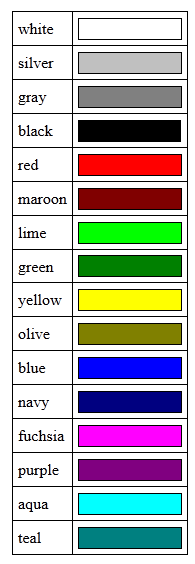
**AJOUTEZ DE LA COULEUR ET UN FOND**

**COULEUR DU TEXTE**

* Il y a déjà la propriété **color** qui permet de changer de couleur, mais il y a encore d’autres facons d’indiquer la couleur.

**INDIQUER LE NOM DE LA COULEUR**

* La méthode la plus simple est de taper son nom mais il n’existe que 16 couleurs dites “standard”, la figure ci-contre montre les seize couleurs “standard” que l’on peut en utiliser le nom : (en anglais bien sûr)
* Pour passer tous les titres en bordeaux, on peut donc écrire :

h1

{

**color**: maroon;

}

**LA NOTATION HEXADÉCIMALE**

* Cette méthode est couramment utilisée sur le web, un nom de couleur en hexadécimal est une combinaison de lettres et de chiffres qui indique une couleur, cela ressemble à **#FF5A28**
* On doit toujours commencer par écrire un **#** (diese) suivie de **6 lettres ou chiffres** allant **de 0 à 9 et de A à F .**
* Ces lettres ou chiffres fonctionnent **deux par deux** et **les premiers indiquent une quantité de rouge** et **les deux suivant une quantité de vert** et **les deux derniers une quantité de bleu**, en les mélangeant on peut obtenir la couleur que l’on souhaite.
* Donc **#000000** correspond à la **couleur noire** et **#FFFFFF** à la **couleur blanche**.
* Certains logiciels de dessin comme Photoshop, gimp et paint.net indiquent les couleurs en hexadécimal, c’est donc simple de copier collé le code hexadécimal d’une couleur dans le fichier CSS.
* Exemple avec les paragraphes appliqué avec la couleur blanche en hexadécimal :

p

{

**color**: #FFFFFF;

}

* Il existe une notation raccourcie que l’on peut écrire avec trois caractères, comme #FA3 équivaut à #FFAA33.

**LA MÉTHODE RGB**

* RGB signifie **Red-Green-Blue** (Rouge,Vert, Bleu)
* Comme avec la notation hexadécimal, on doit définir une quantité de rouge, de vert et de bleu pour choisir une couleur.
* C’est toutefois plus simple qu’avec la notation hexadécimal car la il suffit de se rendre sur un logiciel comme Paint, de se rendre dans la section “**modifier les couleurs**” alors une fenêtre s’ouvre, et dans la zone qui apparaît à droite il faut bouger les curseurs pour sélectionner la couleur qui intéresse.
* Il faut ensuite relever les **quantités de rouge-vert-bleu** correspondantes indiquées en bas à droite.
* Comme **240-96-204**, il faut recopier ces valeurs **dans l’ordre dans le fichier** CSS.
* Exemple :

p

{

**color**: rgb(**240**,**96**,**204**);

}

rgb(rouge, vert, bleu) : En remplaçant par les nombres correspondants.

* Toujours compris entre 0 et 255.
* Plusieurs sites Web permettent de récupérer une couleur en hexadécimal, comme : <https://www.webfx.com/web-design/color-picker/> .
* Des extensions de navigateurs permettent aussi de récupérer une couleur sur un site Web et sa notation hexadécimal comme : <https://addons.mozilla.org/fr/firefox/addon/colorpicker/> pour Firefox et <https://chrome.google.com/webstore/detail/colorzilla/bhlhnicpbhignbdhedgjhgdocnmhomnp> pour Google Chrome.

**COULEUR DE FOND**

* Pour indiquer une couleur de fond on utilise la propriété CSS **background-color**, elle s’utilise comme la propriété **color**, donc on peut taper le nom de la couleur, écrire en notation hexadécimal ou la méthode RGB.
* Pour indiquer la couleur de fond, il faut travailler sur la balise <**body**> car elle correspond à l’ensemble de la page Web, c’est donc en modifiant sa couleur de fond que l’on changera la couleur d'arrière-plan de la page Web.
* Exemple sur fichier CSS :

/\* On travaille sur la balise body, donc sur TOUTE la page \*/

body

{

**background-color**: black; /\* Le fond de la page sera noir \*/

**color**: white; /\* Le texte de la page sera blanc \*/

}

* Seulement, tous les paragraphes <**p**> et titres <**h1**> ont prit la couleurs blanc car ils se trouvent dans la balise <**body**>, ce phénomène se nomme **l’héritage**.

**LE CSS ET L'HÉRITAGE**

* En CSS, si on applique un style à une balise, toutes les balises qui se trouvent à l'intérieur prendront le même style.
* La balise <**body**> contient entre autres les balises de paragraphes <**p**> et de titres <**h1**>,
* Si une couleur de fond noire et une couleur de texte blanche est appliqué à la balise <**body**> alors tous les titres et paragraphes auront une couleur de fond noire et un texte de couleur blanche (l’héritage), car les balises qui se trouvent à l’intérieur d’une autre balise “héritent” de ses propriétés.
* C'est d'ailleurs de là que vient le nom « CSS », qui signifie « *Cascading Style Sheets* », c'est-à-dire « Feuilles de style en cascade ». Les propriétés CSS sont héritées en cascade : si on donne un style à un élément, tous les sous-éléments auront le même style.
* Exemple d’héritage avec la balise <**mark**>
* On a tendance à croire qu'on ne peut modifier que la couleur de fond de la page. C'est faux : vous pouvez changer le fond de n'importe quel élément : vos titres, vos paragraphes, certains mots… Dans ce cas, ils apparaîtront surlignés (comme si on avait mis un coup de marqueur dessus).
* On se souvient par exemple de la balise <**mark**> qui permet de mettre en valeur certains mots ? Utilisons-la à nouveau ici :

<h1>Qui a éteint la lumière ?</h1>

<p>Brr, il fait tout noir sur ce site, c'est un peu <**mark**>inquiétant</**mark**> comme ambiance non vous trouvez pas ?</p>

Par défaut, le texte s'affiche sur un fond jaune. On peut donc changer ce comportement en CSS :

body

{

**background-color**: black;

**color**: white;

}

mark

{

/\* La couleur de fond prend le pas sur celle de toute la page \*/

**background-color**: red;

**color**: black;

}

Sur le texte de la balise <**mark**> , c'est la couleur de fond rouge qui s'applique. En effet, même si le fond de la page est noir, 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 ! Si on dit :
* mes paragraphes ont une taille de 1.2 em ;
* mes textes importants ( <**strong**> ) ont une taille de 1.4 em ;

On pourrait penser qu'il y a un conflit. Le texte important fait partie d'un paragraphe, quelle taille lui donner ? 1.2 em ou 1.4 em ? Le CSS décide que c'est la déclaration la plus précise qui l'emporte : comme <**strong**> correspond à un élément plus précis que les paragraphes, le texte sera écrit en 1.4 em.

**IMAGES DE FOND**

* Dans les exemples qui suivent,on va modifier l'image de fond de la page. Cependant, tout comme pour la couleur de fond,il ne faut pas oublier que l'image de fond ne s'applique pas forcément à la page entière. On peut aussi mettre une image de fond derrière les titres, paragraphes, etc.

**APPLIQUER UNE IMAGE DE FOND**

* La propriété permettant d'indiquer une image de fond est **background-image** . Comme valeur, on doit renseigner **url**("**nom\_de\_l\_image.png**") . Par exemple :

body

{

**background-image**: **url**("neige.png");

}

* Bien entendu, le fond n'est pas forcément en **PNG**, il peut aussi être en **JPEG** ou en **GIF**.
* L'adresse indiquant où se trouve l'image de fond peut être écrite en absolu ( **http://…** ) ou en relatif ( **fond.png**  ).
* Attention lorsque l’on écrit une adresse en relatif dans le fichier CSS ! L'adresse de l'image doit être indiquée *par rapport au fichier .css* et non pas par rapport au fichier .html . Pour simplifier les choses,il est conseillé de placer l'image de fond dans le même dossier que le fichier .css (ou dans un sous-dossier).

**OPTION DISPONIBLE POUR L’IMAGE DE FOND**

* On peut compléter la propriété background-image que l’on vient de voir par plusieurs autres propriétés qui permettent de changer le comportement de l'image de fond.
* **background-attachment** : fixer le fond

La propriété CSS **background-attachment** permet de « fixer » le fond. L'effet obtenu est intéressant, car on voit alors le texte « **glisser** » par-dessus le fond. Deux valeurs sont disponibles :

* **fixed** : L’image de fond reste fixe ;
* **scroll** : L’image de fond défile avec le texte (par défaut).
* Exemple :

body

{

**background-image**: **url**("neige.png");

**background-attachment**: **fixed**; /\* Le fond restera fixe \*/

}

* **background-repeat** : répétition du fond

Par défaut, l'image de fond est répétée en mosaïque. Vous pouvez changer cela avec la propriété background-repeat :

* **no-repeat** : Le fond ne sera pas répété, l’image sera donc unique sur la page ;
* **repeat-x** : Le fond sera répété uniquement sur la première ligne, horizontalement ;
* **repeat-y** : Le fond sera répété uniquement sur la première colonne, verticalement ;
* **repeat** : Le fond sera répété en mosaïque (par défaut).
* Exemple :

body

{

**background-image**: **url**("soleil.png");

**background-repeat**: **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**; (un fond qui ne se répète pas).

* On doit donner à **background-position** deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe, si on applique le fond à un paragraphe). Ainsi, si vous tapez :

**background-position**: 30px 50px;

Le fond sera placé à **30 pixels** de la gauche et à 5**0 pixels** du haut. Il est aussi possible d'utiliser ces valeurs en anglais :

* **top** : En haut ;
* **bottom** : En bas ;
* **left** : À gauche ;
* **center** : Centré ;
* **right** : À droite.

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

**background-position**: **top right**;

* Ainsi, si je veux afficher un soleil en image de fond, en un unique exemplaire ( no-repeat ), toujours visible ( fixed ) et positionné en haut à droite ( top right ), je vais écrire ceci :

body

{

**background-image**: **url**("soleil.png");

**background-attachment**: **fixed**; /\* Le fond restera fixe \*/

**background-repeat**: **no-repeat**; /\* Le fond ne sera pas répété \*/

**background-position**: **top right**; /\* Le fond sera placé en haut à droite \*/

}

**COMBINER LES PROPRIÉTÉS**

* Si on utilise 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** et **background-position**.

On peut donc tout simplement écrire :

body

{

**background**: **url**("soleil.png") **fixed no-repeat** **top right**;

}

C'est la première « super-propriété » que je vous montre, il y en aura d'autres. Il faut savoir que :

* L’ordre des valeurs n'a pas d'importance, on peut combiner les valeurs dans n'importe quel ordre ;
* On est pas obligé de mettre toutes les valeurs. Ainsi, si on ne veut pas écrire fixe, on peut l'enlever sans problème.

**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**("soleil.png") **fixed** **no-repeat** **top right**, **url**("neige.png") **fixed**;

}

* La première image de cette liste sera placée par-dessus les autres. Attention donc, l'ordre de déclaration des images a son importance : si on inverse le soleil et la neige dans le code CSS précédent, on ne verra plus le soleil !
* À noter que les images de fond multiples fonctionnent sur tous les navigateurs sauf sur les anciennes versions d'Internet Explorer, qui ne reconnaît cette fonctionnalité qu'à partir de la version 9 (IE9).

Une dernière chose avant d'en terminer avec les images de fond : dans tous ces exemples, j'ai appliqué un fond à la page entière ( body ). Mais cela ne doit pas vous faire oublier qu'on peut appliquer un fond à n'importe quel élément (un titre, un paragraphe, certains mots d'un paragraphe, etc.).

Je vous conseille donc, pour vous entraîner, d'essayer d'appliquer un fond à vos titres ou paragraphes. Si vous avez un peu de goût (contrairement à moi !) vous arriverez certainement à donner une très belle allure à votre page web.

**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**.

**LA PROPRIÉTÉ OPACITY**

* La propriété **opacity**  , très simple, permet d'indiquer le niveau d'**opacité** (c'est l'inverse de la transparence).
* 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.

Il faut donc choisir une valeur comprise entre 0 et 1. Ainsi, avec une valeur de 0.6, l’élément sera opaque à 60 %… et on verra donc à travers !

Exemple :

p

{

**opacity**: 0.6;

}

* Voici un exemple qui va nous permettre d'apprécier la transparence. On trouvera le rendu à la figure suivante.

body

{

**background**: url('neige.png');

}

p

{

**background-color**: black;

**color**: white;

**opacity**: 0.3;

}

* Notez que la transparence fonctionne sur tous les navigateurs récents, y compris Internet Explorer à partir de IE9.

Si vous appliquez la propriété **opacity** à un élément de la page, *tout* le contenu de cet élément sera rendu transparent (même les images, les autres blocs à l'intérieur, etc.). Si vous voulez juste rendre la couleur de fond transparente, utilisez plutôt la notation **RGBa** que nous allons découvrir.

**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.

Exemple :

p

{

**background-color**: rgba(255, 0, 0, 0.5); /\* Fond rouge à moitié transparent \*/

}

* C'est aussi simple que cela, on peut obtenir exactement le même effet qu'avec **opacity** juste en jouant avec la notation **RGBa**.
* Cette notation est connue de tous les navigateurs récents, y compris Internet Explorer (à partir de IE9). Pour les navigateurs plus anciens, il est recommandé d'indiquer la notation **RGB** classique en plus de **RGBa**.
* Pour ces navigateurs, le fond ne sera alors pas transparent mais, au moins, il y aura bien une couleur d'arrière-plan.

Exemple :

p

{

**background-color**: rgb(255,0,0); /\* Pour les navigateurs anciens \*/

**background-color**: rgba(255,0,0,0.5); /\* Pour les navigateurs plus récents \*/

}

**CRÉEZ DES BORDURES ET DES OMBRES**

**BORDURES STANDARD**

* Le CSS offre un large choix de bordures pour décorer la page, de nombreuses propriété CSS permettent de modifier l’apparence des bordures :

**border-width** ;

**border-color** ;

**border-style**.

* Pour aller à l'essentiel, on peut utiliser directement la super-propriété **border** qui regroupe l’ensemble de ces propriétés. Cela fonctionne sur le même principe que la propriété **background**, on peut combiner plusieurs valeurs.
* Pour **border**, on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

**La largeur** : Indique la largeur de la bordure, mettre une valeur en pixels (comme 2px) ;

**La couleur** : C’est la couleur de la bordure, à utiliser comme appris précédemment, soit un nom de couleur (black, red, ….), soit une valeur en hexadécimal (#FFF000) ou alors une valeur RGB (rgb(128,212,37)) ;

**Le type de bordure** : La on a le choix, la bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc. Voici les différentes valeurs disponibles :

* **none** : Pas de bordure (par défaut) ;
* **solid** : Un trait simple ;
* **dotted**  : Pointillés ;
* **dashed** : Tirets ;
* **double** : Bordure double ;
* **groove** : En relief ;
* **ridge** : Autre effet relief ;
* **inset** : Effet 3D global enfoncé ;
* **outset** : Effet 3D global surélevé.

- Ainsi, pour avoir une bordure bleue, en tirets, épaisse de 3 pixels autour de mes titres, je vais écrire :

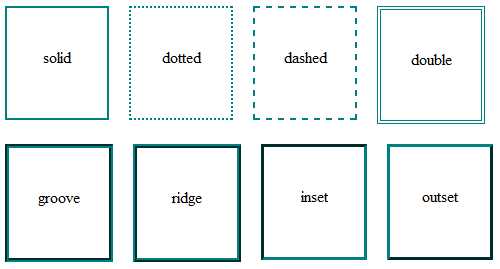
h1

{

**border**: 3px blue dashed;

}

- La figure suivante présente les différents styles de bordure que l’on peut utiliser.



**EN HAUT, A DROITE, A GAUCHE, EN BAS...**

* Pas obligé d’appliquer la même bordure aux quatres côtés de notre élément. Si on veut mettre des bordures différentes en fonction du côté (haut, bas, gauche ou droite), on peut le faire sans problème
* On doit dans ce cas utiliser ces quatres propriétés :

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

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

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

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

* Il existe aussi des équivalents pour paramétrer chaque détail de la bordure si on le désire : **border-top-width**  pour modifier l'épaisseur de la bordure du haut, **border-top-color** pour la couleur du haut, etc.
* Ce sont aussi des super-propriétés, elles fonctionnent comme border mais ne s'appliquent donc qu'à un seul côté.
* Pour ajouter une bordure uniquement à gauche et à droite des paragraphes, on écrira donc :

p

{

**border-left**: 2px solid black;

**border-right**: 2px solid black;

}

* On peut modifier les bordures de n'importe quel type d'élément sur la page. Nous l'avons fait ici sur les paragraphes, mais on peut aussi modifier la bordure des images, des textes importants comme <**strong**>, etc.

**BORDURES ARRONDIES**

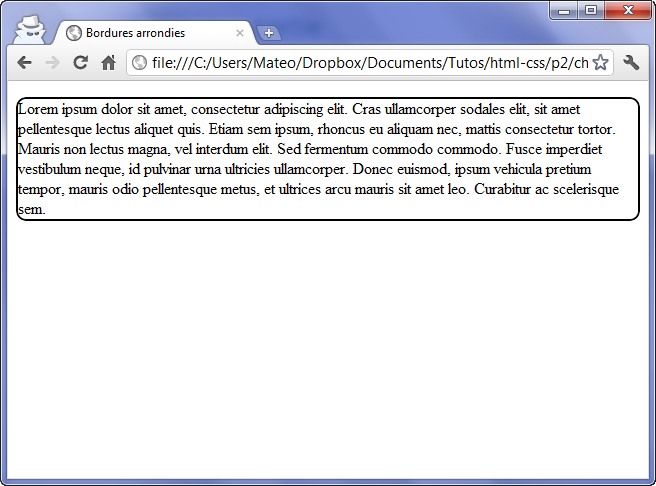
* Les bordures arrondies, c'est un peu le Saint Graal attendu par les webmasters depuis longtemps. Depuis que CSS3 est arrivé, il est enfin possible d'en créer facilement.
* La propriété **border-radius** va permettre d'arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer la taille (« l'importance ») de l'arrondi en pixels :

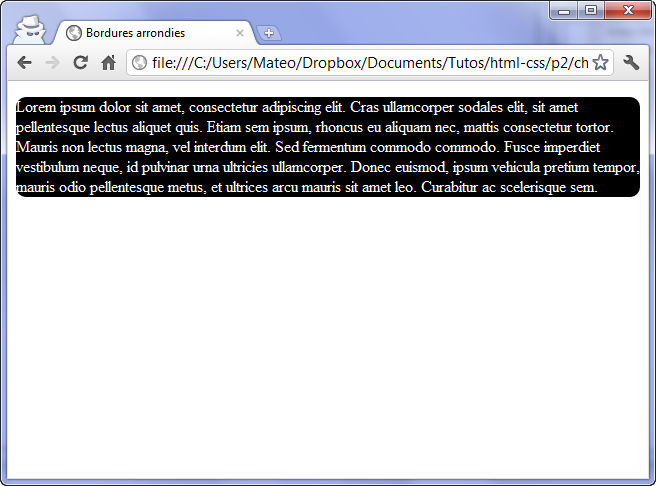
p

{

**border-radius**: 10px;

}

* L'arrondi se voit notamment si l'élément a des bordures, comme sur la figure suivante :
* Ou s'il a une couleur de fond, comme sur la figure suivante :



* On peut aussi préciser la forme de l'arrondi pour chaque coin. Dans ce cas, il faut indiquer quatre valeurs :

p

{

**border-radius**: 10px 5px 10px 5px;

}

* Les valeurs correspondent aux angles suivants dans cet ordre :

1- En haut à gauche.

2- En haut à droite.

3- En bas à droite.

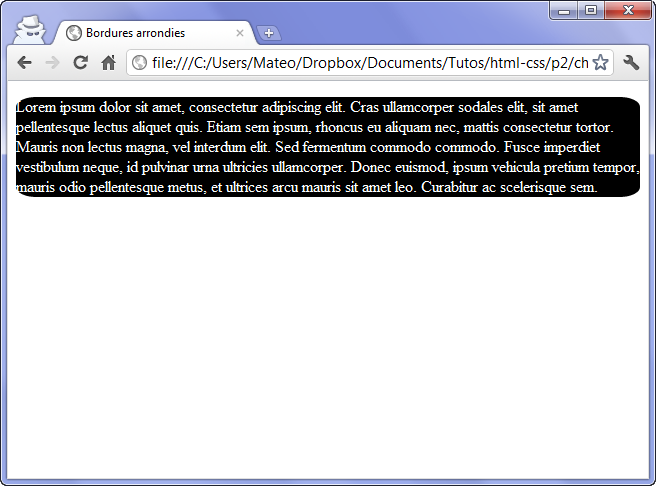
4- En bas à gauche.

* Il est possible d'affiner l 'arrondi de nos angles en créant des courbes elliptiques (figure suivante). Dans ce cas, il faut indiquer deux valeurs séparées par une barre oblique (*slash*, caractère /). Le mieux est certainement de tester pour voir l'effet :

p

{

**border-radius**: 20px / 10px;

}

**Bordures arrondies elliptiques**

**LES OMBRES**

* Les ombres font partie des nouveautés récentes proposées par CSS3. Aujourd'hui, il suffit d'une seule ligne de CSS pour ajouter des ombres dans une page !
* Il y a deux types d’ombres :

Les ombres des boîtes ;

Les ombres du texte.

**Les ombres des boîtes**

**(box-shadow)**

* La propriété **box-shadow** s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :

1- Le décalage horizontal de l’ombre.

2- Le décalage vertical de l’ombre.

3- L'adoucissement du dégradé.

4- La couleur de l’ombre.

Par exemple, pour une ombre noire de 6 pixels, sans adoucissement, on écrira :

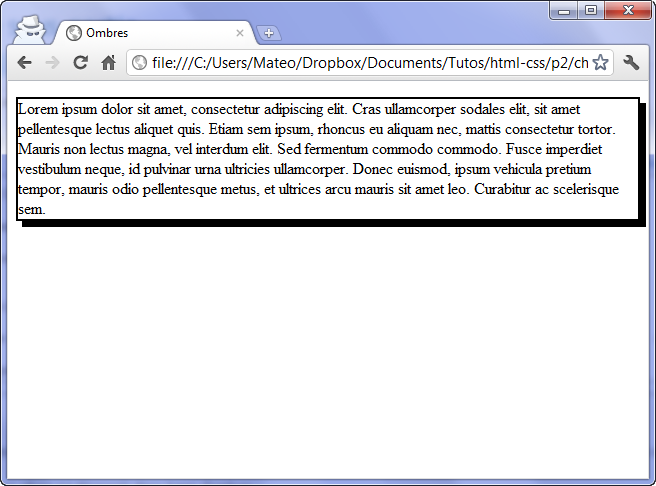
p

{

box-shadow: 6px 6px 0px black;

}

* Cela donne le résultat illustré à la figure suivante (j'ai ajouté une bordure au paragraphe pour qu'on voie mieux l'effet).

**Une ombre sous le paragraphe**

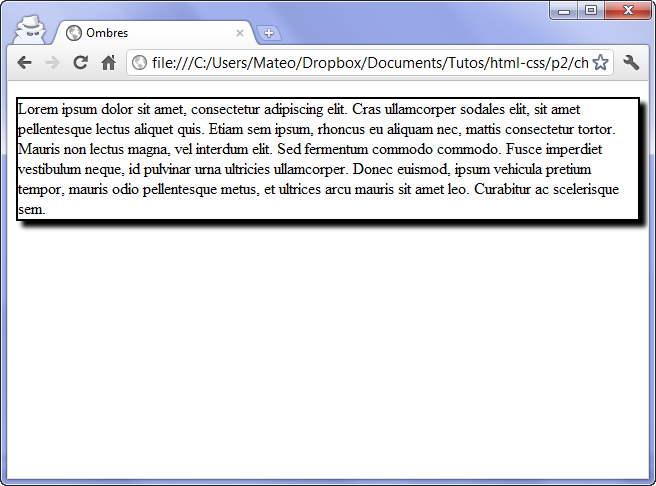
Ajoutons un adoucissement grâce au troisième paramètre (figure suivante). L'adoucissement peut être faible (inférieur au décalage), normal (égal au décalage) ou élevé (supérieur au décalage). Essayons un décalage normal :

p

{

**box-shadow**: 6px 6px 6px black;

}

**Une ombre adoucie sous le paragraphe**

* 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 :

p

{

**box-shadow**: 6px 6px 6px black inset;

}

**L’ombre du texte**

**(text-shadow)**

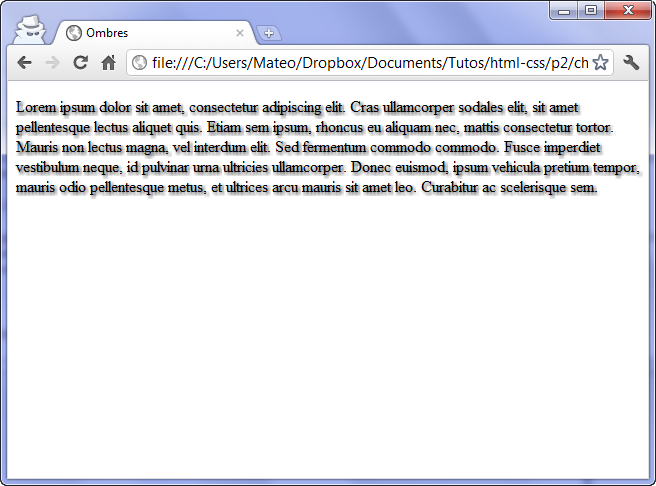
* Avec **text-shadow**, on peut ajouter une ombre directement sur les lettres de son texte ! Les valeurs fonctionnent exactement de la même façon que **box-shadow** : décalage, adoucissement et couleur.

p

{

**text-shadow**: 2px 2px 4px black;

}

Le résultat est illustré à la figure suivante.

**Texte ombré**

**Créez des apparences dynamiques**

* Le CSS nous permet aussi de modifier l'apparence des éléments de façon dynamique, c'est-à-dire que des éléments peuvent changer de forme une fois que la page a été chargée. On va faire appel à une fonctionnalité puissante du CSS : les pseudo-formats.
* On va voir comment changer l'apparence :

Au survol ;

Lors du clic ;

Lors du focus (élément sélectionné) ;

Lorsqu’un lien a été consulté.

**Au survol**

* On va découvrir plusieurs pseudo-formats CSS. L’un s’appelle **:hover**. Comme tous les autres pseudo-formats, c’est une information que l’on rajoute après le nom de la balise (ou de la classe) dans le CSS, comme ceci :

a**:hover**

{

}

* **:hover** signifie “survoler”, **a:hover** peut donc se traduire par : “Quand la souris est sur le lien” (quand on pointe dessus).
* A partir de là, on doit définir l’apparence que doivent avoir les liens lorsqu’on pointe dessus. Il faut laisser libre cours à notre imagination, il n’y a pas de limite.
* Voici un exemple de présentation des liens, mais il ne faut pas hésiter à inventer le nôtre :

a /\* Liens par défaut (non survolés) \*/

{

**text-decoration**: none;

**color**: red;

**font-style**: italic;

}

**a:hover** /\* Apparence au survol des liens \*/

{

**text-decoration**: underline;

**color**: green;

}

On a défini ici deux versions des styles pour les liens :

-Pour les liens par défaut (non survolés) ;

-Pour les liens au survol.

- Le résultat se trouve à la figure suivante.

**Changement d’apparence au survol de la souris**

* Même si on l'utilise souvent sur les liens, vous pouvez modifier l'apparence de n'importe quel élément. Par exemple, vous pouvez modifier l'apparence des paragraphes lorsqu'on pointe dessus :

p:hover /\* Quand on pointe sur un paragraphe \*/

{

}

**AU CLIC ET LORS DE LA SÉLECTION**

On peut interagir encore plus finement en CSS. On va voir ici que nous pouvons changer l'apparence des éléments lorsque l'on clique dessus et lorsqu'ils sont sélectionnés !

**AU MOMENT DU CLIC**

**(:active)**

* Le pseudo-format **:active** permet d’appliquer un style particulier au moment du clic. En pratique, il n’est utilisé que sur les liens.
* Le lien gardera cette apparence très peu de temps : en fait, le changement intervient lorsque le bouton de la souris est enfoncé. En clair, ce n'est pas forcément toujours bien visible.
* On peut par exemple changer la couleur de fond du lien lorsque l’on clique dessus :

**a:active** /\* Quand le visiteur clique sur le lien \*/

{

**background-color**: #FFCC66;

}

**(:focus) : LORSQUE L'ÉLÉMENT EST SÉLECTIONNÉ**

* La c’est un peu différent. Le pseudo-format **:focus** applique un style lorsque l'élément est sélectionné.
* Une fois que l’on a cliqué, le lien reste « sélectionné » (il y a une petite bordure en pointillés autour). C'est cela, la sélection.
* Ce pseudo-format pourra être appliqué à d'autres balises HTML, comme les éléments de formulaires.
* Sur les liens

**a:focus** /\* Quand le visiteur sélectionne le lien \*/

{

**background-color**: #FFCC66;

}

* Sous Google Chrome et Safari, l'effet ne se voit que si l'on appuie sur la touche TAB.

**LORSQUE LE LIEN A DÉJÀ ÉTÉ CONSULTÉ**

* Il est possible d'appliquer un style à un lien vers une page qui a déjà été vue. Par défaut, le navigateur colore le lien en un violet assez laid,
* On peut changer cette apparence avec **:visited** (qui signifie « visité »). En pratique, sur les liens consultés, on ne peut pas changer beaucoup de choses à part la couleur (figure suivante).

**a:visited** /\* Quand le visiteur a déjà vu la page concernée \*/

{

**color**: #AAA; /\* Appliquer une couleur grise \*/

}

Liens visité en gris

* Si l’on ne souhaite pas que les liens déjà visités soient colorés d’une façon différente, il faudra leur appliquer la même couleur qu’aux liens normaux. De nombreux sites Web font cela (OpenClassrooms y compris !). Une exception notable : Google… ce qui est plutôt pratique, puisque l'on peut voir dans les résultats d'une recherche si on a déjà consulté ou non les sites que Google nous présente.