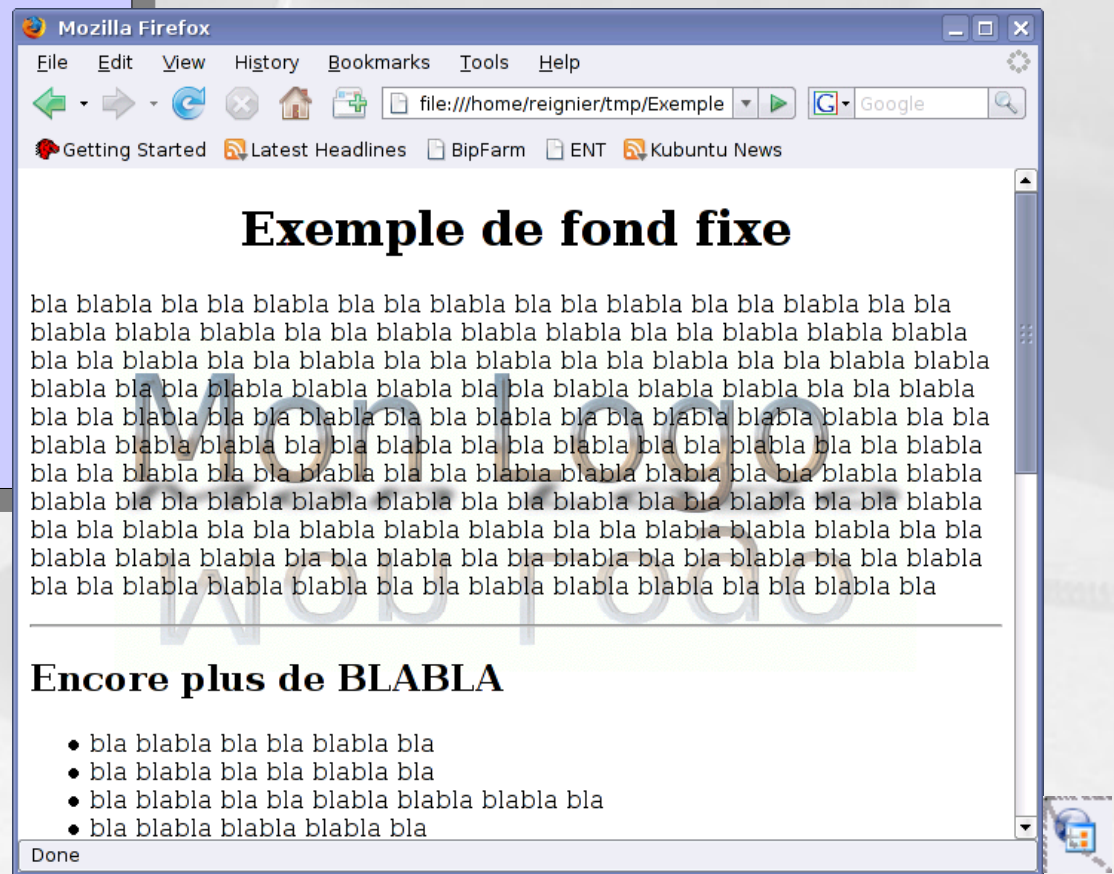
#div3 { background-position : 25% 75% ; }
```



# Exemple 2

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

body
{
  background-image : url(logo.gif) ;
  background-repeat : no-repeat;
  background-attachment: fixed;
  background-position : center center ;
}
```



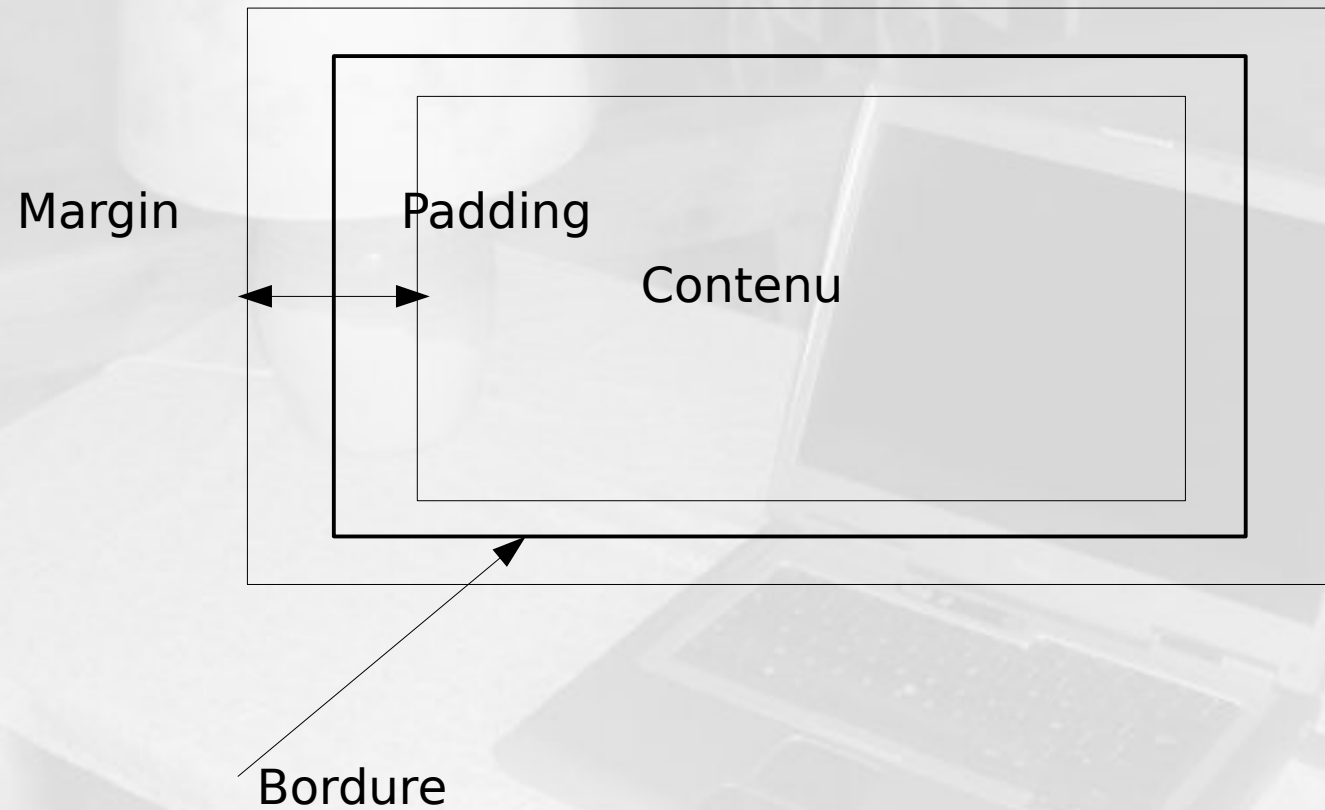


# Positionnement

# Boîtes

- Chaque élément HTML est une boîte
- Dimension de la boîte :
  - Contrôlable pour les éléments de type bloc
  - Fixées par le contenu pour les éléments de type en ligne

# Boîtes



# Boîtes

- div :
  - margin = padding = 0
- Autres balises :
  - Valeur par défaut fonction des navigateurs
  - Fixer les valeurs pour un rendu indépendant du navigateur



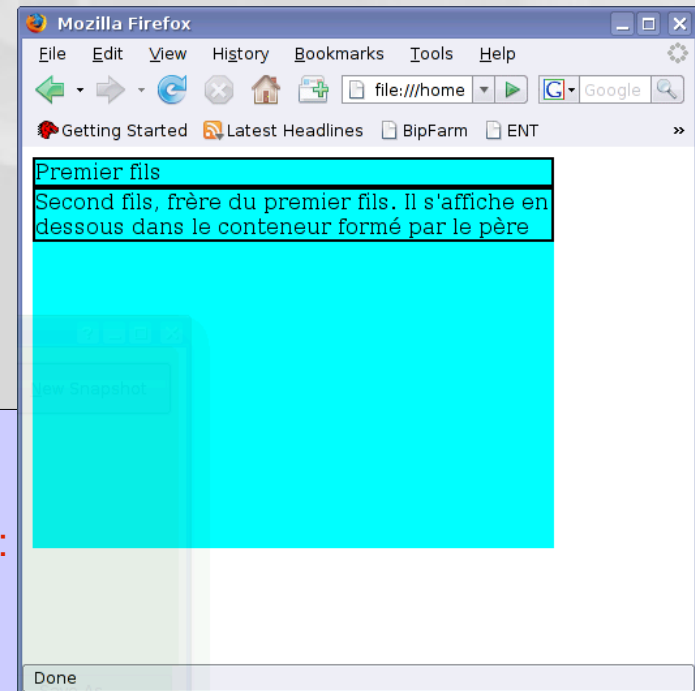
- Par défaut :
  - Deux balises *bloc* frères s'affichent l'une en dessous de l'autre
  - Deux balises *ligne* frères s'affichent l'une à côté de l'autre
  - Un balise père est un conteneur pour ses balises filles. Les balises filles s'affichent dans le conteneur.

# Exemple

```
<body>
  <div id="pere">
    <div>
      Premier fils
    </div>
    <div>
      Second fils, frère du premier fils.
      Il s'affiche en dessous dans le conteneur
      formé par le père
    </div>
  </div>
</body>
```

```
#pere
{
  background-color :
  width : 400px ;
  height : 300px ;
}
```

```
#pere div
{
  border-style : solid;
  border-width: 2px;
}
```

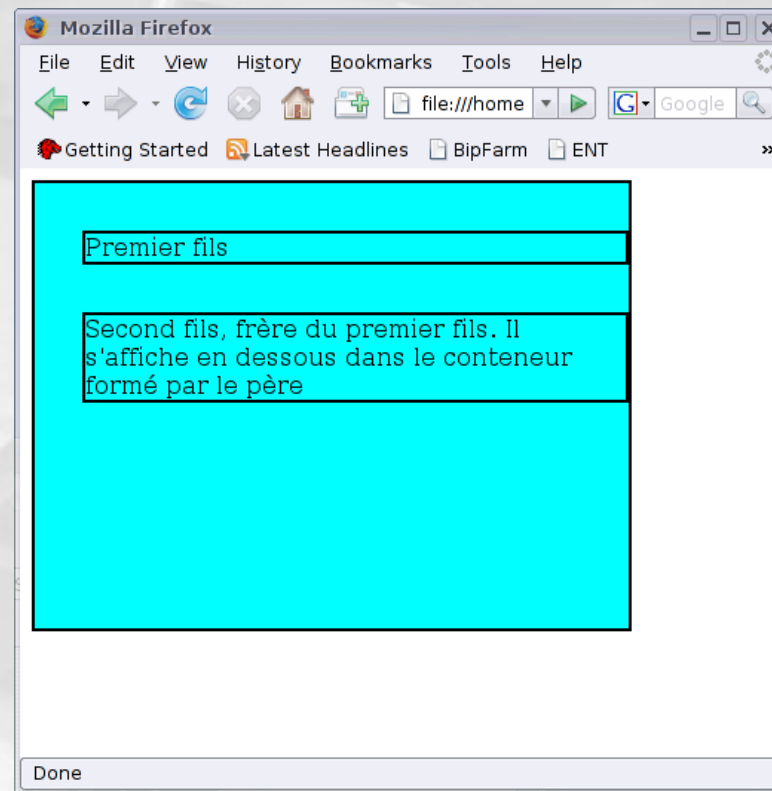


# Remarque

- Exemple précédent
  - Pas de contrainte de taille pour les fils
  - Hauteur fixée par le contenu
  - Largeur = largeur du conteneur
  - Positionnement dans le conteneur
    - Flux courant
- Ajustement de la position :
  - Marges

# Exemple : suite

```
#pere div
{
border-style : solid;
border-width: 2px;
margin-left : 2em;
margin-top: 2em;
}
```



# Positionnement

- Au delà du flux
- Trois type de positionnement :
  - positionnement relatif
  - positionnement fixe ou absolu
  - positionnement flottant

# *Positionnement relatif*

- Décaler un élément dans le flux courant
- Relatif par rapport à sa position « normale »
  - calculée en fonction du flux et des marges
- Déclaration :
  - `position: relative`

# *Positionnement relatif*

- left: décalage par rapport au bord gauche
- right: décalage par rapport au bord droit
- bottom: décalage par rapport au bas
  - positif => « remonte » dans le flux.
- top: décalage par rapport au haut
  - positif => « descend » dans le flux

# Exemple

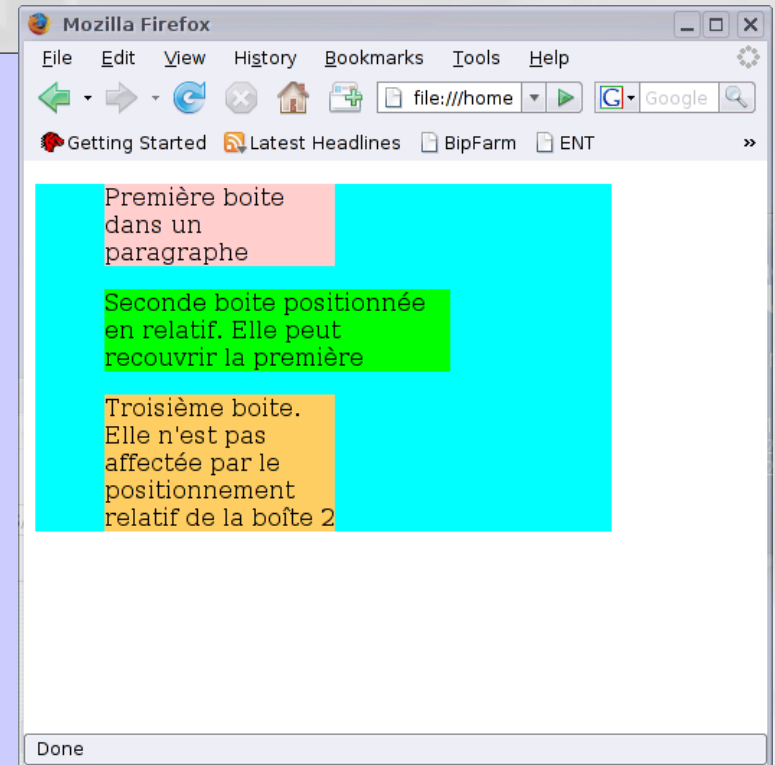
```
<p id="conteneur">
  <p id="boite1">
    Première boîte dans u
  </p>
  <p id="boite2">
    Seconde boîte positio
    Elle peut recouvrir la p
  </p>
  <p id="boite3">
    Troisième boîte. Elle r
    pas affectée par le po
    relatif de la boîte 2
  </p>
</p>
```

```
#conteneur {
  background-color : aqua;
  width: 400px;
}

#boite1{
  background-color : #ffcccc;
  width : 10em ;
  margin-left: 3em;
}

#boite2{
  background-color: lime;
  margin-left: 3em;
  width: 15em;
}

#boite3
{
  background-color: #ffcc66;
  margin-left: 3em;
  width: 10em
}
```





# Exemple

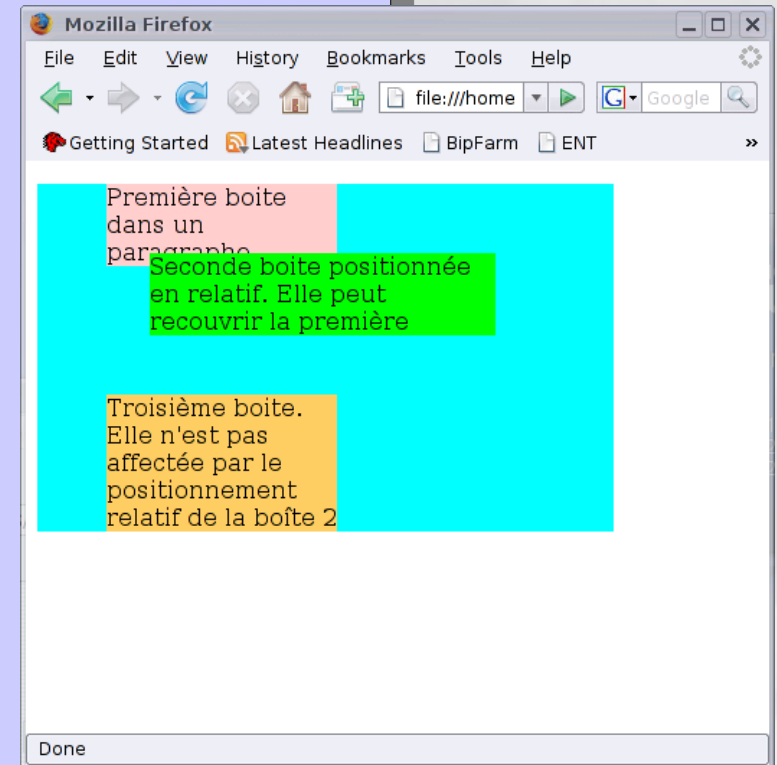
```
<p id="conteneur">
  <p id="boite1">
    Première boîte dans u
  </p>
  <p id="boite2">
    Seconde boîte positio
    Elle peut recouvrir la p
  </p>
  <p id="boite3">
    Troisième boîte. Elle r
    pas affectée par le po
    relatif de la boîte 2
  </p>
</p>
```

```
#conteneur {
  background-color : aqua;
  width: 400px;
}

#boite1{
  background-color : #ffcccc;
  width : 10em ;
  margin-left: 3em;
}

#boite2{
  background-color: lime;
  margin-left: 3em;
  width: 15em;
  position: relative;
  left: 30px ;
  bottom: 25px;
}

#boite3
{
  background-color: #ffcc66;
  margin-left: 3em;
  width: 10em
}
```



# *Positionnement absolu ou fixe*

- Sortie d'un élément du flux
- Ne participe plus à la position de ses frères
- Déclaration :
  - `position: absolute;`
  - `position: fixed; // indépendant du scroll`
  - `left` ou `right`, `bottom` ou `fixed`

# Principe

- Interprétation des coordonnées
  - Repère = premier parent positionné
    - soit en relatif
    - soit en fixe
  - Body est toujours considéré comme positionné

# Exemple

```
<body>
  <p>
    Voici un exemple de positionnement
    absolu.
  </p>
  <p id="conteneur">
    <p id="boite1">
      Première boîte dans un paragraphe.
      Elle n'est pas positionnée
    </p>
    <p id="boite2">
      Seconde boîte. Elle est une fille du conteneur bleu
    </p>
    <p id="boite3">
      Troisième boîte.
    </p>
  </p>
</body>
```

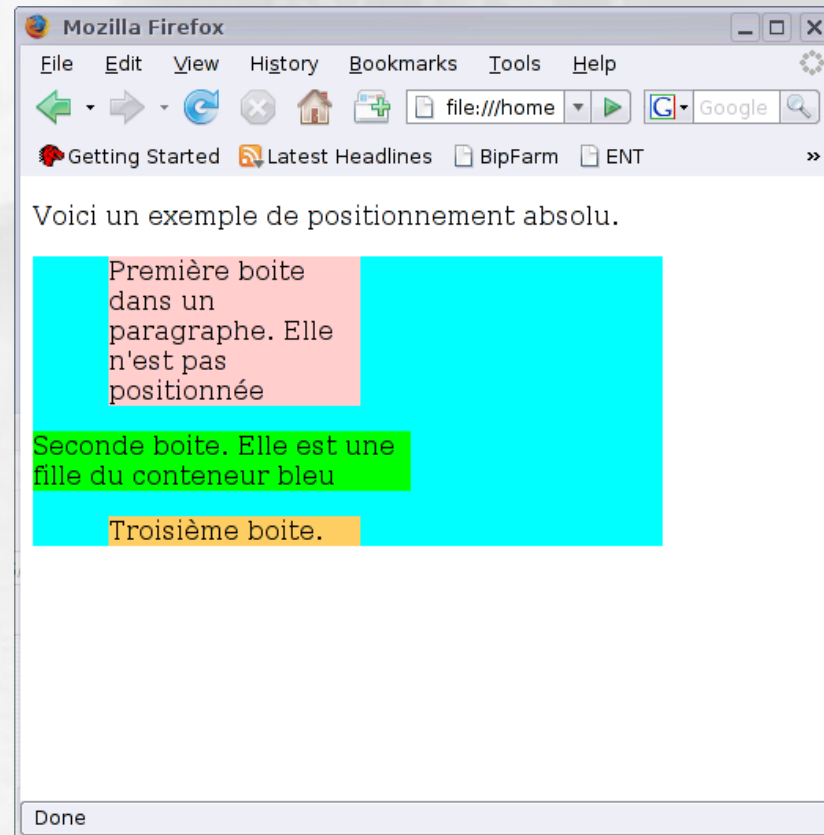
```
#conteneur{
  background-color : aqua;
  width: 400px;
}

#boite1{
  background-color : #ffcccc;
  width : 10em ;
  margin-left: 3em;
}

#boite2{
  background-color: lime;
  width: 15em;
}

#boite3{
  background-color: #ffcc66;
  margin-left: 3em;
  width: 10em
}
```

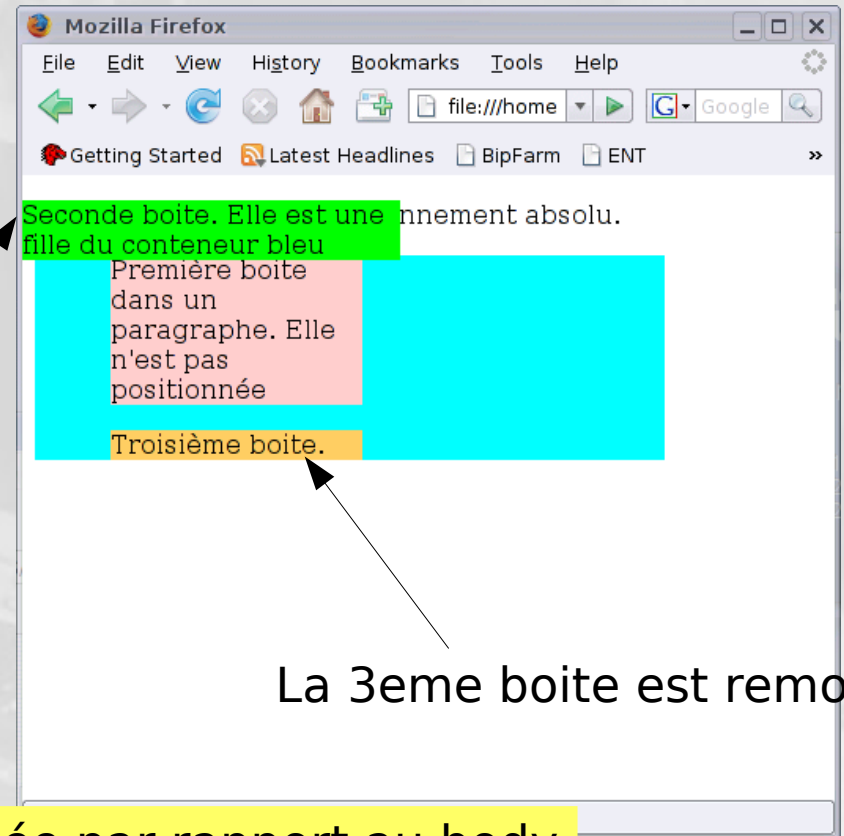
# Exemple (suite)



# Exemple (suite)

```
p{  
  margin: 0;  
}  
  
#conteneur{  
  background-color : aqua;  
  width: 400px;  
}  
  
#boite1{  
  background-color : #ffcccc;  
  width : 10em ;  
  margin-left: 3em;  
}  
  
#boite2{  
  background-color: lime;  
  width: 15em;  
  position: absolute;  
  left: 0px;  
  top: 0px;  
}  
  
...
```

La seconde boîte est positionnée par rapport au body  
Elle est sortie de son conteneur



La 3eme boîte est remontée

# Exemple (suite)

```
p{
  margin: 0;
}

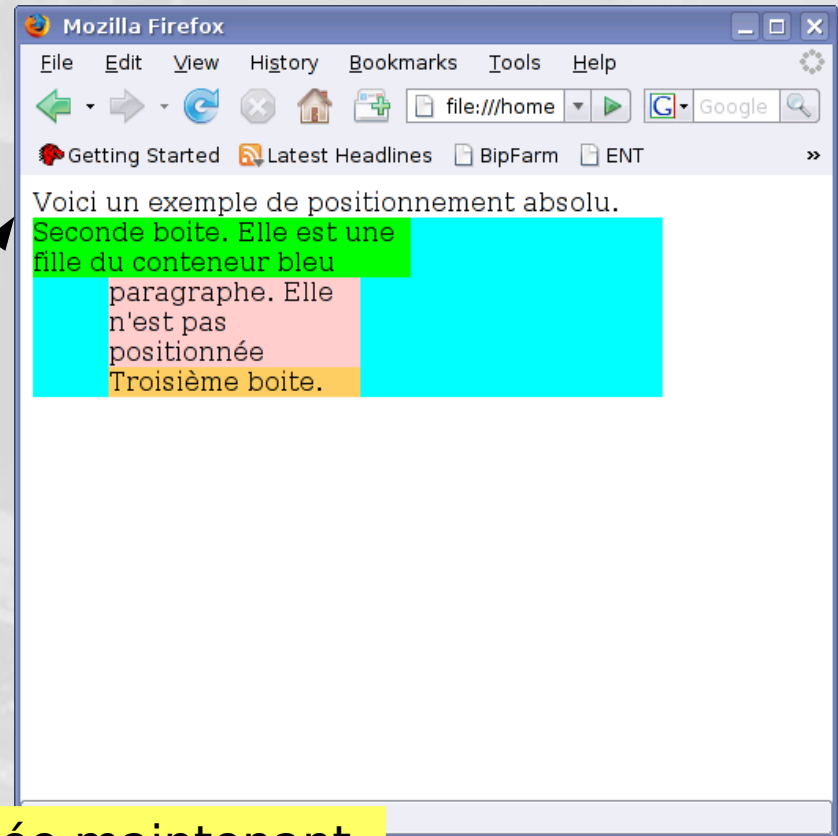
#conteneur{
  background-color : aqua;
  width: 400px;
  position: relative;
}

#boite1{
  background-color : #ffcccc;
  width : 10em ;
  margin-left: 3em;
}

#boite2{
  background-color: lime;
  width: 15em;
  position: absolute;
  left: 0px;
  top: 0px;
}

...
```

La seconde boîte est positionnée maintenant par rapport au conteneur.



# Positionnement flottant

- Sortie d'un élément du flux
- Déclaration :
  - float: left
  - float: right
- L'élément est positionné à gauche (ou à droite) du bloc le contenant
- Les éléments flottant « s'empilent »
- Les autres éléments les « entourent »



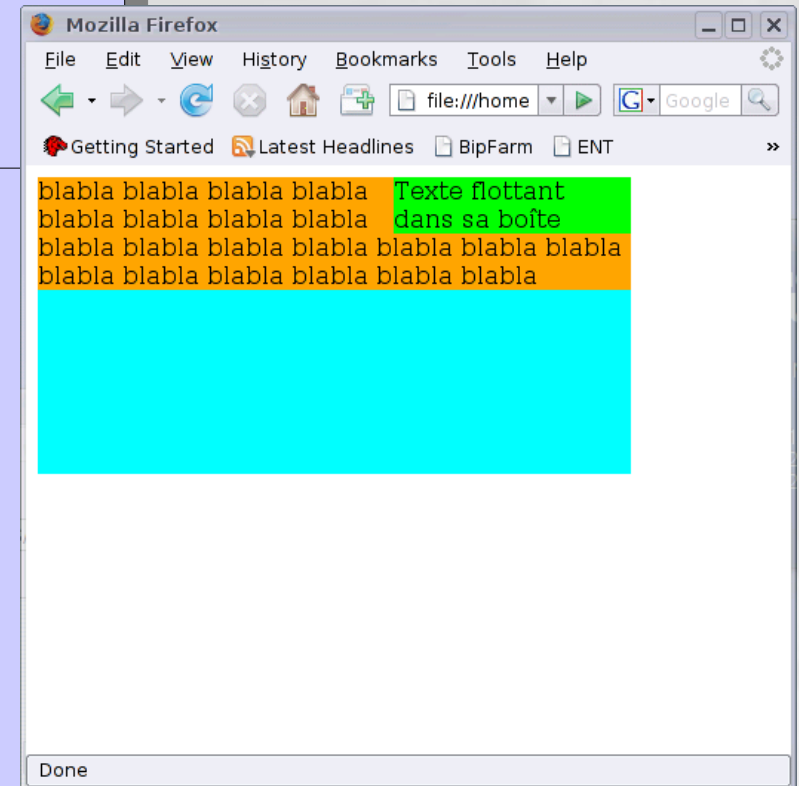
# Exemple

```
<body>
  <p id="conteneur">
    <p id="boiteFlottante">
      Texte flottant dans sa boîte
    </p>
    <p id="boiteNormale">
      blabla blabla blabla
      blabla blabla blabla
      blabla blabla blabla
      blabla blabla blabla
      blabla blabla blabla
      blabla blabla blabla
    </p>
  </p>
</body>
```

```
#conteneur{
  background-color : aqua;
  width: 400px;
  height : 200px ;
}

#boiteFlottante{
  float: right;
  width: 10em;
  background-color : lime;
}

#boiteNormale{
  background-color : orange;
}
```



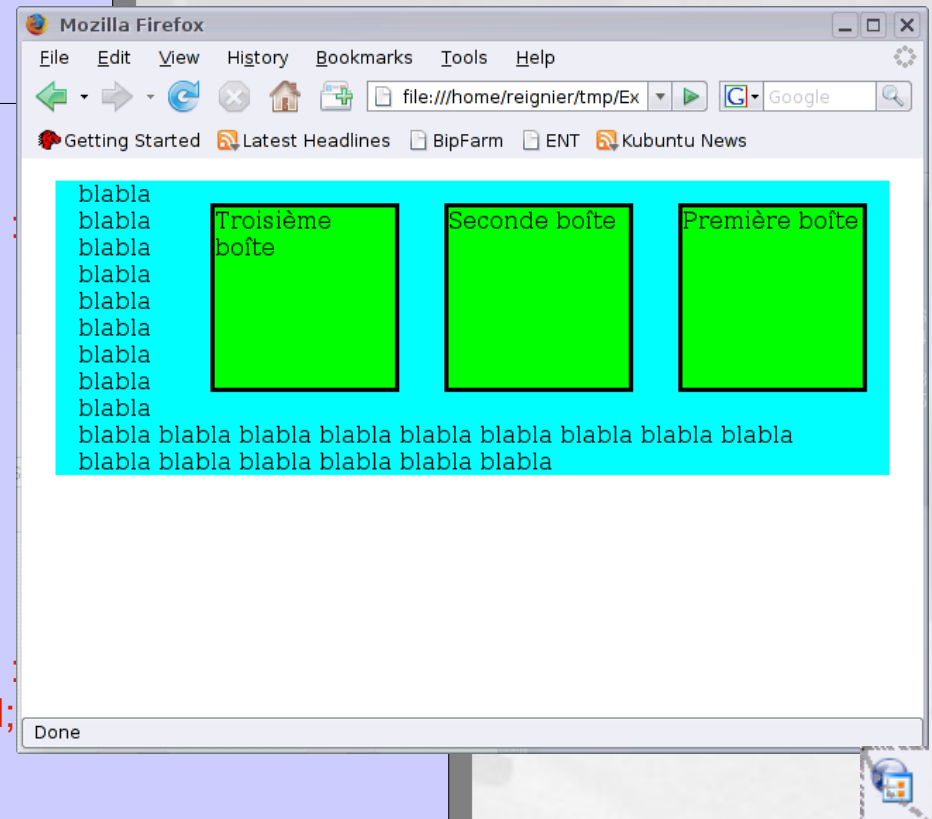
## Exemple : empilement de float

```
<body>
  <p id="conteneur">
    <p class="boiteFlottante">
      Première boîte
    </p>
    <p class="boiteFlottante">
      Seconde boîte
    </p>
    <p class="boiteFlottante">
      Troisième boîte
    </p>
    <p>
      blabla blabla blabla blabla
      blabla blabla blabla blabla
      ....
    </p>
  </p>
</body>
```

```
#conteneur
{
    background-color :
}

p
{
    margin: 1em;
}

.boiteFlottante
{
    background-color :
    border-style: solid;
    width: 8em;
    height:8em;
    float: right;
}
```



# Profondeur

- Par défaut :
  - Tout élément positionné s'affiche par dessus tous ses prédécesseurs dans le conteneur.
- Changement de l'ordre de recouvrement
  - Profondeur
  - z-index:
  - Ordre de recouvrement : z croissant

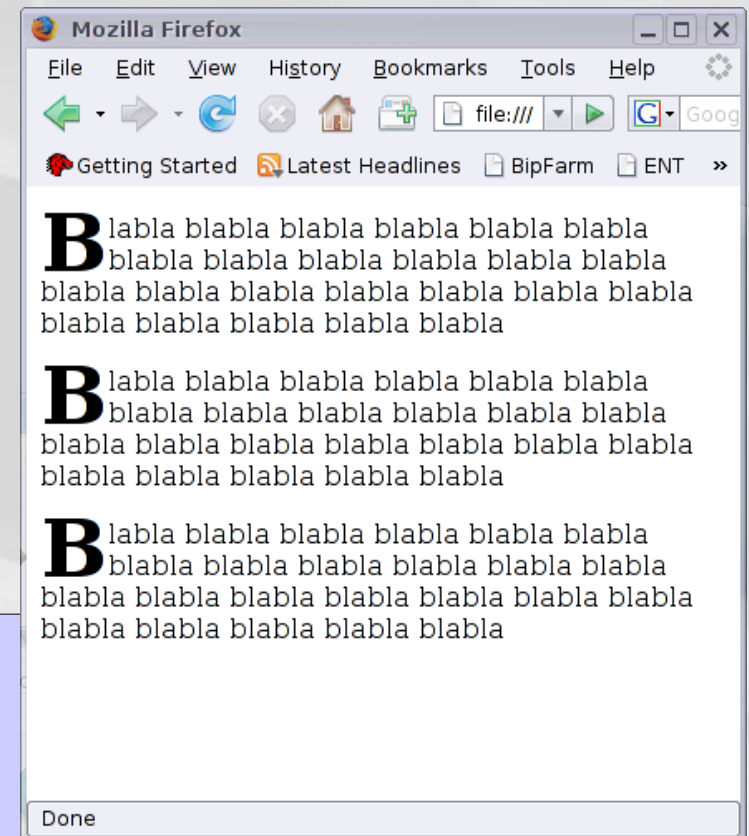


# Exemples

# Lettrine

```
<body>
  <p>
    blabla blabla blabla blabla
    blabla blabla blabla blabla
  ...
  </p>
  <p>
    blabla blabla blabla blabla
    blabla blabla blabla blabla
  ...
  </p>
  <p>
    blabla blabla blabla blabla
    blabla blabla blabla blabla
  ...
  </p>
</body>
```

```
p:first-letter
{
  font-weight: bold;
  text-transform : capitalize ;
  font-size : 3em ;
  float: left;
}
```



# Menu graphique

- Menu avec image de fond
- Eclaircissement de l'image
  - Lors du passage de la souris sur une entrée



# Principe



# Menu graphique

```
<body>
  <p id="menu">
    <ul>
      <li><a href="#">Menu 1</a></li>
      <li><a href="#">Menu 2</a></li>
      <li><a href="#">Menu 3</a></li>
      <li><a href="#">Menu 4</a></li>
    </ul>
  </p>
</body>
```

```
ul, li
{
  list-style-type : none ;
  margin: 0;
  padding: 0;
}
```

Suppression des puces



# Image de fond

- Dimension d'une entrée menu :
  - 400 pixels x 50 pixels
  - 4 entrées
  - Image de fond = 400 x 200 pixels
- Contrôle de la dimension :

```
#menu li a  
{  
  display :block ;  
  width: 400px;  
  height : 50px ;  
  text-decoration: none ;  
  color : #666666;  
}
```

On considère l'ancre  
comme un bloc (au lieu  
d'un élément en ligne)

# Image de fond

```
#menu ul
{
  background: url(fondMenu.jpg);
  text-align : center ;
  width: 400px
}
```

Image de fond « par défaut »

Association de la même image pour le survol de chaque entrée du menu

```
#menu li a:hover
{
  background-image : url(fondMenu.jpg) ;
  color: red;
}
```

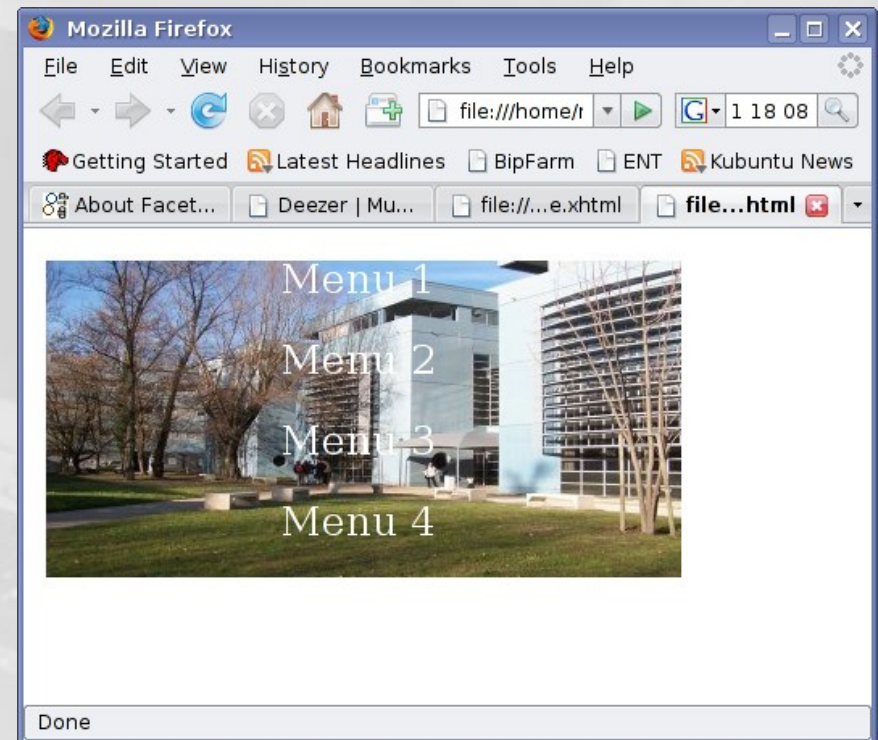
# Décalage pour chaque entrée

```
#menu1: hover
{
  background-position : 0px -200px;
}

#menu2: hover
{
  background-position : 0px -250px;
}

#menu3: hover
{
  background-position : 0px -300px;
}

#menu4: hover
{
  background-position : 0px -350px;
}
```



# *Bibliographie*

- CSS2 : Pratique du design web, Raphaël Goetter
- <http://www.csszengarden.com>
- <http://openweb.eu.org>
- <http://wiki.media-box.net/documentation/css>
- <http://www.w3.org/TR/1998/REC-CSS2-19980512/propidx.html>
- <http://www.w3schools.com/css/>