

# Programmation web I

---



Dr. Fatma Ellouze  
fatma.ellouze@iit.ens.tn

**1<sup>ère</sup> année GI**

# Plan

- Chapitre 1: Introduction à la programmation web
- Chapitre 2: Introduction au langage HTML5
- **Chapitre 3: Introduction au langage CSS3**
- Chapitre 4: Introduction au langage JavaScript

---

# Chapitre 3 : Cascading Style Sheets (CSS3)

---



# Objectifs

- Maîtriser les principes de base de CSS3
  - Lien entre HTML et CSS
  - Sélecteur et déclaration CSS
  - Classes vs identifiants
  - Utilité de l'utilisation d'une feuille CSS
  - Notion de cascade
  - Polices et familles de polices
- Maîtriser les sélecteurs avancés
- Maîtriser les styles de formatage du texte & les styles de traitement des paragraphes
- Maîtriser les styles de bordures
- Maîtriser le modèle des boîtes
- Maîtriser le positionnement en CSS3
- Maîtriser les Media Queries de CSS3



# Présentation vs contenu

- Le **contenu** désigne les mots et les images d'un document HTML.
- La **présentation** est liée aux **styles** et à l'aspect des mots et des images dans un document HTML.
- Le contenu est géré en HTML et le style en CSS.



# CSS : définition

- Les CSS, CascadingStyle Sheets (feuilles de styles en cascade), servent à **mettre en forme** les pages HTML.
- Comparable à la mise en forme des documents word : .docx et .dot
- CSS est une séquence de **règles**
- CSS3 est la **dernière** version du projet qui correspond à **HTML5**
- CSS3 est **compatible** avec les versions précédentes de CSS



# Où écrit-on le CSS ?

- On peut écrire du code en langage CSS à **trois endroits différents** :
  - directement dans les balises du fichier HTML *via* un **attribut style** (*méthode la moins recommandée*),
  - dans l'en-tête **<head>** du fichier HTML ;
  - dans un **fichier .css** (*méthode la plus recommandée*).



# Illustration

```
<HTML>

<HEAD>

  <link href = "style.css" rel = "stylesheet" <!-- Lien externe -->

  <style>
    h2{
      color: purple <!-- Style interne à la page -->
    }
  </style>

</HEAD>

<BODY>

  <h1> Niveau 1 : Programmation WEB </h1>

  <h2> Niveau 2 : Cours CSS </h2>

  <p style = "color: red" > 2LIG </p> <!-- Style interne à la balise -->

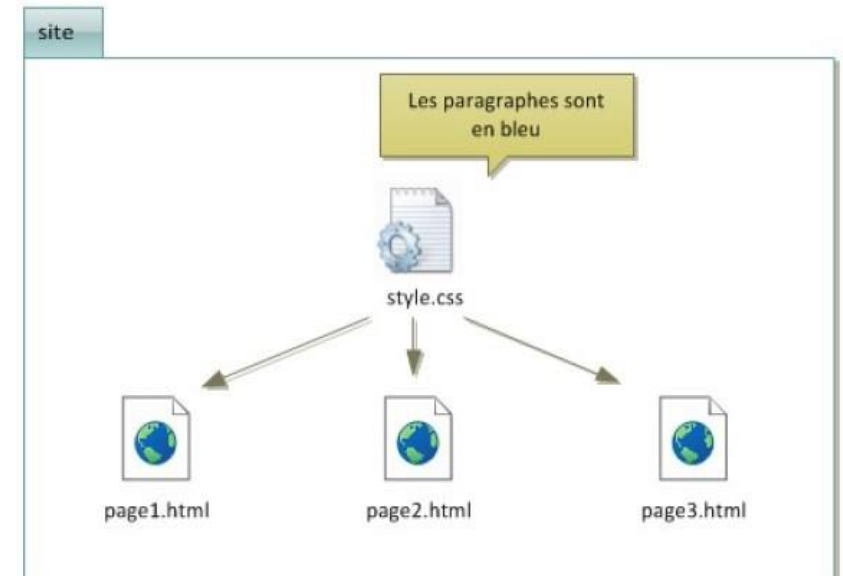
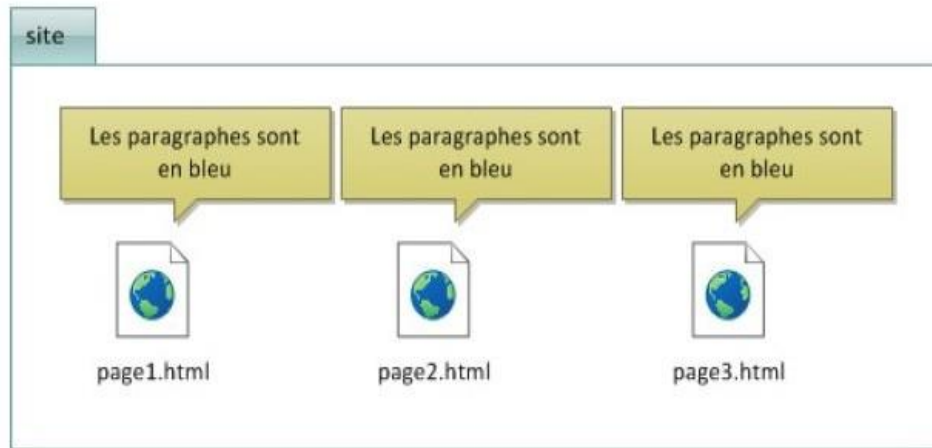
</BODY>

</HTML>
```



# Utilité de l'utilisation d'une feuille CSS

- L'objectif est de dissocier le contenu de la page de son apparence visuelle. Ceci permet:
  1. de ne pas répéter dans chaque page le même code de mise en forme
  2. de pouvoir changer l'apparence d'un site web complet en ne modifiant qu'un seul fichier
  3. de faciliter la lecture du code de la page



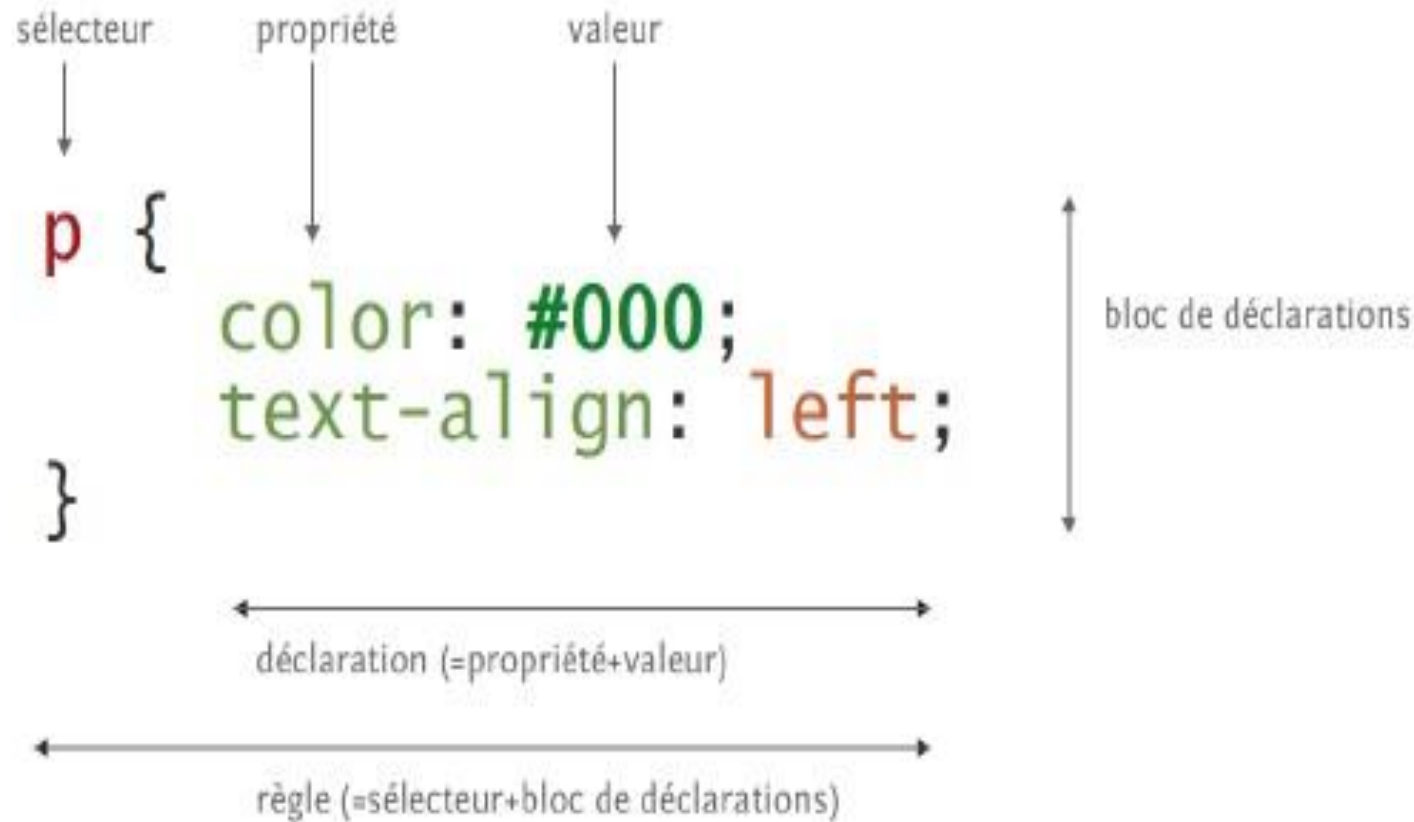


# Notion de cascades

- Les styles peuvent être déclarés à différents endroits, et selon ces endroits ils seront plus ou moins prioritaires.
- On obtient donc une ***cascade de styles***.
- Déclaration des styles dans une feuille de style **externe** : elle a le **moins de poids**
- Déclaration des styles en **interne**, dans l'en-tête de la page <head>. Les styles déclarés auront plus de poids que ceux de la feuille de style externe et donc l'emporteront en cas de conflits.
- Déclaration des **styles en attributs** des balises html. Ces styles **l'emporteront** sur tous les autres.



# La syntaxe d'une règle CSS



Le **sélecteur** est généralement l'élément HTML auquel vous appliquez un style.

La **déclaration** est le style d'un sélecteur spécifique. Une déclaration possède une propriété, qui est un attribut de style et une valeur.



# Sélecteur

- Un sélecteur peut être :
  - Une **balise**
  - L'attribut **class** (associé aux balises HTML)
  - L'attribut **id** (associé à une balise HTML)



# Priorité

- Priorité des sélecteurs
  - Si deux règles CSS différentes ciblent le même élément, c'est la **dernière déclarée** (la plus basse dans le code source) qui s'applique et qui écrase la précédente.

- Appliquer un style à plusieurs balises

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

```
h1 { color: blue; }
```



```
p, h1 { color : blue;}
```



# Priorité

- Nous pouvons également avoir pour une même balise différents styles, par exemple :
  - des paragraphes ayant comme couleur de texte le noir
  - et d'autres paragraphes dans la même page HTML ayant comme couleur de texte le bleu

## Que doit-on faire ?

- Pour résoudre le problème, on peut utiliser ces attributs spéciaux qui fonctionnent sur toutes les balises :
  - l'attribut **class**
  - l'attribut **id**



# Les attributs **class** et **id**

- Appliquer un style avec l'attribut **class** ou **id**

partie HTML    `<p class="style1">.....</p>`  
                  `<p > paragraphe 2 </p>`

partie CSS    **.style1**  
                  { color: green; }

partie HTML    `<p id="id1">.....</p>`

partie CSS    **#id1**  
                  { color: yellow; }



# Les balises universelles

- Appliquer un style à certains mots **qui ne sont pas entourés par des balises**
  - **<span> </span>** : une balise que l'on place au sein d'un paragraphe pour sélectionner certains mots  
C'est une balise inline : deux balises successives de **<span>** seront affichés l'un après l'autre horizontalement.
  - **<div> </div>** : une balise qui entoure un bloc de texte. elle crée un nouveau « bloc » dans la page et provoque obligatoirement un retour à la ligne  
C'est une balise bloc : deux balises successives de **<div>** seront affichés l'un après l'autre verticalement.





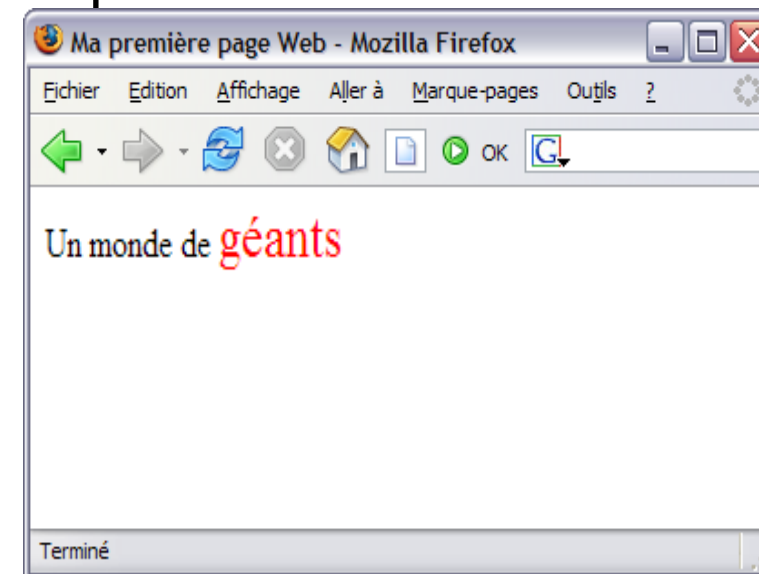
# Les balises span & div

```
<html>
  <head>
    <style>
      .element{
        font-size : x-large ;
        color      : red ;
      }
    </style>
  </head>

  <body>

    Un monde de <span class= "element"> géants  </span>

  </body>
</html>
```



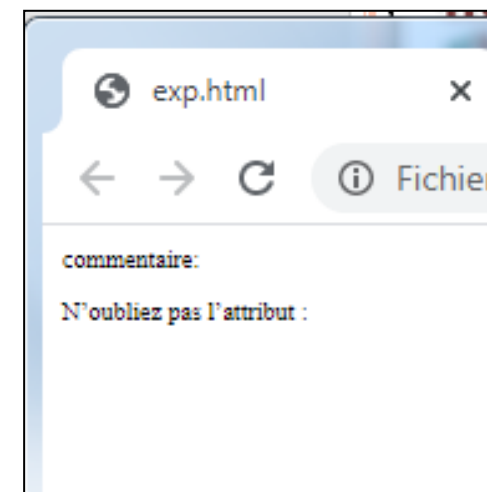
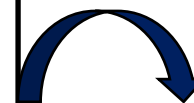


# Les balises span & div

```
<html>
  <head>
    <style>
      .zone
      {
        font-size : x-small ;
      }
    </style>
  </head>
  <body>
    <div class='zone'>

      <p>commentaire:</p>
      <p>N'oubliez pas l'attribut:</p>

    </div>
  </body>
</html>
```





# Les sélecteurs avancés



# Les sélecteurs avancés

- **\* : sélecteur universel**

Sélectionne toutes les balises sans exception.

```
*  
{ margin : 0 px; }
```

- **Balise contenue dans une autre**

- Sélectionne toutes les balises **<em>** situées à l'intérieur d'une balise **<p>**.

```
p em  
{ Color : red; }
```

```
<p>parapgraphe avec <em>texte important</em> </p>
```

NB : il n'y a pas de virgule entre les deux noms de balises **p** et **em**.



# Les sélecteurs avancés

- **Balise suivie d'une autre**
  - Sélectionne la première balise **<p>** (le premier paragraphe) située après un titre **<h3>**

```
h3 + p  
{ Color : red; }
```

```
<h3>Titre 3 </h3>  
<p>Paragraphe</p>
```

- **Balise qui possède un attribut**
  - Sélectionne tous les liens **<a>** qui possèdent un attribut title.

```
a[title]  
{ ..... }
```

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



# Les sélecteurs avancés

- **Balise qui possède un attribut et une valeur exacte**
  - Sélectionne tous les liens **<a>** qui possèdent un attribut **title** avec la valeur « Cliquez ici ».

```
a [title="Cliquez ici"]  
{ ..... }
```

```
<a href="http://site.com" title="Cliquez ici">
```

- **Balise qui possède un attribut et une valeur correspondante**
  - Sélectionne tous les liens **<a>** qui possèdent un attribut **title** l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position)..

```
a [title*="ici"]  
{ ..... }
```

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



# Apparence dynamique : style pour un état

- En CSS, on peut modifier l'apparence de certaines sections dynamiquement, après le chargement de la page, lorsque certains événements se produisent. On utilise pour cela les pseudo-formats.
  - **:hover** permet de changer l'apparence au survol (par exemple : **a:hover** pour modifier l'apparence des liens lorsque la souris pointe dessus).
  - **:active** modifie l'apparence des liens au moment du clic,
  - **:visited** lorsqu'un lien a déjà été visité.
  - **:focus** permet de modifier l'apparence d'un élément sélectionné.



# Apparence dynamique : style pour un état

- Les apparences dynamiques sont très utiles pour les liens mais aussi on peut les utiliser pour d'autres balises.

**Exemple:** Au survol sur un paragraphe, sa couleur devient bleu

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

**Exemple:** un clic long sur un texte italique change sa couleur en vert

```
em:active  
{  
    color: green;  
}
```





# Le formatage du texte



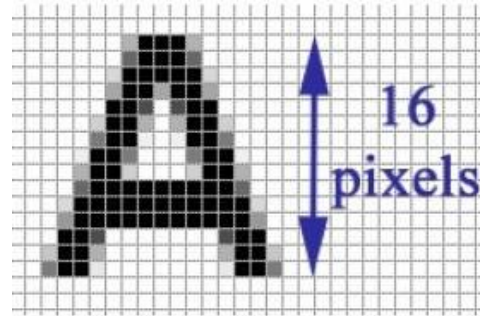
# Formatage du texte : La taille

- Pour modifier la taille du texte, on utilise la propriété CSS **font-size**:
  - Indiquer une **taille absolue** : en **pixels**, en **centimètres** ou **millimètres**. Cette méthode est très précise mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car on risque d'indiquer une taille trop petite pour certains lecteurs.
  - Indiquer une **taille relative** : en **pourcentage** ou en **em**. Cette technique a l'avantage d'être plus souple. Elle s'adapte plus facilement aux préférences de taille des visiteurs.



# Formatage du texte : La taille absolue

```
font-size: 16px;
```



Une lettre de 16 pixels de hauteur

```
p{  
  font-size: 14px; /* Paragraphes de 14 pixels */  
}  
h1  
{  
  font-size: 40px; /* Titres de 40 pixels */  
}
```

- Vous pouvez également définir des tailles en centimètres ou millimètres.
- Remplacez « px » par « cm » ou « mm ».
- Ces unités sont cependant **moins bien adaptées** aux écrans.



# Formatage du texte : La taille relative

C'est la méthode **recommandée**.

Il y a plusieurs moyens d'indiquer une valeur relative. Vous pouvez par exemple écrire la taille avec des mots en anglais :

- **xx-small** : minuscule ;
- **x-small** : très petit ;
- **small** : petit ;
- **medium** : moyen ;
- **large** : grand ;
- **x-large** : très grand ;
- **xx-large** : gigantesque

Le problème de cette méthode :  
c'est qu'il n'y a que **sept** tailles  
disponibles



# Formatage du texte : La taille relative

Il est possible d'indiquer la taille en « **em** ».

- Si vous écrivez **1em**, le texte a une taille normale.
- Si vous voulez grossir le texte, vous pouvez inscrire une valeur supérieure à 1, comme **1.3em**.
- Si vous voulez réduire le texte, inscrivez une valeur inférieure à 1, comme **0.8em**

```
p{  
  font-size: 0.8em;  
}  
h1{  
  font-size: 1.3em;  
}
```

- Selon le même principe, il est possible d'utiliser les pourcentages (80%, 130%, etc)



# Formatage du texte: La police

```
balise  
{  
font-family: police;  
}
```

- Arial ;
- Arial Black ;
- Comic Sans MS ;
- Courier New ;
- Georgia ;
- Impact ;
- Times New Roman ;
- Trebuchet MS ;
- Verdana.



# Remarque importante

- Toutes les polices de caractères n'existent pas sur tous les ordinateurs.
- Conséquence : si la police que vous avez choisie n'existe pas, une autre s'affichera ruinant probablement le design que vous espériez.
- Mac et PC ne possèdent pas toujours les mêmes noms de polices, ni même la même résolution.
- Conséquence : bien vérifier les équivalences entre Mac et PC Tous les navigateurs ne gèrent pas le rendu des polices de la même façon.



# Polices génériques

- Il existe 5 familles de polices de caractères dites "génériques" :
  - Serif
  - Sans-serif
  - Monospace
  - Cursive
  - Fantasy





# Polices génériques

## Avec Firefox

- Serif
- Sans-serif
- Monospace
- **Cursive**
- Fantasy

## Avec Microsoft Internet Explorer

- Serif
- Sans-serif
- Monospace
- Cursive
- *Fantasy*

## Avec Opera

- Serif
- Sans-serif
- Monospace
- **Cursive**
- Fantasy

## Avec SeaMonkey

- Serif
- Sans-serif
- Monospace
- **Cursive**
- **IFANTASY**

Rq. Se contenter des familles  
**Serif** et **Sans-serif**

Et puis une cinquième sur **Safari** (Mac) :

- Serif
- **Sans-serif**
- Monospace
- *Cursive*
- Fantasy



# Famille Serif

Les fontes de caractères de la famille **serif** assez courantes sont :

- Times new roman (PC)
- Times (Mac)
- Georgia (Mac/PC)
- Palatino Linotype (PC)
- Palatino (Mac)

Elles n'ont pas toutes le même aspect, les associations à respecter pour une homogénéité de rendu pourraient donc être :

- `body { font-family: times new roman, times, serif; }`
- `body { font-family: georgia, serif; }`



# Famille Sans-serif

- Verdana (Mac/PC)
- Arial (Mac/PC)
- Trebuchet (PC)
- Helvetica (Mac)
- Tahoma (PC)
- Geneva (Mac)
- Utiliser des associations: exemples:
  - `body { font-family:trebuchet, helvetica, sans-serif; }`
  - `body { font-family:tahoma, geneva, sans-serif; }`



# Formatage du texte : Le style

- **Le font-style** peut avoir une valeur parmi trois :
  - **Normal**: le texte sera normal (par défaut). Cela permet d'annuler une mise en italique.
  - **Italic** : le texte sera mis en italique
  - **Oblique** : le texte sera passé en oblique (les lettres sont penchées, le résultat est légèrement différent de l'italique)

```
font-style: normal;
```

```
font-style: italic;
```

```
font-style: oblique;
```



# Formatage du texte : Le poids

- **Mettre en gras** : La propriété CSS pour mettre en gras est **font-weight** et prend les valeurs suivantes :
  - bold : le texte sera en gras ;
  - normal : le texte sera écrit normalement (par défaut).

```
font-weight: normal;
```

```
font-weight: bold;
```



# Formatage du texte : Décoration

- **Soulignement et autres décorations** : La propriété **text-decoration** peut prendre une valeur parmi cette liste :
  - **underline** : souligné.
  - **line-through** : barré.
  - **overline** : ligne au-dessus.
  - **blink** : clignotant. Ne fonctionne pas sur tous les navigateurs (Internet Explorer et Google Chrome, notamment).
  - **none** : normal (par défaut).

La propriété CSS *text-decoration* permet de décorer un peu son texte :

en le soulignant...

~~en le barrant...~~

...ou encore en mettant une ligne au-dessus.



# Formatage du texte : Alignement

- On utilise la propriété **text-align** et on indique l'alignement désiré :
  - **left** : le texte sera aligné à gauche (c'est le réglage par défaut).
  - **center** : le texte sera centré.
  - **right** : le texte sera aligné à droite.
  - **justify** : le texte sera « justifié ».



# Formatage du texte : Les flottants

- Le CSS nous permet de faire flotter un élément autour du texte. On dit aussi qu'on fait un **habillage**.
- La propriété « float » peut prendre deux valeurs :
  - **left** : l'élément flottera à gauche.
  - **right** : l'élément flottera à droite

## Code HTML

```
<p>  
      
    Ceci est un texte  
</p>
```

## Code CSS

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

dipisicing elit. Aliquam  
corporis ipsam. Maiores  
que nulla quod quasi.  
one praesentium nisi  
ad molestiae natus quaerat quidem  
unt! Expedita vitae explicabo, quo

Image  
80x80

**NB.** Vous devez placer l'élément flottant en premier dans le code HTML.  
Si vous placez l'image après le paragraphe, l'effet ne fonctionnera pas.





# La couleur

- On change la couleur du texte avec la propriété **color**
- On change la couleur de l'arrière plan avec la propriété **background-color**
- On peut indiquer une couleur :
  - en écrivant son nom en anglais (black, par exemple),
  - sous forme hexadécimale (#FFC8D3)
  - ou en notation RGB (rgb(250,25,118)).



# Exercice

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <link rel="stylesheet" href="style2.css">
```

```
    <title>programmation web</title>
```

```
</head>
```

```
<body>
```

```
    <h1>programmation web</h1>
```

<p>La programmation web est la programmation informatique qui permet d'éditer des sites web. Elle permet la création d'applications, destinées à être déployées sur Internet ou en Intranet.

Ces applications web sont constituées de pages web pouvant prendre différentes formes,

```
    </p>
```

```
</body> </html>
```



# Exercice

- Apparence par défaut
- Apparence avec CSS





# Correction : CSS

```
h1 {  
  font-family: arial;  
  color:red;  
  text-align:center;  
}  
  
body {  
  font-family: arial;  
  color:blue;  
  text-align:justify;  
  background-color: SILVER;  
}
```



# Traitement des paragraphes



# Traitement des paragraphes

- Les dimensions d'un paragraphe
  - **width** : largeur du bloc (en pixels (px) ou en pourcentage (%)).
  - **height** : hauteur du bloc (en pixels (px), soit en pourcentage (%))



- la propriété **overflow** : Quand le texte dépasse les limites du paragraphe
  - **visible** (par défaut) : le texte reste visible et sort volontairement du bloc.
  - **hidden** : le texte sera tout simplement coupé. On ne pourra pas voir tout le texte.
  - **scroll** : le texte sera coupé mais le navigateur mettra en place des barres de défilement pour qu'on puisse lire l'ensemble du texte.
  - **auto** : c'est le navigateur qui décide de mettre ou non des barres de défilement



# Traitement des paragraphes

- Les dimensions d'un paragraphe
  - **word-wrap** : couper les textes trop larges

http://www.superlongueadresse.com/chemin/vers/une/page/web/fichier.html

```
p  
{  
    word-wrap: break-word;  
}
```

http://www.superlongueadresse.com/che  
min/vers/une/page/web/fichier.html



# Traitement des paragraphes

- On peut demander à ce qu'un bloc ait des dimensions minimales et maximales.
- Cela permet de définir des dimensions « limites » pour que notre site s'adapte aux différentes résolutions d'écran de nos visiteurs :
  - **min-width** : largeur minimale ;
  - **min-height** : hauteur minimale ;
  - **max-width** : largeur maximale ;
  - **max-height** : hauteur maximale.

```
p {  
  width: 50%;  
  min-width: 200px;  
}
```





# L'ombre

- On peut ajouter une ombre aux blocs de texte avec **box-shadow**.
- On doit indiquer le décalage vertical et horizontal de l'ombre, son niveau d'adoucissement et sa couleur.

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

test de paragraphe

- Le texte peut lui aussi avoir une ombre avec **text-shadow**.

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

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.



```
html, body {
  margin: 0;
  padding: 0;
}
body {
  background-color: white;
  font-family: Verdana, sans-serif;
  font-size: 100%;
}
h1 {
  font-size: 200%;
  color: navy;
  text-align: center;
}
h2 {
  font-size: 150%;
  color: red;
  padding-left: 15px;
}
p,ul,li,td {
  color: black;
}
a:link {
  color: green;
  text-decoration: underline;
}
a:visited {
  color: gray;
}
a:hover {
  color: red;
  text-decoration: none;
}
a:active, a:focus {
  color: red;
}
```

## Exercice

Cet exemple de feuille de style donnera une page :

- à fond blanc,
- avec une police de caractère par défaut Verdana noire,
- un titre de premier niveau bleu marine centré,
- un titre de deuxième niveau rouge décalé de 15 pixels,
- des liens passant du **vert** au **gris** avec un petit effet rouge non souligné au passage de la souris.



# Les bordures



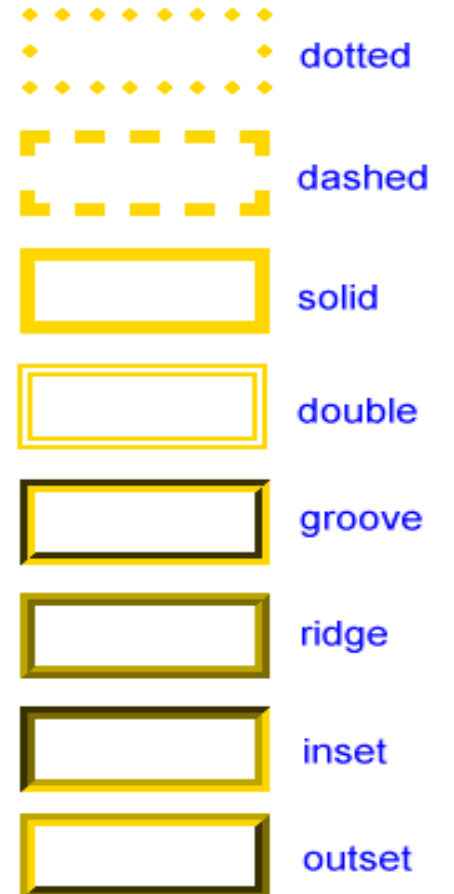
# Les bordures

- Il existe plusieurs types de bordures.

```
p {  
  border-style: solid;  
  border-color: gold; }
```

- On peut utiliser les valeurs "none" ou "hidden" si on ne veut aucune bordure.

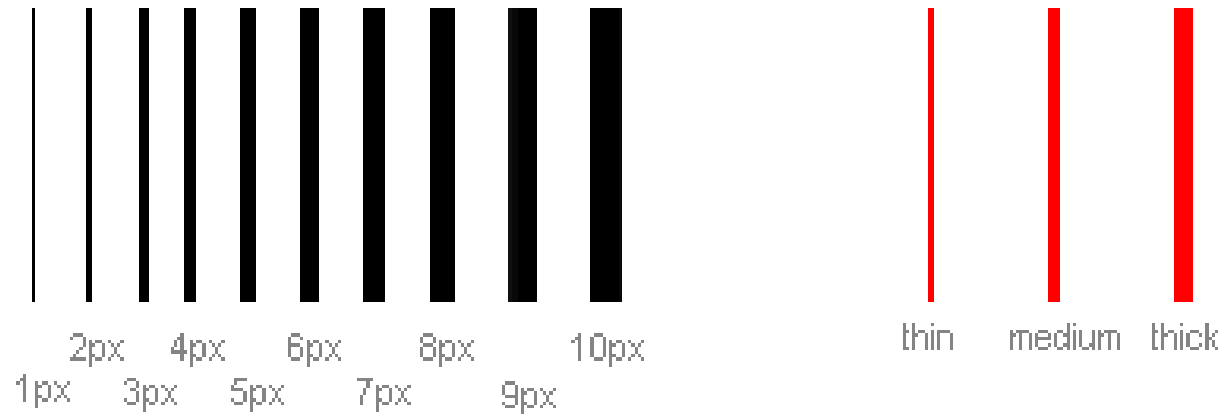
```
p {  
  border-style: hidden/none;  
}
```





# Les bordures

- L'épaisseur des bordures est définie par la propriété `border-width`, dont les valeurs peuvent être "**thin**", "**medium**" et "**thick**", ou une valeur numérique en pixels.



`p {`

`border-width: 1px;`

`border-style: solid;`

`border-color: blue; }`

`= p { border: 1px solid blue; }`



# Réaliser des coins arrondis

## Propriétés utilisées

- **border**
- **border-radius**
- On peut utiliser les préfixes propriétaires –moz (e.g., firefox) ou -webkit (e.g., chrome, Safari) pour avoir la bonne restitution de l'arrondissement des coins.
- La propriété border-radius peut accepter 4 valeurs pour l'arrondissement de chaque coins. La 1ère valeur correspond au coin haut gauche, puis on tourne dans le sens des aiguilles d'une montre.
- On peut n'en indiquer que 2, qui correspondront aux coins opposés (voir l'exemple ci-dessous), ou une seule pour un même arrondis sur les 4 coins.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vulputate laoreet urna. Integer magna. Donec facilisis lectus sed quam. Curabitur sit amet lacus id lacus facilisis venenatis.



# Réaliser des coins arrondis

## Code HTML

```
<div class="coin">  
    <p> Lorem ipsum ... </p>  
</div>
```

## Code CSS

```
.coin {  
background-color: #E4EFFF;  
border: 1px solid #9FC6FF;  
padding: 5px;  
/*arrondir les coins en haut à gauche et en bas à droite*/  
-moz-border-radius: 10px 0;  
-webkit-border-radius: 10px 0;  
border-radius: 10px 0;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vulputate laoreet urna. Integer magna. Donec facilisis lectus sed quam. Curabitur sit amet lacus id lacus facilisis venenatis.



# Arrondir les coins d'une image

- Pour arrondir les coins d'une image, même principe :

## Code HTML

```
<img src= "images/IIT.jpg" width="300" height="400" class="arrondie" />
```

## Codes CSS

**.arrondie** {

-moz-border-radius: 50px 0px;

-webkit-border-radius: 50px 0px;

border-radius: 50px; 0px}







# Réaliser un fond en dégradé **sans image**

On utilise la propriété « background-image »

Ses valeurs sont :

- linear-gradient()
- radial-gradient()

Le principe de base est très simple

1. On définit le type de dégradé voulu :
  - linéaire (linear-gradient)
  - radial (radial-gradient)
2. On indique les couleurs de départ et d'arrivée



# Exemples

```
body {  
  background-image:  
    radial-gradient(white, red);  
}
```



```
body {  
  background-image:  
    linear-gradient(to right, white, black);  
}
```





# Image de fond en css

- La propriété **background-image** permet de définir une **image** comme arrière-plan pour un élément.
- Exemple

```
body {  
  
    background-image: url(images/logoIT.jpg);  
    background-repeat:no-repeat;  
    background-position:right top;  
  
}
```

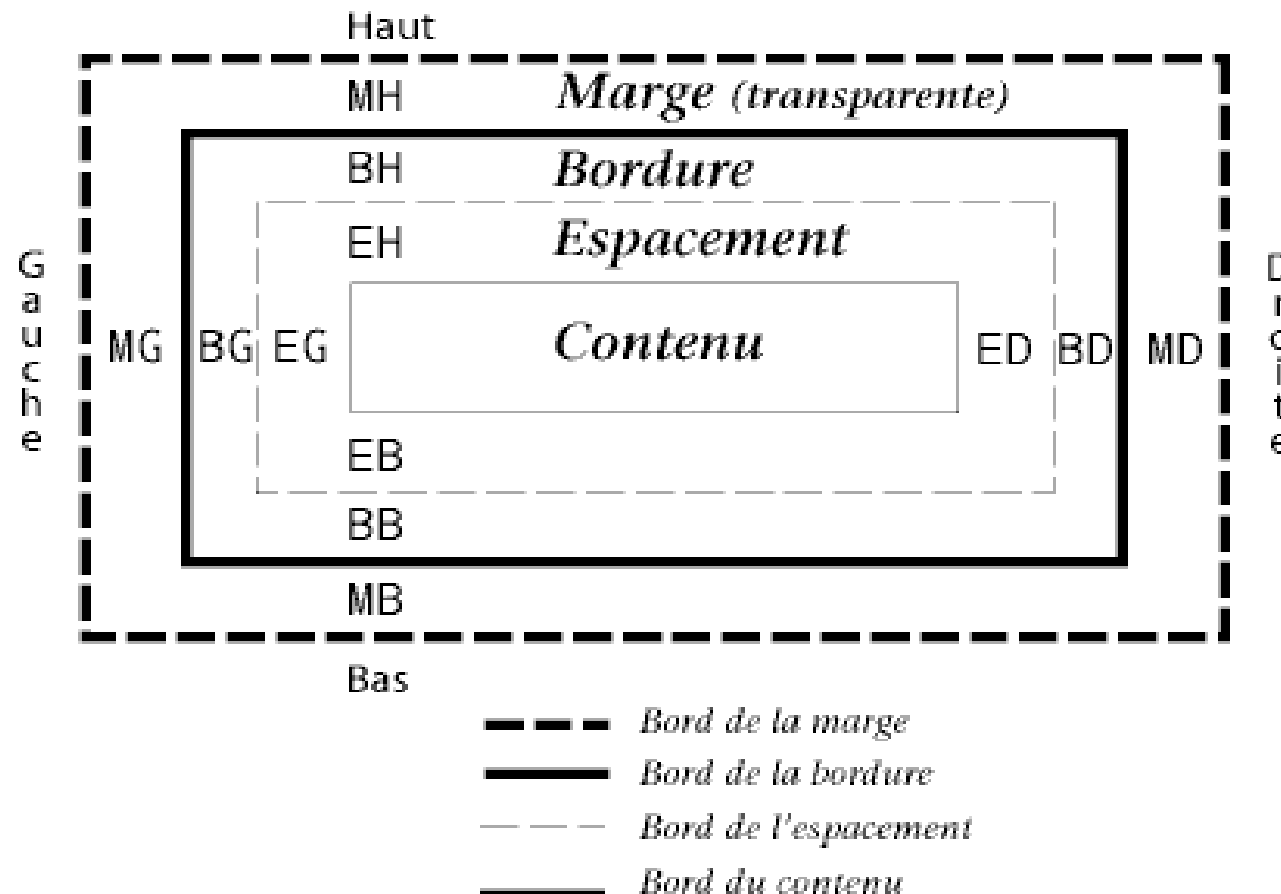


# Le modèle des boites



# Le modèle des boîtes

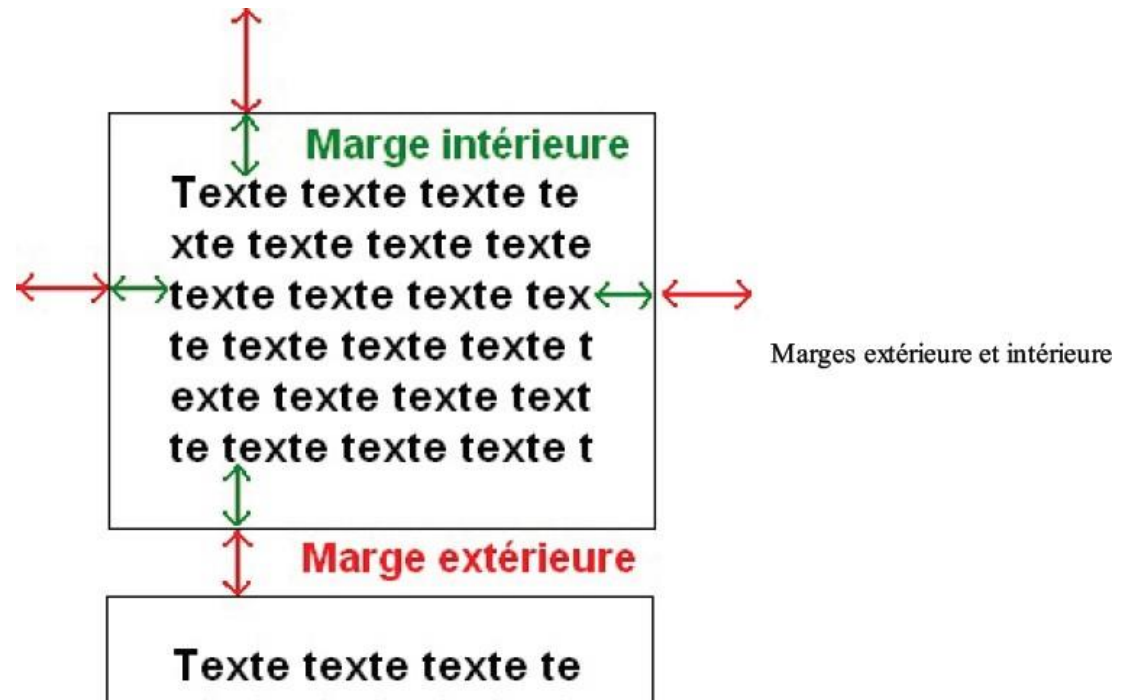
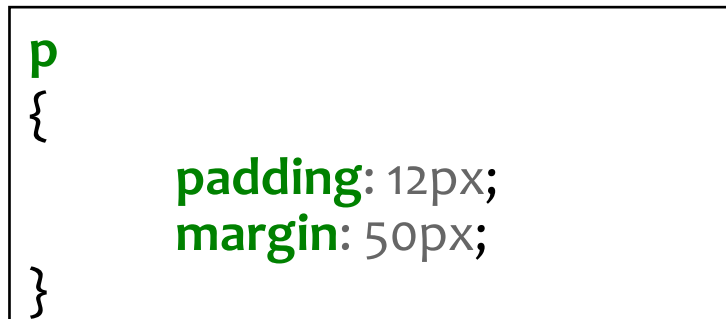
- Le modèle des boîtes de CSS décrit les boîtes générées pour les éléments HTML.
- Il contient également des options détaillées pour l'**ajustement des marges**, des **bordures**, des **espacements** et du **contenu** de chaque élément.





# Les marges et les espacements (Margin et Padding)

- Les blocs possèdent des marges.
- Il existe deux types de marges :
  - les marges **intérieures** (**Padding**)
  - les marges **extérieures**/ les espacements (**Margin**)



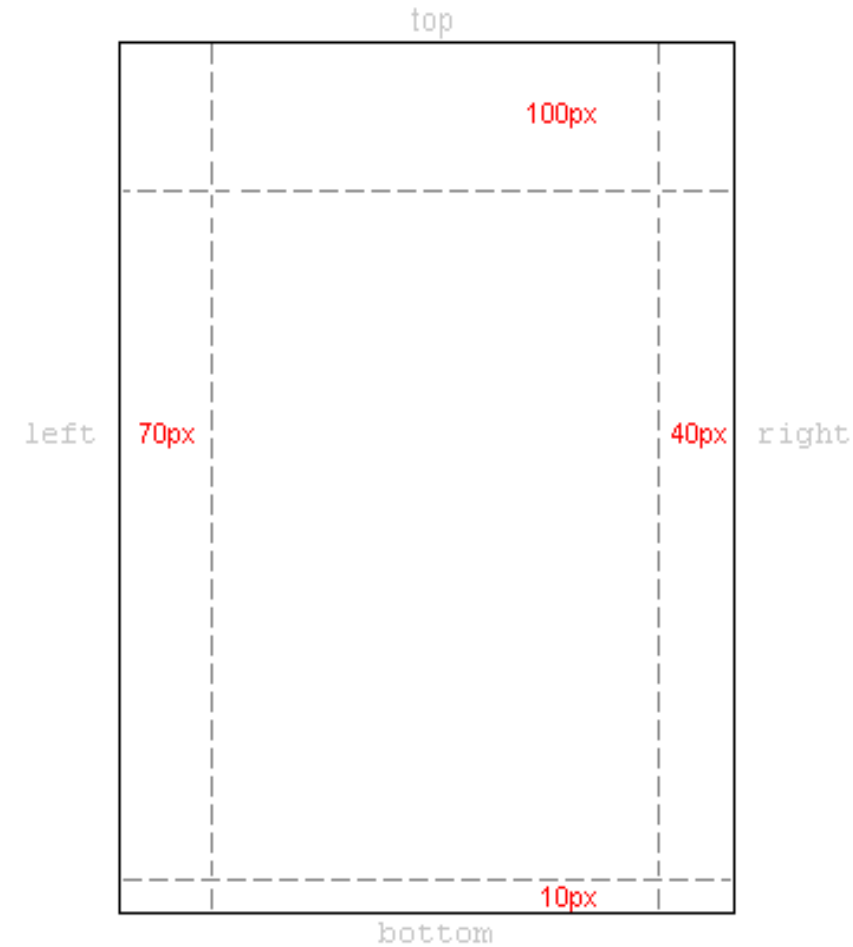


# Les Marges (Margin)

- Un élément a quatre côtés : droit ("right"), gauche ("left"), supérieur ("top") et inférieur ("bottom").
- La marge est la distance entre chaque côté et l'élément avoisinant (document).
- Pour les marges du document :

```
body { margin-top: 100px;  
        margin-right: 40px;  
        margin-bottom: 10px;  
        margin-left: 70px; }
```

Ou `body { margin: 100px 40px 10px 70px; }`





# Exemple

```
p {  
  width:200px;  
  padding:40px;  
  margin:auto;  
}
```



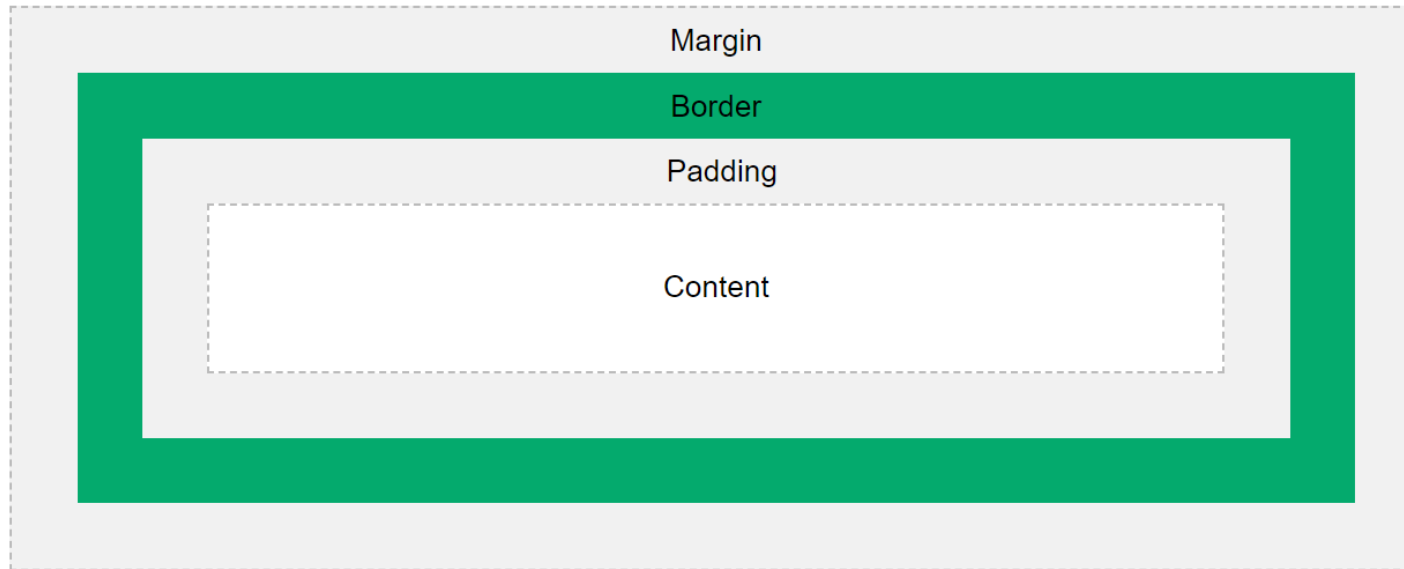
Le paragraphe aura une largeur totale de  $200 + 2 \times 40 = 280$  pixels et sera centré dans son élément parent.





# L'espacement (Padding)

- L'espacement peut aussi être assimilé à un « remplissage ».
- L'espacement définit la distance interne entre la bordure et le contenu de l'élément.



- Exemple : la définition de l'espacement des titres change le remplissage (couleur) autour du texte dans chaque titre.

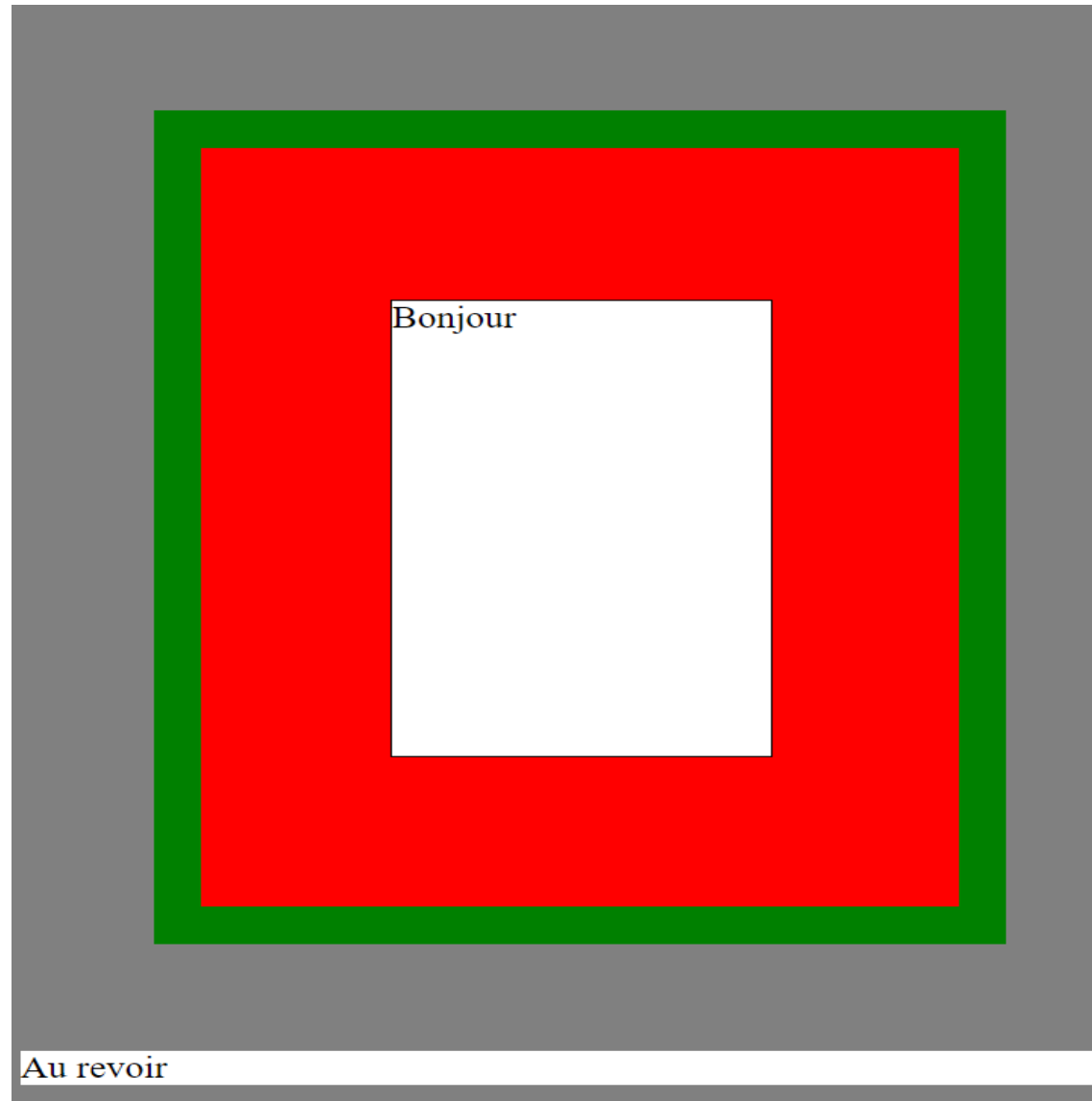
```
h1 { background-color: yellow; padding: 20px 20px 20px 80px; }
```

```
h2 { background-color: orange; padding-left: 120px; }
```



# Exercice

- Réaliser le modèle en boîte suivant.





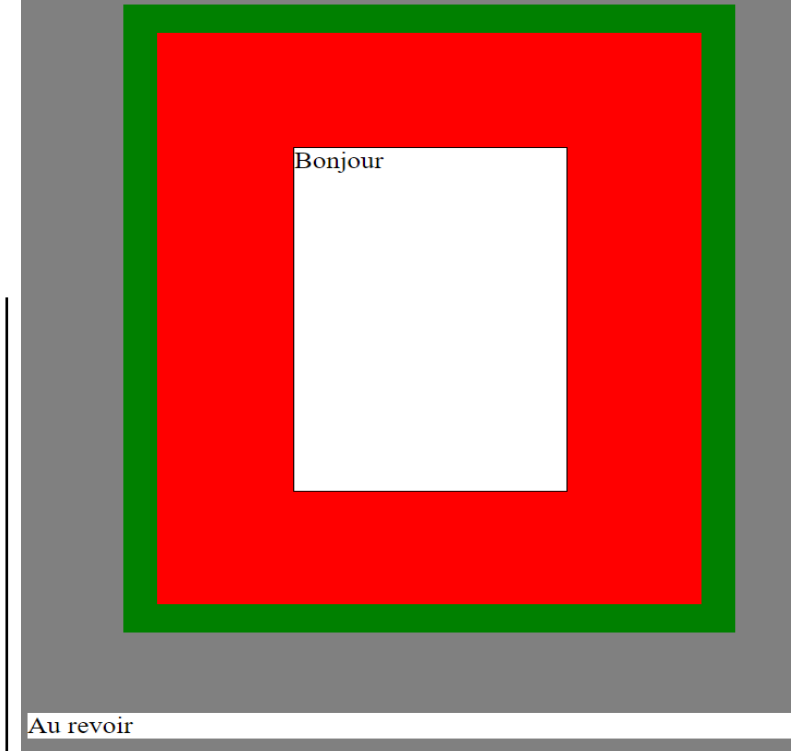
# Correction

```
<html>

<head>
  <style>
    body {
      background-color: grey;
    }

    #a {
      background-color: red;
      border: 25px solid green;
      width: 200px;
      height: 300px;
      padding: 100px;
      margin: 70px;
    }
  </style>
</head>
```

```
    #a div {
      width: 100%;
      height: 100%;
      border: 1px solid black;
      background-color: white;
      font-size: 20px;
    }
    #b {
      background-color: white;
      font-size: 20px;
    }
  </style>
</head>
```



```
<body>
  <div id="a">
    <div>Bonjour</div>
  </div>
  <div id="b">Au revoir</div>
</body>

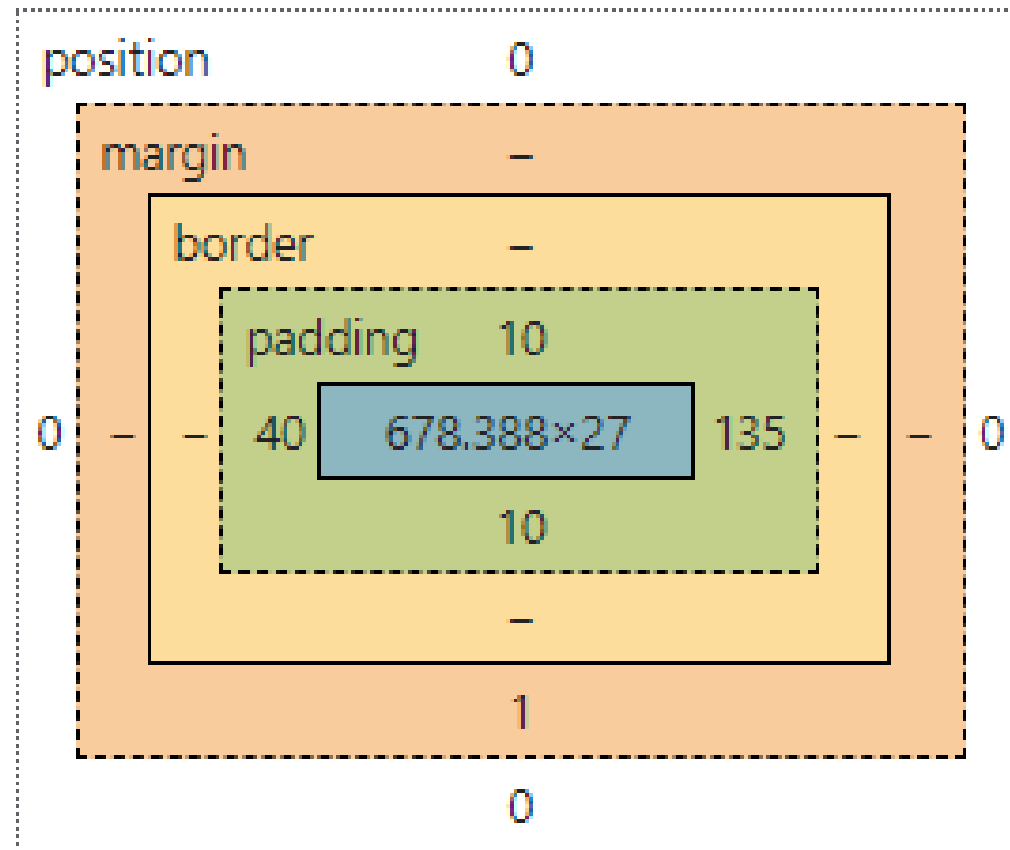
</html>
```



# Le positionnement en CSS



# Le positionnement en CSS





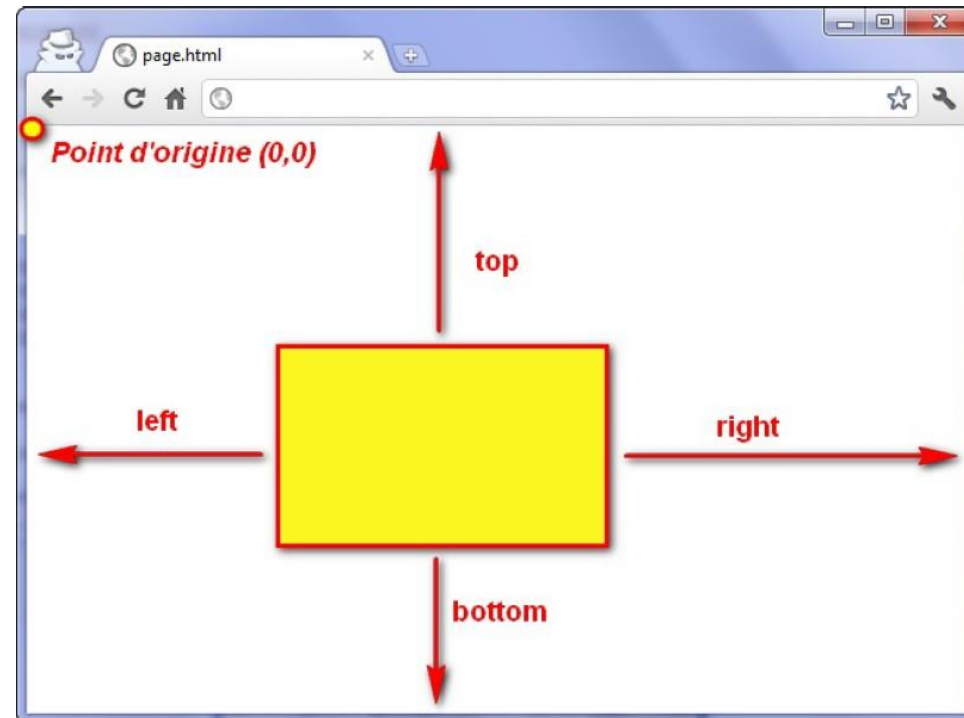
# Le positionnement en CSS

- Il existe d'autres techniques un peu particulières permettant de positionner avec précision des éléments sur la page :
  - **Le positionnement absolu** : (*absolute*) il permet de placer un élément n'importe où sur la page (en haut à gauche, en bas à droite, tout au centre, etc.).
  - **Le positionnement fixe** : (*fixed*) identique au positionnement absolu. Mais, cette fois, l'élément **reste toujours visible**, même si on descend plus bas dans la page.
  - **Le positionnement relatif** : (*relative*) permet de décaler l'élément par rapport à sa position normale.

# Le positionnement en CSS

- **Le positionnement absolu :**
  - Il faut donc utiliser la propriété **position** et au moins une des quatre propriétés (**top**, **left**, **right** ou **bottom**).

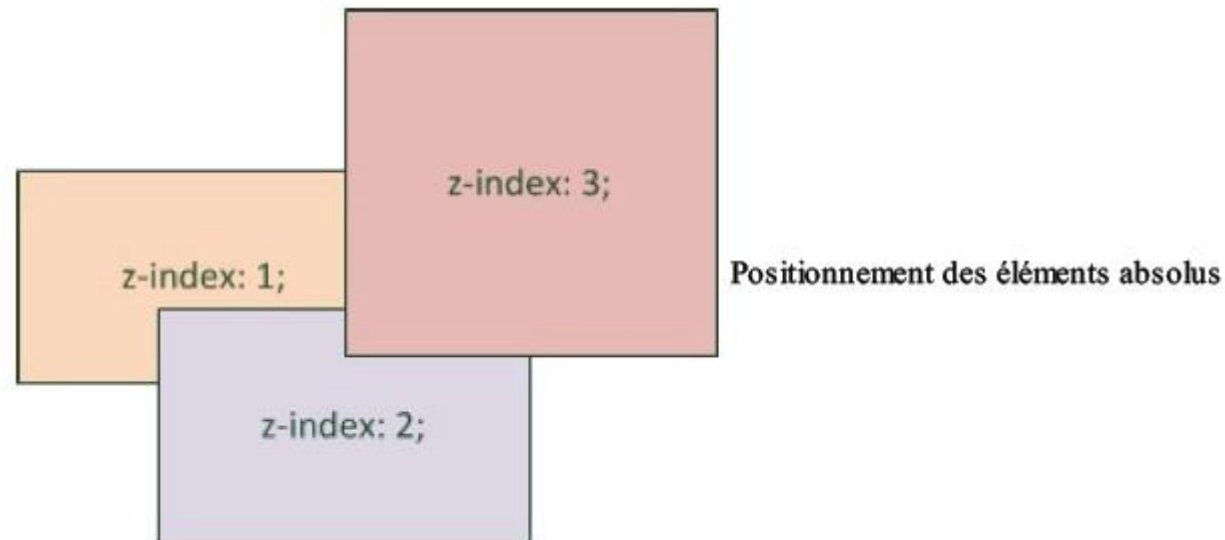
```
div
{
  position: absolute;
  right: 0px;
  bottom: 0px;
}
```





# Le positionnement en CSS

- **Le positionnement absolu :**
  - Deux éléments en absolu placé vers le même endroit risquent de se chevaucher.
  - Dans ce cas, utilisez la propriété **z-index** pour indiquer quel élément doit apparaître au-dessus des autres.
  - L'élément ayant la valeur de **z-index** la plus élevée sera placé par dessus les autres.







# Le positionnement en CSS

- **Le positionnement absolu :**
  - Il ne se fait pas forcément toujours par rapport au coin en haut à gauche de la fenêtre
  - En fait, si vous positionnez **en absolu un bloc A** qui se trouve **dans un autre bloc B**, lui-même positionné en absolu (ou fixe ou relatif),
  - Alors **le bloc A se positionnera par rapport au coin supérieur gauche du bloc B.**
- **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 on descend plus bas dans la page.



# Le positionnement en CSS

- **Le positionnement relatif**

- Ce positionnement permet d'effectuer des « ajustements » : l'élément est décalé par rapport à sa position initiale.
- **Par exemple :**

**strong**

{

**background-color:** red;

**color:** yellow; **position:** relative; **left:** 55px;

**top:** 10px;

}

Pas de doute,  si on veut comprendre

point d'origine (0, 0)

Pas de doute,  si on veut com Le texte

55px

10px



# Media Queries de CSS3



# Cibler la largeur de la fenêtre avec CSS3

- On peut cibler la largeur de la fenêtre avec les Media Queries de CSS3

## Exemples

- On peut changer la couleur de fond initialement blanche de la page lorsque la taille de la fenêtre du navigateur passe au dessous des 1000 pixels de large.
- un smartphone aura un fond noir en orientation portrait et blanc en orientation paysage.

@media only screen and (max-width: 1000px) {

body{

background-color: black;

color: white;

}

}



# Cibler la largeur de la fenêtre avec CSS3

## Petits écrans (téléphones)

```
@media only screen and (max-width: 600px) {  
  /* Styles pour les petits écrans (téléphones) */  
}
```

## Écrans moyens (tablettes)

```
@media only screen and (min-width: 601px) and (max-width: 1024px) {  
  /* Styles pour les écrans moyens (tablettes) */  
}
```

## Grands écrans (ordinateurs de bureau) :

```
@media only screen and (min-width: 1025px) {  
  /* Styles pour les grands écrans (ordinateurs de bureau) */  
}
```



# Récap : les propriétés CSS

- Les styles de police
- Les styles de texte
- Les styles des arrières plans
- Les styles des boîtes
- Les styles des listes



# Les styles de police

Propriété	Valeurs (exemples)	Description
font-family	serif, sans-serif, arial	Nom de police
font-size	1.3em, 16px, 120%, small, large..	Taille du texte
font-weight	bold, normal	Epaisseur de la police
font-style	italic, oblique, normal	Style de l'écriture
font-variant	small-caps, normal	définit une variante par rapport à la normale
Font	P {font: bold italic}	Super propriété de police. Combine: font-weight, font-style, font-size, font-variant, font-family.



# Les styles de texte

Propriété	Valeurs (exemples)	Description
text-align	left, center, right, justify	Alignement horizontal
line-height	18px, 120%, normal...	Espace entre les lignes du texte
text-indent	25px	Retrait dans la première ligne d' un bloc de texte <p>
white-space	pre, nowrap, normal	Espace ou blanc
text-decoration	Underline, line-through, overline, none	Souligné, barré, ligne au dessus,
color	#000080, red,rgb(250,25,118)	Couleur du texte





# Les styles des arrières plans

Propriété	Valeurs (exemples)	Description
background-color	Identique à color	Couleur de fond
background-image	url('image.png')	Image de fond
background-attachment	fixed, scroll	si l'image d'arrière-plan reste fixe avec les déplacements de l'écran
background-repeat	repeat-x, repeat-y, no-repeat, repeat	la façon de répéter l'image d'arrière-plan
background-position	(x y), top, center, bottom, left, right	Position du fond
background	P {background: image.gif fixed repeat}	Super propriété du fond. Combine : background-image, background-repeat, background-attachment, background-position



# Les styles des boites

Propriété	Valeurs (exemples)	Description
width	150px, 80%...	Largeur
height	150px, 80%...	Hauteur
margin-top	23px	Marge en haut
margin-left	23px	Marge à gauche
margin-right	23px	Marge à droite
margin-bottom	23px	Marge en bas



# Les styles des boites

Propriété	Valeurs (exemples)	Description
padding-left	23px	Marge intérieure à gauche
padding-right	23px	Marge intérieure à droite
padding-bottom	23px	Marge intérieure en bas
padding-top	23px	Marge intérieure en haut
padding	23px 5px 23px 5px ( <i>haut, droite, bas, gauche</i> )	Super-propriété de marge intérieure. Combine : padding-top, padding-right, padding-bottom, padding-left.

# Les styles des boites

Propriété	Valeurs (exemples)	Description
border-width	3px	Épaisseur de bordure
border-color	<i>nom</i> , rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20...	Couleur de bordure
border-style	solid, dotted, dashed, double, groove, ridge, inset, outset	Type de bordure
border	3px solid black	Super-propriété de bordure. Combine border-width, border-color, border- style. Existe aussi en version border- top, border-right, border- bottom, border-left.



# Les styles des listes

Propriété	Valeurs (exemples)	Description
list-style-type	disc ou circle ou square decimal ou lower-roman ou upper-roman ou lower-alpha ou upper-alpha UL {list-style-type: square}	détermine le type de puces ou de numérotation
list-style-image	url ou none UL {list-style-image: image.gif}	permet de remplacer les puces par une image
list-style-position	inside ou outside UL {list-style-position: inside}	spécifie si les puces sont à l'intérieur ou à l'extérieur du texte
list-style	UL { list-style: disc url(puce.gif) inside; }	regroupe toutes les propriétés de liste