

# CSS



```
body {  
  font: x-small;  
  background: #  
  color: black;  
  margin: 0;  
  padding: 0;
```

# CSS3 – indispensable avec HTML 5

---

Une nouvelle page de l'histoire du web est en train de se tourner. CSS3 est désormais plus prisé que jamais car c'est le compagnon indispensable à la superstar HTML5.

C'est CSS3 qui fera que votre site ou votre application web sera attrayante ou non.



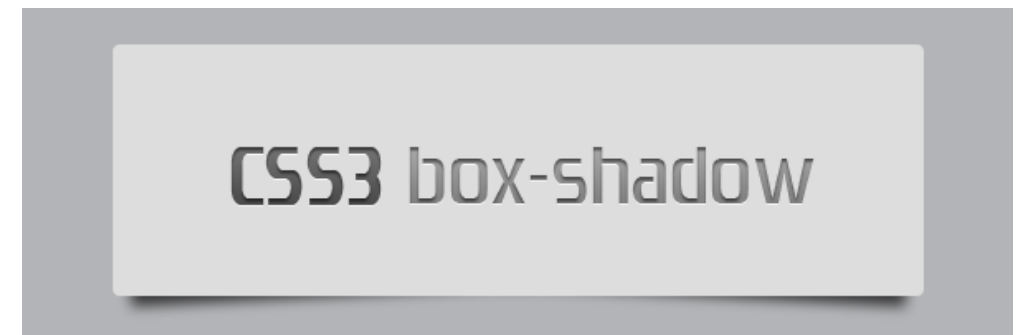
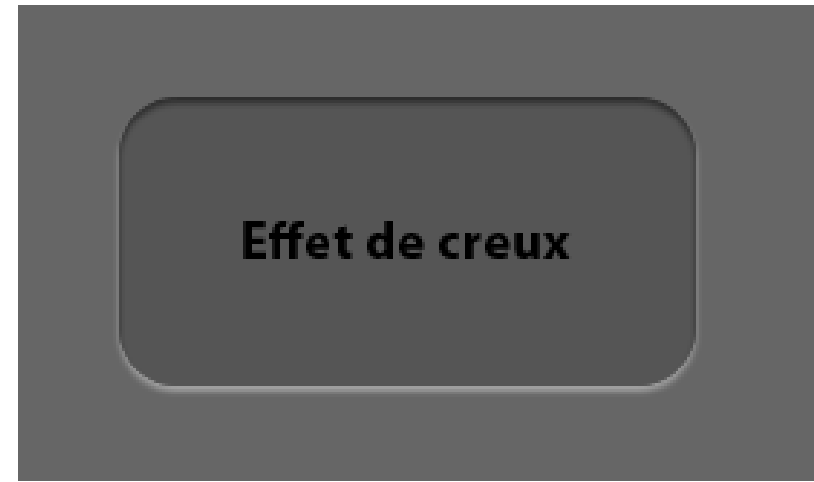
# Attente des utilisateurs

---

Les attentes des utilisateurs vis-à-vis des interfaces ont nettement évoluées suite aux succès des smartphones et tablettes.

Ils veulent des interfaces novatrices, attirantes et efficaces.

CSS3 propose pour cela toute une palette d'effets graphiques à base d'ombre, de dégradés, de bords non rectangulaires, de transformations 2D voire même en 3D !



# Où écrire le CSS ?

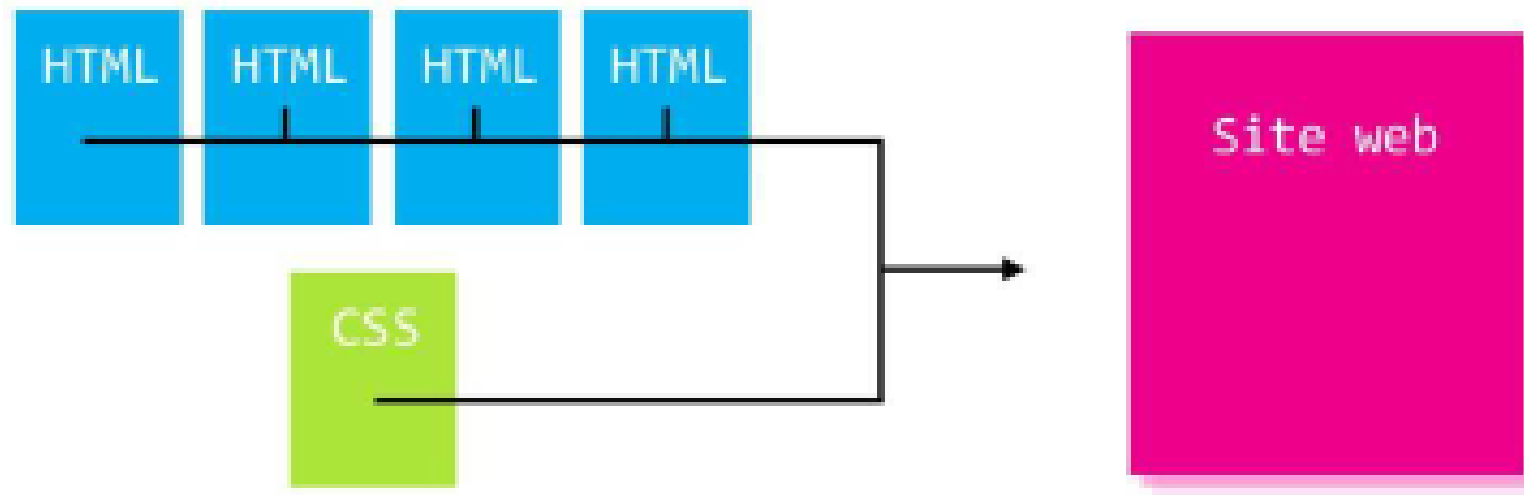
---

Il est recommandé d'écrire le css dans un fichier à part avec pour extensions .css et de l'inclure dans le <head> de votre fichier html .

Il est possible mais fortement déconseillé d'écrire du style directement dans les balises html avec l'attribut « style » .

# Où écrire le CSS ?

---



**Figure 1.2** — Le principe de la séparation fond/forme

# Intérêts de la séparation multiple

---

- Pouvoir disposer d'une feuille de styles unique, applicable à des documents multiples ;
- Pouvoir disposer de feuilles de styles alternatives, en fonction du choix des concepteurs ou des utilisateurs ou en fonction des médias ;
- Avoir une meilleure maintenabilité d'un site web (les mises à jour sont plus simples et plus rapides) ;
- Avoir une plus grande flexibilité (une seule modification du CSS modifie l'ensemble des pages d'un site web).

# Appeler un fichier CSS

---

Pour commencer, il va vous falloir appeler votre feuille de style :

- Placez-vous sur votre page html, dans la partie <head>
- Et notez : `<link rel=« stylesheet » href=« adresse de votre .css » />`

# Sélecteurs CSS

Les attributs « class » et « id » se ressemblent beaucoup

**Ils servent à identifier vos balises, la class peut se répéter alors que l'ID doit lui être unique dans une page.**

Vous pouvez les nommer comme vous voulez du moment que le nom commence par une lettre.

Fichier .html :

```
<h1>Bailly Benjamin</h1>
<p id="firstParagraph">
  Je m'appelle <strong>Bailly Benjamin</strong>. Je suis développeur Web et référent de votre groupe.</br>
  J'ai été coordinateur et directeur dans la formation et l'animation avant d'être développeur.
</p>
<p>
  <a class="contactLink" href="mailto:benjaminbailly@adrar-formation.com">Envoyez un e-mail à Benjamin</a>
</p>
<a href="images/borat.jpg" target="_blank"></a>
```

Fichier .css :

```
/* selectionne tous les h1 */
h1{
  font-size: large;
  color: red;
}

/* Selectionne l'id firstParagraph */
#firstParagraph {
  text-decoration:underline;
}

/* Selectionne toutes les class contactLink */
.contactLink{
  text-decoration: none;
  color: red;
}
```



# Sélecteurs CSS

---

Je vous invite à vous rendre à l'adresse suivante : <https://www.w3schools.com/cssref/trysel.asp>

Nous allons en apprendre plus sur les sélecteur et plus particulièrement nth-child()

# Sélecteurs CSS universel

---

Il sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel :

```
*  
{  
  
}
```

# Sélecteurs CSS une balise dans une autre

---

Sélectionne toutes les balises `<strong>` à l'intérieur des balises `<p>` :

```
p strong  
{  
  
}
```

```
<p>Bonjour <strong>Toto</strong></p>
```

# Sélecteurs CSS une balise avec un attribut

---

Sélectionne tous les liens <a> ayant un attribut « title » :

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

```
<a href=« www.monsite.fr » title=« titi »>Toto</a>
```

# Balises universelle

---

Ces balises n'ont aucune signification particulière et peuvent être utilisées pour ajouter une class ou un id à un objet sans balise :

- `<span></span>` : type inline
- `<div></div>` : type block

# Balises inline

---

Les éléments de rendu **inline** se placent toujours l'un à côté de l'autre afin de rester dans le texte

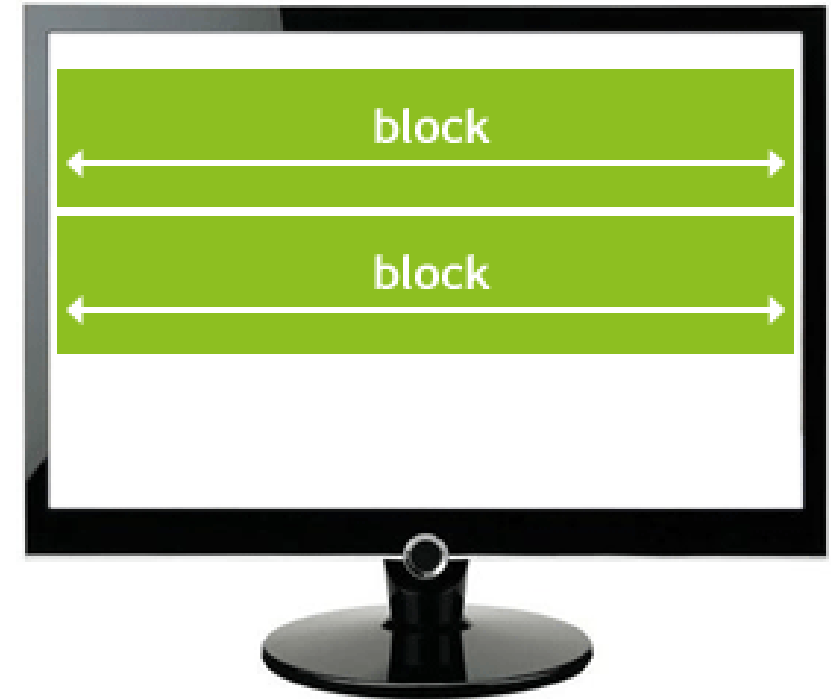
Par défaut, il n'est pas prévu qu'ils puissent se positionner sur la page (même si cela est possible), ni de leur donner des dimensions (hauteur, largeur, profondeur) : leur taille va être déterminée par le texte ou l'élément qu'ils contiennent. Certaines propriétés de marges peuvent être appliquées aux éléments, comme les marges latérales.



# Balises block

Les éléments de rendu CSS **block** se placent toujours l'un en dessous de l'autre par défaut (comme un retour chariot). Par exemple: une suite de paragraphes `<p>`.

Par ailleurs, un élément "bloc" occupe automatiquement, par défaut, toute la largeur disponible dans son conteneur et peut être dimensionné à l'aide des propriétés telles que « width », « height », « min-width », etc...



# Les couleurs

Indiquer le nom  
de la couleur

Aqua #00FFFF	Black #000000	Blue #0000FF	Fuschia #FF00FF
Gray #808080	Green #008800	Lime #00FF00	Maroon #800000
Navy #000080	Olive #808000	Purple #800080	Red #FF0000
Silver #C0C0C0	Teal #008080	White #FFFFFF	Yellow #FFFF00



# Les couleurs

---

Vous pouvez aussi trouver des extensions de navigateur qui permettent de "récupérer" n'importe quelle couleur qui vous plaît sur un site web :

[ColorPicker](#) pour Firefox

[ColorZilla](#) pour Chrome

# Les couleurs de fond et de texte

---

Il est possible de rentrer la version hexadécimal, RGB et « texte » d'une couleur.

```
body
{
    background-color: black;
    color: white;
}
```

# La transparence

---

Le CSS nous permet de jouer très facilement avec les niveaux de transparence des éléments ! Pour cela, nous allons utiliser des fonctionnalités de CSS3 : la propriété « opacity » et la notation RGBa.

Avec une valeur de 1, l'élément sera totalement opaque : c'est le comportement par défaut.

Avec une valeur de 0, l'élément sera totalement transparent.

```
p
{
    opacity: 0.3;
}
```

**ATTENTION: la propriété opacity s'hérite aux enfants de tous les éléments contenus**  
**A utiliser avec précaution.**

# La notation RGBa

---

CSS3 nous propose une autre façon de jouer avec la transparence : la notation RGBa. Il s'agit en fait de la notation RGB que nous avons vue précédemment, mais avec un quatrième paramètre : le niveau de transparence (appelé « canal alpha »). De la même façon que précédemment, avec une valeur de 1, le fond est complètement opaque. Avec une valeur inférieure à 1, il est transparent.

```
p
{
    background-color: rgba(255, 0, 0, 0.5);
}
```

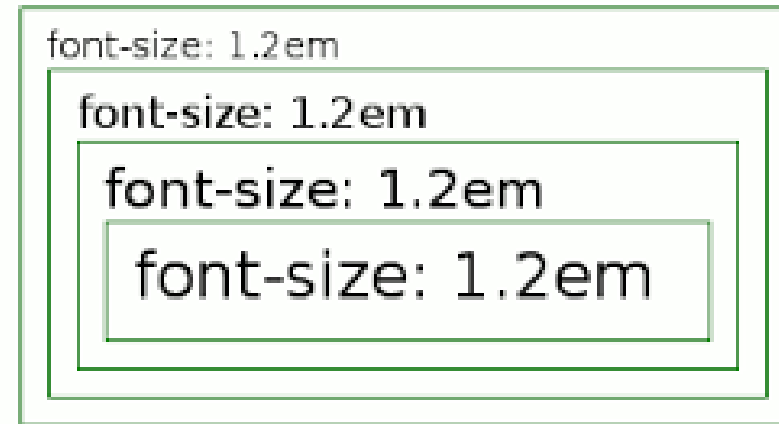
# Modifier la taille d'un texte

On utilise la propriété « font-size »

On peut indiquer la taille en :

- Pixels, centimètres ou millimètres (taille absolue)
- En pourcentage : em ou ex (taille relative)
- Utiliser les mots anglais spécifiques

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



# Modifier la police

---

La propriété CSS qui permet d'indiquer la police à utiliser est « font-family »

Il vous faudra préciser plusieurs noms de police, séparés par des virgules, si une police n'est pas disponible, la suivante sera sélectionnée.

```
balise
{
    font-family : police1, police2, police3, etc...;
}
```

Times New Roman  
Myriad Pro  
**Impact** Arial Tahoma  
Georgia Trebuchet  
Lucida  
Comic Sans

# Google Font

---

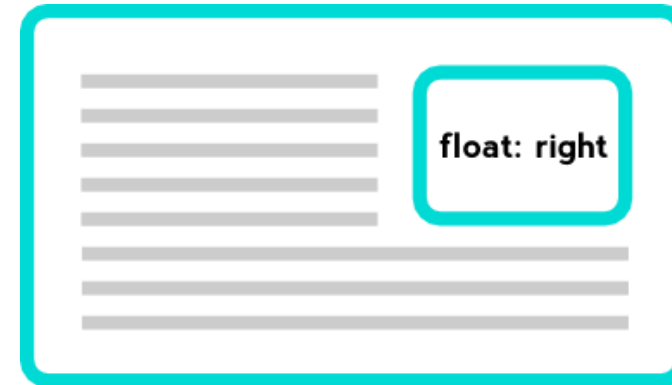
Vous trouvez le choix de polices limités ?

Allez faire un tour sur Google Font et suivez le tutorial pour installer la police de votre choix;-)

# Float

Nous pouvons faire « flotter » un élément autour du texte en utilisant la propriété « float », qui pourra prendre deux valeurs :

- Left : gauche
- Right : droite
- *Clear: both -> permet de stopper le flottement (à écrire dans le css de l'élément qui suit l'entourage du flottant)*





# Image de fond - options

---

Background-attachment : permet de fixer le fond. Le texte semble alors glisser par-dessus.

Deux valeurs sont disponibles :

- **fixed** -> l'image de fond reste fixe
- **scroll** -> l'image de fond défile avec le texte (par défaut)

```
body
{
    background-image: url(«plage.jpg»);
    background-attachment: fixed;
}
```

# Répétition du fond

---

- no-repeat : le fond ne sera pas répété
- repeat-x : le fond est répété horizontalement
- repeat-y : le fond est répété verticalement
- repeat : le fond est répété en mosaïque (par défaut)

```
body
{
    background-image: url(«plage.jpg»);
    background-attachment: no-repeat;
}
```

# Background-position : position du fond

---

On peut indiquer où doit se trouver l'image de fond avec background-position. Cette propriété n'est intéressante que si elle est combinée avec background-repeat: no-repeat;

Vous devez donner deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe, si vous appliquez le fond à un paragraphe). Ainsi, si vous tapez :

```
background-position : 50px 150px;
```

Votre fond sera placé à 50 pixels de la gauche et à 150 pixels du haut.

# Background-position : position du fond

---

Il est aussi possible d'utiliser ces valeurs en anglais :

Il est possible de combiner ces mots. Par exemple, pour aligner une image en haut à droite, vous taperez :

- `top` : en haut ;
- `bottom` : en bas ;
- `left` : à gauche ;
- `center` : centré ;
- `right` : à droite.

```
background-position : top right;
```

# Background-position : position du fond

---

Si vous utilisez beaucoup de propriétés en rapport avec le fond (comme c'est le cas sur ce dernier exemple), vous pouvez utiliser une sorte de « super-propriété » appelée « background » dont la valeur peut combiner plusieurs des propriétés vues précédemment :

- background-image
- background-repeat
- background-attachment
- background-position

On peut donc tout simplement écrire :

```
background: url(«plage.jpg») fixed no-repeat bottom right;
```

# Plusieurs images de fond

---

Depuis CSS3, il est possible de donner plusieurs images de fond à un élément. Pour cela, il suffit de séparer les déclarations par une virgule, comme ceci :

```
body
{
    background: url(«plage.jpg») fixed no-repeat bottom right, url(«palmier.jpg») fixed;
}
```

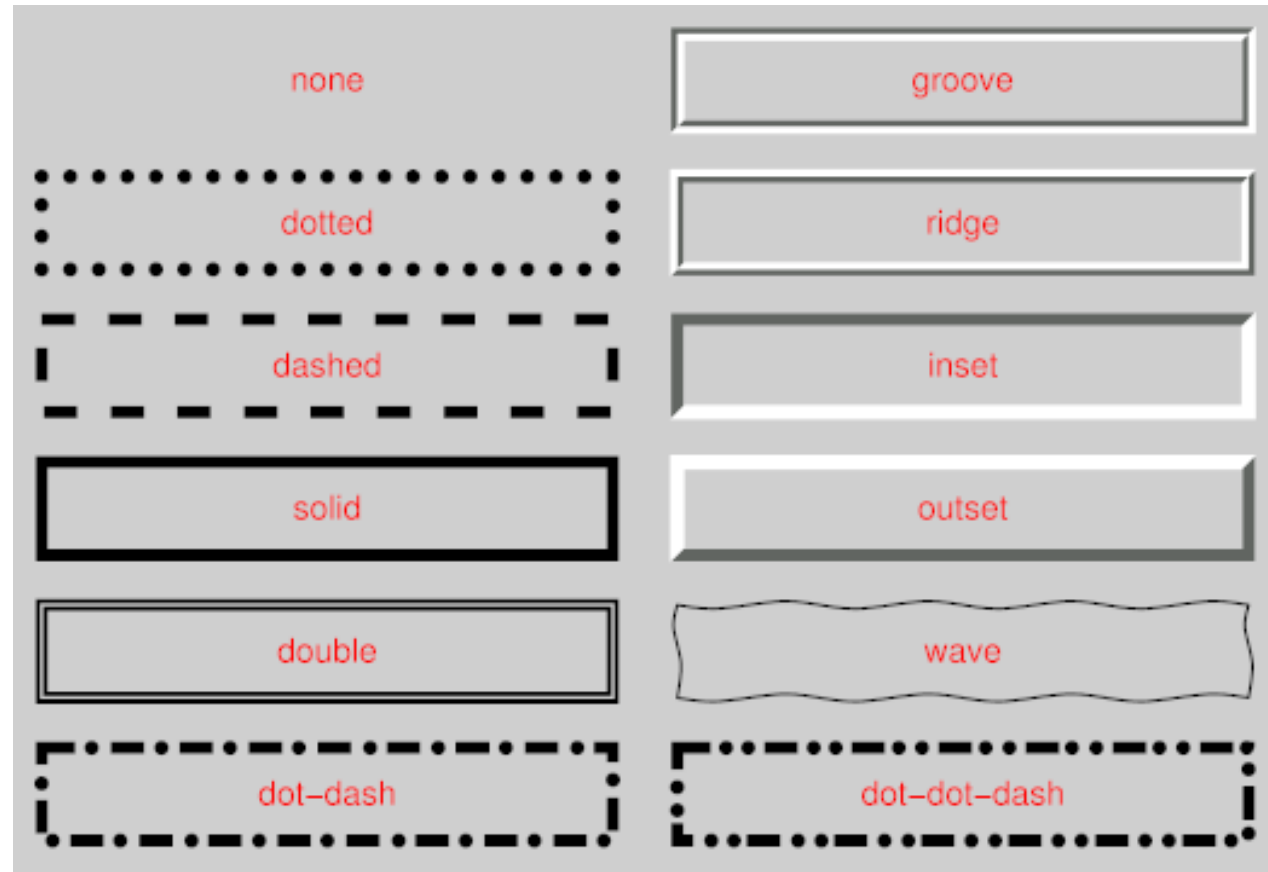
La première image de cette liste sera placée par-dessus les autres (figure suivante). Attention donc, l'ordre de déclaration des images a son importance!

# Bordures standard

La super-propriété border va vous permettre de gérer :

- La largeur : valeur en px
- La couleur : nom de la couleur, valeur hexadécimale ou valeur RGB
- Le type de bordure : type selon les valeurs proposées

```
border : 5px red ridge;
```



# Bordures standard

---

Si vous voulez mettre des bordures différentes en fonction du côté (haut, bas, gauche ou droite), vous pouvez le faire sans problème. Dans ce cas, vous devrez utiliser ces quatre propriétés :

- `border-top` : bordure du haut ;
- `border-bottom` : bordure du bas ;
- `border-left` : bordure de gauche ;
- `border-right` : bordure de droite.

`border-top : 3px blue ridge;`

Vous pouvez modifier les bordure de n'importe quel type d'élément. Essayez de le faire sur des paragraphes, des images, des titres, des caractères, etc...



# Bordures arrondies

---

La propriété « border-radius » va nous permettre d'arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer l'arrondi en pixels :

```
border-radius : 5px;
```

Pour préciser la forme de l'arrondi pour chaque coin, utilisez :

```
border-radius : 5px 3px 5px 3px;
```

Les valeurs correspondent, dans cet ordre :

- En haut à gauche
- En haut à droite
- En bas à droite
- En bas à gauche

# Bordures elliptiques

---

Il est possible d'affiner l'arrondi de nos angles en créant des courbes elliptiques.

Il faut indiquer deux valeurs séparées par *slash* :

```
border-radius : 5px / 10px;
```

# Les ombres

---

Box-shadow : les ombres des boites

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.

Par exemple, pour une ombre noire de 10 pixels, sans adoucissement, on écrira :

```
box-shadow: 10px 10px 0px black;
```

**N'hésitez pas à faire des tests ! 😊**

# Attribut : inset

---

On peut aussi rajouter une cinquième valeur facultative : « inset »

Dans ce cas, l'ombre sera placée à l'intérieur du bloc, pour donner un effet enfoncé :

```
box-shadow: 10px 10px 0px black inset;
```

# Text-shadow : l'ombre du texte

---

Ajoutez une ombre à vos textes avec text-shadow

Les valeurs sont les mêmes qu'avec box-shadow

```
text-shadow: 2px 3px 2px #000;
```

X Offset    Y Offset    Blur    Color

```
text-shadow: 10px 10px 0px black;
```



**Text Shadow**

# Apparence dynamique - Au survol :hover

---

Quand la souris se trouve sur l'élément (lien, image, paragraphe, etc...)

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

# Lorsque le lien a déjà été visité :visited

---

```
a:visited  
{  
    color:blue;  
}
```

# Utilisation des préfixes

---

Afin de permettre le développement progressif et l'implémentation des modules CSS, les éditeurs de navigateurs ont la possibilité d'introduire le support de propriétés spécifiques.

Chaque moteur de navigateur utilise alors propre préfixe afin de limiter les interférences entre eux. Voici les différents préfixes connus :

- **-ms-** pour Microsoft (Internet Explorer, IE Mobile) ;
- **-moz-** pour les produits Mozilla (Firefox, Firefox Mobile) ;
- **-webkit-** pour Chrome, Safari, leurs versions mobiles et en règle générale tous les navigateurs basés sur le moteur Webkit ;
- **-o-** pour les navigateurs Opéra (Opéra, Opéra Mobile, Opéra Mini) ;



# Le principe

Ce principe permet aux navigateurs d'ajouter le support de fonctionnalités particulières sans attendre la publication définitive dans la norme, notamment CSS3, et de tester en situation réelle le rendu, souvent réclamé avec insistance par les développeurs web ou suggéré par l'équipe même de l'éditeur. Le but est aussi de permettre la migration vers la syntaxe finale en retirant uniquement le préfixe lors de la validation officielle de la propriété et des valeurs qu'elle peut accepter.

## Préfixe des propriétés CSS des différents navigateurs

- Les nouvelles propriétés CSS3 mettent du temps à être normalisées.
- Grâce aux « **préfixes** », les navigateurs peuvent supporter des fonctionnalités CSS particulières sans attendre la publication définitive d'une nouvelle version du standard.
- Elles se distinguent par un préfixe propre à chaque navigateur :

- **-webkit-** Chrome, Safari, Android...
- **-khtml-** Konqueror
- **-moz-** Mozilla
- **-ms-** Internet Explorer
- **-o-** Opera



# Le CSS et l'héritage

---

En CSS, si vous appliquez un style à une balise, toutes les balises qui se trouvent à l'intérieur prendront le même style.

Si j'applique une couleur de fond noire et une couleur de texte blanche à la balise `<body>`, tous mes titres et paragraphes auront eux aussi un arrière-plan de couleur noire et un texte de couleur blanche...

C'est ce phénomène qu'on appelle l'héritage : on dit que les balises qui se trouvent à l'intérieur d'une autre balise « héritent » de ses propriétés.

# Le CSS et l'héritage

---

```
body
{
    background-color: black;
    color: white;
}
p
{
    color: red;
}
```

C'est la propriété CSS de l'élément le plus précis qui a la priorité

Le même principe vaut pour toutes les balises HTML et toutes les propriétés CSS !

Le CSS décide que c'est la déclaration la plus précise qui l'emporte