



# CSS 3

## Table des matières

|   |    |
|---|----|
| 1.1 - Généralités.....  | 3  |
| 1.1.1 - Nouveautés.....   | 3  |
| 1.1.2 - Syntaxe.....  | 4  |
| 1.2 - Compatibilités.....   | 5  |
| 1.3 - Le fond.....  | 6  |
| 1.3.1 - background-size.....  | 6  |
| 1.3.2 - background-clip.....  | 10 |
| 1.3.3 - background multiple.....                                    | 11 |
| 1.3.4 - background gradué.....                                      | 13 |
| 1.4 - Les ombrages.....   | 14 |
| 1.4.1 - box-shadow.....   | 14 |
| 1.4.2 - text-shadow.....  | 16 |
| 1.5 - Les contours.....   | 17 |
| 1.6 - Les coins arrondis.....                                       | 19 |
| 1.7 - Les transitions.....  | 26 |
| 1.7.1 - Présentation.....   | 26 |
| 1.7.2 - Exemples élémentaires.....                                  | 27 |
| 1.7.3 - Le délai.....   | 29 |
| 1.7.4 - La modification de l'interpolation.....                     | 30 |
| 1.7.5 - L'utilisation du raccourci.....                             | 32 |
| 1.7.6 - L'utilisation de code JavaScript sans pseudo-code CSS3..... | 33 |
| 1.8 - Transformations.....  | 34 |
| 1.9 - Opacité.....  | 35 |
| 1.10 - Le Responsive Web Design ou les Media Queries.....           | 37 |
| 1.10.1 - Objectif.....  | 37 |
| 1.10.2 - Syntaxe.....   | 38 |
| 1.10.3 - Exemple.....   | 39 |
| 1.10.4 - Exercice.....  | 42 |
| 1.11 - Menu Hamburger.....  | 43 |

## 1.1 - Généralités

---

### 1.1.1 - Nouveautés

Les nouveaux sélecteurs : nth-child(even), nth-child(odd), :not, first-child ...

Nouveau support des polices : @font-face ...

Text wrapping: text-overflow : ellipsis ...

Gestion de colonnes : column-count, column-rule, column-gap

Bordures sur texte : text-fill-color, text-stroke-color, text-stroke-width

**Opacité** : rgba()

Saturation, luminosité : hsla()

**Bordures arrondies** : border-radius

Fond dégradés en CSS : gradient

**Ombres** : text-shadow et box-shadow

**Background** : Ajustement de la taille en fonction de la div, backgrounds multiples ...

**Transitions** : transition

Transformation et rotations d'éléments HTML : transform

Animations

En septembre 2012 toutes les fonctionnalités ne sont pas implémentées dans les dernières versions des navigateurs.

**Quelques exemples :**

<http://www.jonathan-menet.fr/blog/2010/08/07/les-nouveautes-de-css3-en-20-exemples/>

---

### 1.1.2 - Syntaxe

Comme en CSS3 :

propriété : valeur;

Lorsque la propriété n'est pas implémentée selon le W3C il faut la préfixer.

-webkit- pour WebKit (Chrome, Safari)






moz- pour Gecko (Mozilla)

-ms- pour Microsoft

-o- pour Opera

et parfois pour MSIE c'est plus complexe.

## 1.2 - Compatibilités

|                   |  |  |  |  |  |
|-------------------|---|---|---|---|--|
| Le fond           | OK  | OK  | OK  | OK  |  |
| Bordures ombrées  |   |   |   |   |  |
| Texte ombré       |   |   |   |   |  |
| Contour           |   |   |   | OK  |  |
| Coins arrondis    |   |   |   |   |  |
| Transition        |   |   |   |   |  |
| Multiple colonnes |   |   |   |   |  |
| Redimensionnement |   |   |   |   |  |

<http://caniuse.com/#cats=CSS>

[http://www.w3schools.com/cssref/css3\\_browsersupport.asp](http://www.w3schools.com/cssref/css3_browsersupport.asp)

## 1.3 - Le fond

### 1.3.1 - background-size

La propriété CSS background-size spécifie la taille de l'image dans l'arrière plan.

```
background-size: x y | cover | contain ;
```

x détermine la dimension horizontale (pixels, em, auto, pourcentage %, etc),

y détermine la dimension verticale (pixels, em, auto, pourcentage %, etc)

cover : l'image couvrira toute la surface sans déformer l'image. Elle sera éventuellement rognée.

contain : forcera l'image à ne pas dépasser l'élément sans la déformer.

#### **La taille de l'image est égale au conteneur :**

cover : rien

contain : rien

x y égaux à la taille du conteneur : rien

x y différents de la taille du conteneur : si < mosaïque, si > rognage.

#### **La taille de l'image est supérieure au conteneur :**

cover : redimensionnement sans déformation

contain : si les proportions ne sont pas les mêmes alors mosaïque

x y égaux à taille du conteneur : rien

x y différents de la taille du conteneur : si < mosaïque, si > rognage et déformation.

#### **La taille de l'image est inférieure au conteneur :**

cover : agrandissement sans déformation; rognage si les proportions ne sont pas les mêmes

contain : mosaïque

x y égaux à taille du conteneur : redimensionnement de l'image et éventuellement déformation

x y différents de la taille du conteneur : si < mosaïque, si > rognage et déformation.

## Exemples :

### fonds.html

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

```
|<div class="divImg"></div>
```

### fonds.css

Taille du conteneur = taille de l'image et cover ou contain

```
|.divImg {  
|  width: 190px;  
|  height: 250px;  
|  /* L'original est de 190/250 */  
|  background-image: url(http://localhost/images/cinema/salma_hayek.jpg);  
|  background-size: contain;  
|  border: 1px solid black;  
|}
```



Taille du conteneur > taille de l'image sans disproportion et cover ou contain

```
.divImg {  
  width: 285px;  
  height: 375px;  
  /* L'original est de 190/250 */  
  background-image: url(http://localhost/images/cinema/salma_hayek.jpg);  
  background-size: cover;  
  border: 1px solid black;  
}
```






Taille du conteneur < taille de l'image sans disproportion et cover ou contain

```
.divImg {  
  width: 95px;  
  height: 125px;  
  /* L'original est de 190/250 */  
  background-image: url(../images/salma_hayek.jpg);  
  background-size: contain;  
  border: 1px solid black;  
}
```





Taille du conteneur < taille de l'image avec disproportion et cover ou contain ou dimensions.

| Cover   | Contain  | 95px 125px  |
|---|--|---|
|  |  |  |

```
.divImg {  
  width: 250px;  
  height: 300px;  
  /* L'original est de 190/250 */  
  background-image: url(../images/salma_hayek.jpg);  
  background-size: contain;  
  border: 1px solid black;  
}
```

---

### 1.3.2 - background-clip

**Objectif :** attacher l'image au conteneur.

**Syntaxe :**

`background-clip: content-box | border-box | padding-box`

`content-box` : le fond est égal au contenu

`border-box` : le fond s'étend jusqu'à la limite externe de la bordure

`padding-box` : le fond remplit jusqu'à la bordure et donc s'étend au padding

**Exemples :**

| content-box  | border-box   | padding-box  |
|--|--|--|
|  |  |  |

`<div class="divImgClip"></div>`

```
.divImgClip {
  width: 190px;
  height: 250px;
  /* L'original est de 190/250 */
  background-image: url(../images/salma_hayek.jpg);
  background-clip: padding-box; /* content-box | border-box | padding-box */
  background-size: cover;
  border: 5px dotted silver;
  padding: 10px;
}
```

---

### 1.3.3 - background multiple

**Objectif :** visualiser plusieurs images dans l'arrière-plan.

**Syntaxes :**

```
background-image: url(url1), url(url2), ...;  
background-position: positionY1 positionX1, positionY2 positionX2, ...;  
background-repeat: répétition, ...;
```

ou

```
background:  
  url(url1) positionX1 positionY1 répétition,  
  url(url2) positionX2 positionY2 répétition,  
  ...
```

positions : top, bottom, left, right

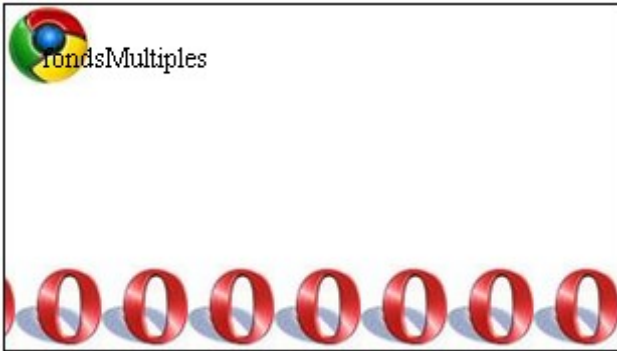
repeat : no-repeat | repeat-x | repeat-y | repeat

**Exemples :**

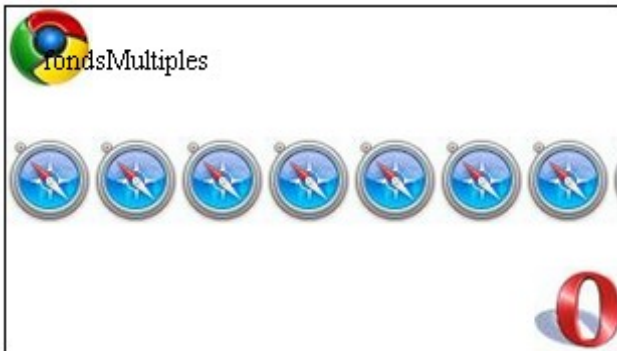
**fondsMultiples.html**

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>fondsMultiples</title>  
    <meta name="viewport" content="user-scalable=no, width=device-width,  
initial-scale=1.0, maximum-scale=1.0" />  
    <meta charset="utf-8" />  
    <link rel="stylesheet" href="../css/fondsMultiples.css" />  
  </head>  
  
  <body>  
    <div class="fondsMultiples">  
      fondsMultiples  
    </div>  
  </body>  
</html>
```

## fondsMultiples.css



```
.fondsMultiples {  
    width: 300px;  
    height: 150px;  
    padding: 20px;  
    border: 1px solid black;  
  
    background-image: url("../images/chrome.png"), url("../images/opera.png");  
    background-position: top left, left bottom;  
    background-repeat: no-repeat, repeat-x;  
}
```



```
/* fondsMultiples.css */  
  
.fondsMultiples {  
    width: 300px;  
    height: 150px;  
    padding: 20px;  
    border: 1px solid black;  
  
    background:  
        url("../images/chrome.png") top left no-repeat,  
        url("../images/safari.png") left repeat-x,  
        url("../images/opera.png") bottom right no-repeat;  
}
```

---

### 1.3.4 - background gradué

**Objectif :** coloriser de façon graduée le fond d'un élément.

**Syntaxe W3C :**

```
type-gradient();
```

Ce site permet de générer le code CSS pour tous les navigateurs en fonction d'un choix de dégradé.

<http://www.colorzilla.com/gradient-editor/>

**Exemple :**



**Code HTML**

```
<p class="bouton"></p>
```

**Code CSS**

```
.bouton {  
  /* webkit */  
  background: -webkit-gradient(linear, left top, left bottom, color-  
    stop(0%, rgba(245,246,246,1)), color-stop(21%, rgba(219,220,226,1)), color-  
    stop(49%, rgba(184,186,198,1)), color-stop(80%, rgba(221,223,227,1)), color-  
    stop(100%, rgba(245,246,246,1)));  
  
  /* Opera */  
  background: -o-linear-gradient(top, rgba(245,246,246,1) 0%, rgba(219,220,226,1)  
    21%, rgba(184,186,198,1) 49%, rgba(221,223,227,1) 80%, rgba(245,246,246,1)  
    100%);  
  
  /* W3C KO */  
  background: linear-gradient(to bottom, rgba(245,246,246,1) 0%,  
    rgba(219,220,226,1) 21%, rgba(184,186,198,1) 49%, rgba(221,223,227,1) 80%,  
    rgba(245,246,246,1) 100%);  
  
  width: 200px;  
  height: 50px;  
  border: 1px solid black;  
}
```

## 1.4 - Les ombrages

### 1.4.1 - box-shadow

#### Syntaxe

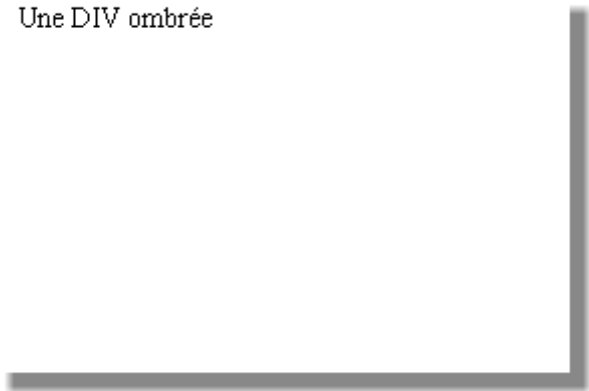
```
box-shadow: dh dv fd couleur;
```

dh : décalage horizontal vers la droite,  
dv : décalage vertical vers le bas,  
fd : force du dégradé (avec 0px pas de dégradé de l'ombre).

La propriété s'applique sur la boîte de l'élément, et non sur sa bordure. L'ombrage n'affecte pas la taille de la boîte de l'élément. L'ombre est donc à l'extérieur de l'élément.

#### Exemple avec une <div>

Une DIV ombrée



## Code CSS

```
/* borduresOmbrees.css */


```

## Code HTML

```
<!DOCTYPE html>
<!-- borduresOmbrees.html -->
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="stylesheet" href="../../css/borduresOmbrees.css" />
    </head>

    <body>
        <div class="divOmbree">
            Une DIV ombrée
        </div>
    </body>
</html>
```

# Un texte ombré

## Syntaxe

```
text-shadow: dh dv fd couleur;
```

Identique à box-shadow pour les paramètres.

## Le style

```
.texteOmbre {  
    text-shadow: 5px 5px 10px #777;  
}
```

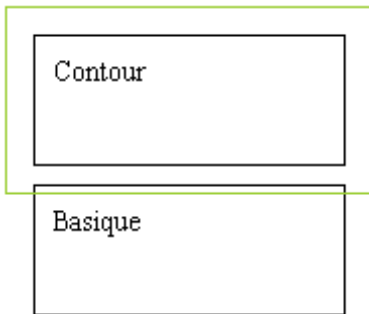
## Le contenu

```
<div>  
    <h1 class="texteOmbre">Un texte ombré</h1>  
</div>
```



## 1.5 - Les contours

**Objectif** : ajouter un pourtour à un élément.



### Syntaxe

```
outline: épaisseur couleur;  
outline-offset: déplacement en unités de mesure;
```

Note : outline ne prend pas de place.

## Le code HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>contour</title>
    <meta name="viewport" content="user-scalable=no, width=device-width,
initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <link rel="stylesheet" href="../../css/contour.css" />
  </head>

  <body>
    <div class="contour">Contour</div>

    <div class="basique">Basique</div>
  </body>
</html>
```

## Le code CSS

```
/* contour.css */

body {padding: 10px; }

.basique {
  width: 150px;
  height: 50px;
  margin: 10px;
  padding: 10px;
  border: 2px solid black;
}

.contour {
  width: 150px;
  height: 50px;
  margin: 10px;
  padding: 10px;
  border: 2px solid black;

  outline: 2px solid yellowgreen;
  outline-offset: 15px;
}
```

## 1.6 - Les coins arrondis

**Syntaxe W3C :**

```
border-radius: taille; /* Tous de la meme taille */
```

ou

```
border-radius: hg hd bd bg; /* Tous de tailles differentes */
```

ou

```
border-radius: hg-bd hd-bg; /* La diagonale */
```

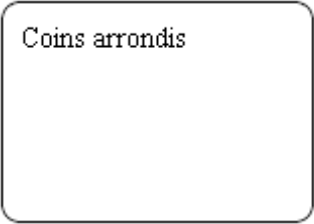
hg :haut gauche

hd : haut droit


bd : bas droit

bg : bas gauche

**Exemple :**



Coins arrondis



Onglet

## coinsArrondis.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>coinsArrondis.html</title>
    <meta name="viewport" content="user-scalable=no, width=device-width,
initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <link rel="stylesheet" href="../../css/coinsArrondis.css" />
  </head>

  <body>
    <div class="coinsArrondis">
      Coins arrondis
    </div>

    <h3 class="onglet">
      Onglet
    </h3>
  </body>
</html>
```

## coinsArrondis.css

```
/* coinsArrondis.css */

.coinsArrondis {
    background-color: white;
    border: 2px solid black;
    padding: 10px;

    /* Arrondir les coins */

    /* Tous de la meme taille */
    border-radius: 10px;

    /* Tous de tailles differentes; ca commence en haut a gauche et dans le
sens des aiguilles d'une montre */
/*    border-radius: 0 10px 20px 5px;*/

    /* Par la diagonale haut-gauche et bas-droit puis haut-droit et bas-gauche
*/
/*    border-radius: 10px 0;*/

    width: 150px;
    height: 100px;
}

.onglet {
    background-color: white;
    border: 1px solid black;
    padding: 10px;

    /* Arrondir le coin en haut a droite */
    border-radius: 0 10px 0 0;

    width: 100px;
    height: 20px;
}
```

## Le cas des tables

Comme il est impossible d'appliquer le code précédent aux éléments d'un <table> (<thead>, <tr>, <th>, <tr>, ...).

La propriété display :

Cette propriété peut prendre une foule de valeurs.

Celles qui concernent la mise en forme en table sont :

table, table-cell et table-row, table-caption, table-column, table-column-group, table-header-group, table-footer-group, table-row-group.

### TableSansTable

| Tarifs             |             |                |          |       |
|--------------------|-------------|----------------|----------|-------|
| Prestation         | Prix unique | Cheveux courts | Mi-longs | Longs |
| Coupe / Brushing   |             | 25 €           | 30 €     | 35 €  |
| Coupe / Shampooing |             | 20 €           | 25 €     | 30 €  |
| Pied de table      |             |                |          |       |

```
<!DOCTYPE html>
<!--
TableSansTable.html
-->
<html>
  <head>
    <title>TableSansTable</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">

    <style>
      #tableTarifs {
        display: table;
        /*          margin:auto;*/
      }
      .chapeauTable{
        display: table-caption;
        border: black solid thin;
        text-align: center;
      }
      .piedTable{
        display: table-footer-group;
        border: black solid thin;
        text-align: center;
      }
      .ligne {
        display:table-row;
      }
      .celluleEntete{
        display: table-cell;
        width: 100px;
        border: black solid thin;
        text-align: center;
        padding: 5px;
      }
      .celluleEnteteLarge300{
        display: table-cell;
        width: 300px;
```

```

        border: black solid thin;
        padding: 5px;
    }
    .cellule{
        display: table-cell;
        width: 100px;
        border-left: black dashed thin;
        border-right: black dashed thin;
        border-bottom: black dashed thin;
        padding: 5px;
    }
    .celluleLarge300{
        display: table-cell;
        width: 300px;
        border-left: black dashed thin;
        border-right: black dashed thin;
        border-bottom: black dashed thin;
        padding: 5px;
    }
    .coinsArrondisHautGauche {
        /* Arrondir les coins : Le haut gauche */
        padding: 5px;
        border-top-left-radius: 5px;
    }
    .coinsArrondisHautDroite{
        /* Arrondir les coins : Le haut droite */
        padding: 5px;
        border-top-right-radius: 5px;
    }
}
</style>
</head>

<body>
    <div>
        <h3>TableSansTable</h3>
        <div id="tableTarifs">
            <div class="chapeauTable coinsArrondisHautGauche
coinsArrondisHautDroite">
                Tarifs
            </div>
            <!-- entetes -->
            <div class="ligne">
                <span class="celluleEntete celluleEnteteLarge300">
                    Prestation
                </span>
                <span class="celluleEntete">
                    Prix unique
                </span>
                <span class="celluleEntete">
                    Cheveux courts
                </span>
                <span class="celluleEntete">
                    Mi-longs
                </span>
                <span class="celluleEntete">
                    Longs
                </span>
            </div><!-- Fin ligne-entete -->

            <div class="ligne">
                <!-- Ligne -->
                <span class="cellule celluleLarge300">
                    Coupe / Brushing
                </span>
            </div>
        </div>
    </div>

```

```

        <span class="cellule">

        </span>
        <span class="cellule">
            25 &euro;
        </span>
        <span class="cellule">
            30 &euro;
        </span>
        <span class="cellule">
            35 &euro;
        </span>
    </div><!-- Fin ligne-->

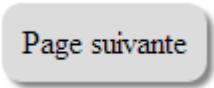
    <div class="ligne">
        <!-- Ligne -->
        <span class="cellule celluleLarge300">
            Coupe / Shampooing
        </span>
        <span class="cellule">

        </span>
        <span class="cellule">
            20 &euro;
        </span>
        <span class="cellule">
            25 &euro;
        </span>
        <span class="cellule">
            30 &euro;
        </span>
    </div><!-- Fin ligne-->
    <div class="piedTable">
        Pied de table
    </div>
</div> <!-- Fin table -->
</div>
</body>
</html>

```



**Exercice : un bouton à partir d'un lien**



## 1.7 - Les transitions

### 1.7.1 - Présentation

Une transition permet des changements de valeurs de propriétés CSS en douceur sur une durée spécifiée.

<http://www.w3.org/TR/css3-transitions/>

<http://www.css3.info/preview/css3-transitions/>

Les propriétés CSS3 pour les transitions :

| Propriétés                 | Description   |
|----------------------------|---|
| transition-property        | La ou les propriétés CSS à transformer. Les valeurs possibles sont none, all, une propriété ou une liste de propriétés séparées par des virgules. |
| transition-duration        | La durée de la transition. Exprimée en seconde ou en milliseconde (s   ms).   |
| transition-timing-function | La fonction de transition à utiliser, le modèle d'interpolation (accélération, décélération, ...).  |
| transition-delay           | Le délai pour le démarrage de la transition. Exprimé en seconde ou milliseconde (positif ou négatif).   |
| transition                 | Le raccourci.   |

Les deux premières propriétés sont essentielles.

Une transition va être déclenchée quand la valeur de la propriété va changer.

---

### 1.7.2 - Exemples élémentaires

**Premier exemple : changement de couleur d'un lien (du noir au rouge) sur 10 secondes.**

Lien ... Lien ... Lien

#### transitions1.css

```
/* transitions1.css */

a {
  color: black;

  -webkit-transition-property: color;
  -webkit-transition-duration: 10s;

  -o-transition-property: color;
  -o-transition-duration: 10s;

  -moz-transition-property: color;
  -moz-transition-duration: 10s;

  -ms-transition-property: color;
  -ms-transition-duration: 10s;

  transition-property: color;
  transition-duration: 10s;
}
```

#### transitions1.html

```
<!DOCTYPE html>
<!-- transitions1.html -->
<html>
  <head>
    <title></title>
    <meta name="viewport" content="user-scalable=no, width=device-width,
initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
    <link rel="stylesheet" href="../css/transitions.css" />
  </head>

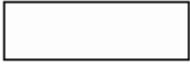
  <body>
    <div>
      <a href="#" id="lien">Lien</a>
    </div>

    <script type="text/javascript">
      document.getElementById("lien").style.color = "red";
    </script>
  </body>
</html>
```

## Deuxième exemple : transition sur 2 propriétés

La couleur et le positionnement d'une div.

Lien



Lien



Lien



```
<div id="boite">
</div>

<script type="text/javascript">
    // --- La boite
    document.getElementById("boite").style.backgroundColor = "black";
    document.getElementById("boite").style.left = "100px";
</script>
```

```
#boite {
    width: 100px;
    height: 30px;
    border: 1px solid black;
    background-color: white;
    position: relative;
    left: 0px;
    top: 0px;

    -webkit-transition-property: background-color, left;
    -webkit-transition-duration: 10s, 10s;
}
```

Note : puisque la durée est la même pour les 2 transitions il n'est pas nécessaire de la dupliquer.

---

### 1.7.3 - Le délai

La propriété `transition-delay` permet de préciser un délai avant le démarrage de la transition.

```
transition-delay: durée;
```

A titre d'exemple, reprenez la `<div>` identifiée par "boite".

```
-webkit-transition-delay: -1s, 10s;
```

La transition pour la couleur commence immédiatement et celle pour le déplacement 10 secondes après la modification de la propriété "left".

---

### 1.7.4 - La modification de l'interpolation

La propriété transition-timing-function permet de modifier le mode de calcul des valeurs intermédiaires durant la transition.

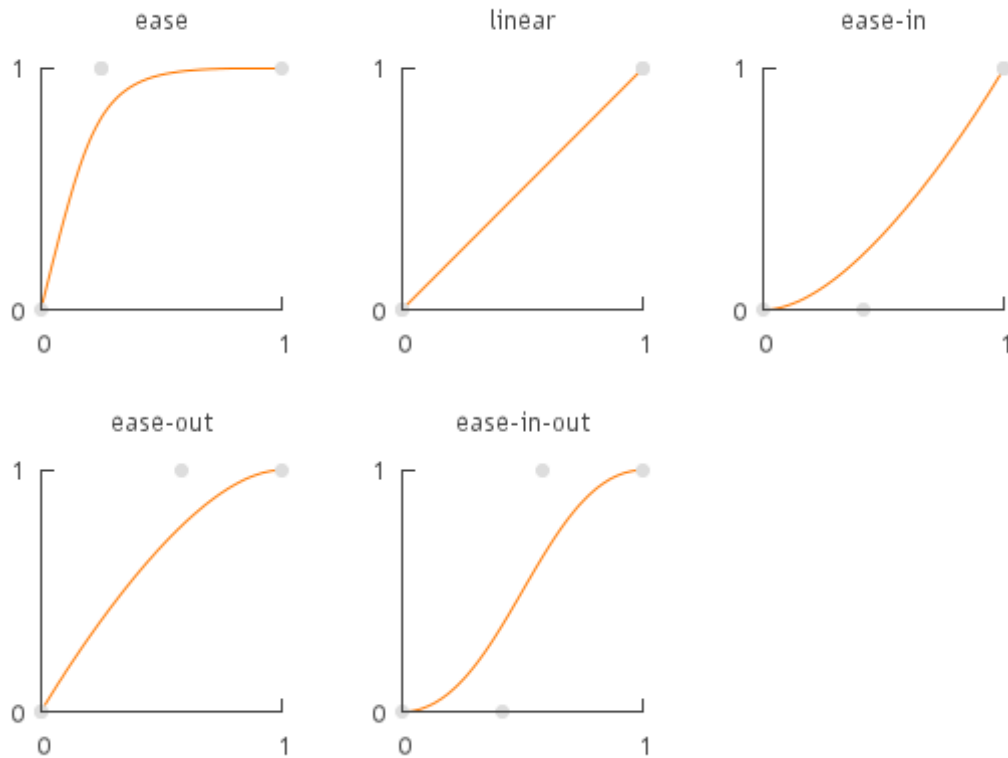
linear : vitesse constante.

ease : Rapide au début et lent à la fin.

ease-in : Lent au début et rapide à la fin.

ease-out : Rapide sur le début et décélère sur la fin.

ease-in-out : Le départ et la fin sont lents et rapide au milieu.



## Codes

```
<div id="interpolation">Interpolation</div>

<script type="text/javascript">
    document.getElementById("interpolation").style.left = "700px";
    document.getElementById("interpolation").style.backgroundColor = "red";
</script>
```

```
#interpolation {
    width: 100px;
    height: 30px;
    border: 1px solid black;
    background-color: white;
    position: relative;
    left: 0px;
    top: 0px;

    -webkit-transition-property: left, background-color;
    -webkit-transition-duration: 10s;
    /* linear, ease, ease-in, ease-out, ease-in-out, bezier(n, n, n, n) */
    -webkit-transition-timing-function: ease-in-out, bezier(1, 0.1, 1, .5);
    -webkit-transition-delay: 0s;
}
```

---

### 1.7.5 - L'utilisation du raccourci

#### Syntaxe raccourcie :

```
selecteur {  
  transition:  
    <transition-property>  
    <transition-duration>  
    <transition-timing-function>  
    <transition-delay> ;  
}
```

#### Exemple

Dans la page HTML

```
<span id="raccourci">Raccourci</span>  
  
<script type="text/javascript">  
  document.getElementById("raccourci").style.color = "blue";  
</script>
```

Dans le fichier CSS

```
#raccourci {  
  /* transition: property duration function delay */  
  -webkit-transition: color 3s linear 0s;  
}
```



---

### 1.7.6 - L'utilisation de code JavaScript sans pseudo-code CSS3

**Objectif :** effectuer une transition sans feuille de style ni code CSS3.

**Le code HTML :**

```
<br/>
<span id="pasDePseudoCode">Pas De Pseudo Code</span>
```

**Le code JavaScript (Pour webkit)**

```
<script type="text/javascript">
    document.getElementById("pasDePseudoCode").style.webkitTransitionProperty =
"color";
    document.getElementById("pasDePseudoCode").style.webkitTransitionDuration =
"10s";

    document.getElementById("pasDePseudoCode").style.color = "red";
</script>
```

## **1.8 - Transformations**

## 1.9 - Opacité

### Syntaxes

opacity: opacité allant de 0 à 1;

rgba(R, G, B, opacité );

Plus c'est opaque moins ça laisse passer la lumière. Donc plus c'est opaque plus c'est foncé.

### Exemple

Cent

Cinquante

Dix

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta name="viewport" content="user-scalable=no, width=device-width,
initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

    <style type="text/css">
      div{width: 100px; height: 30px; border: 3px solid red; background-
color: red; color: white; margin: 1em; text-align: center; font-weight: bold;}

      #cent{opacity: 1;}
      #cinquante{opacity: .5;}
      #dix{opacity: .1;}
    </style>

  </head>

  <body>
    <div id="cent">Cent</div>
    <div id="cinquante">Cinquante</div>
    <div id="dix">Dix</div>
  </body>
</html>
```

Le fond et le bord subissent l'effet de la propriété opacity.

Pour que ce ne soit que le fond.

Remplacez par :

```
#cent{background-color: rgba(255, 0, 0, 1);}
#cinquante{background-color: rgba(255, 0, 0, 0.5);}
#dix{background-color: rgba(255, 0, 0, 0.1);}
```

## 1.10 - Le Responsive Web Design ou les Media Queries

---

### 1.10.1 - Objectif

Adapter la présentation au terminal.

Les terminaux sont multiples : écrans, imprimantes, etc.

Et parmi les écrans, surtout avec les mobiles – smartphones et autres tablettes – les dimensions et résolutions sont nombreuses.

Les media queries du CSS3 permettent d'adapter la présentation en fonction du terminal.

Les critères d'adaptation sont multiples : device-width, device-height, width, height, orientation, resolution, monochrome, color, color-index, device-aspect-ratio, aspect-ratio, grid, scan,.

<http://www.w3.org/TR/css3-mediaqueries/>

cf aussi : <http://www.alsacreations.com/article/lire/930-css3-media-queries.html>

---

### 1.10.2 - Syntaxe

Combinaison d'un périphérique et d'une ou plusieurs expressions booléennes.

`@media [only] screen and [not] (max|min-propriété: unités)`

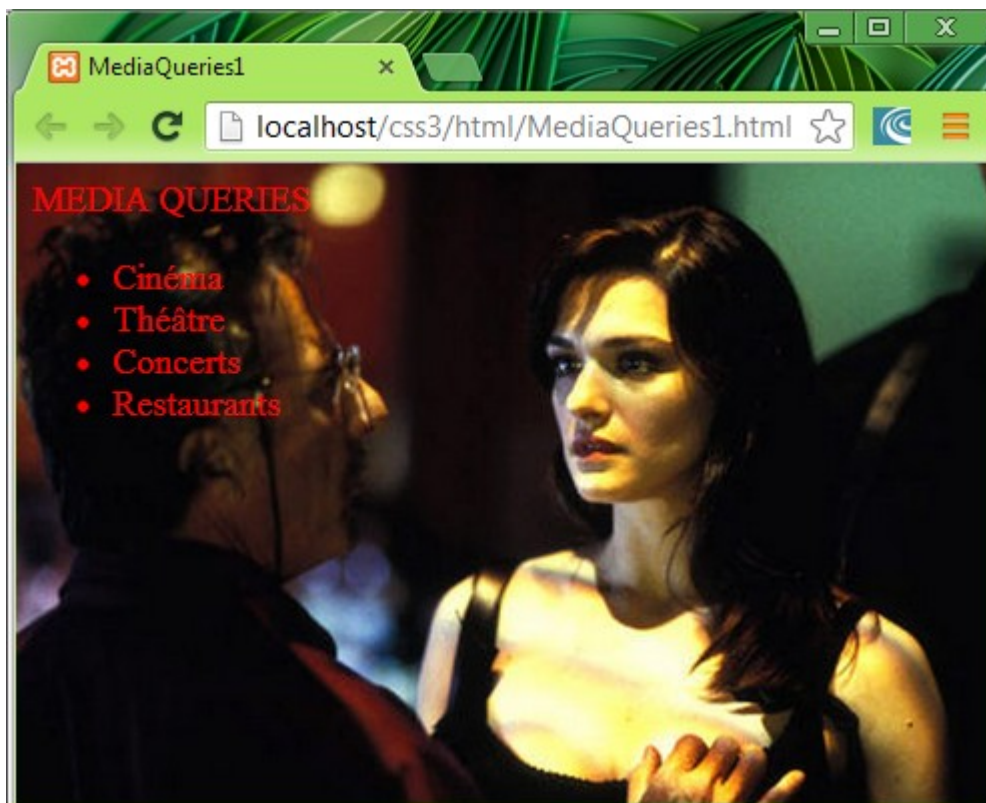
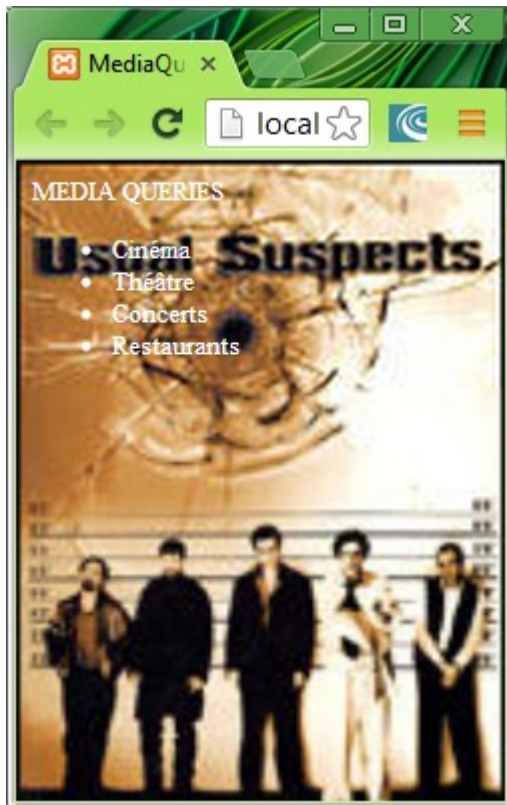
Exemple pour les écrans de moins de 480 pixels de large :

```
@media only screen and (max-width: 480px) {  
  nav ul li {  
  }  
}
```

Exemple pour les écrans des smartphones :

```
@media only screen and (orientation: portrait) {  
  nav ul li {  
  }  
}  
  
@media only screen and (orientation: landscape) {  
  nav ul li {  
  }  
}
```

### 1.10.3 - Exemple



Dans cet exemple la taille de la police et l'image change en fonction de la largeur de l'écran.

## MediaQueries1.html

```
<!--
MediaQueries1.html
-->
<!DOCTYPE html>
<html>
  <head>
    <title>MediaQueries1</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width" />
    <link href="../css/MediaQueries1.css" rel="stylesheet" />
  </head>

  <body>
    <div>
      MEDIA QUERIES
      <ul>
        <li>Cinéma</li>
        <li>Théâtre</li>
        <li>Concerts</li>
        <li>Restaurants</li>
      </ul>
    </div>
  </body>
</html>
```



## MediaQueries1.css

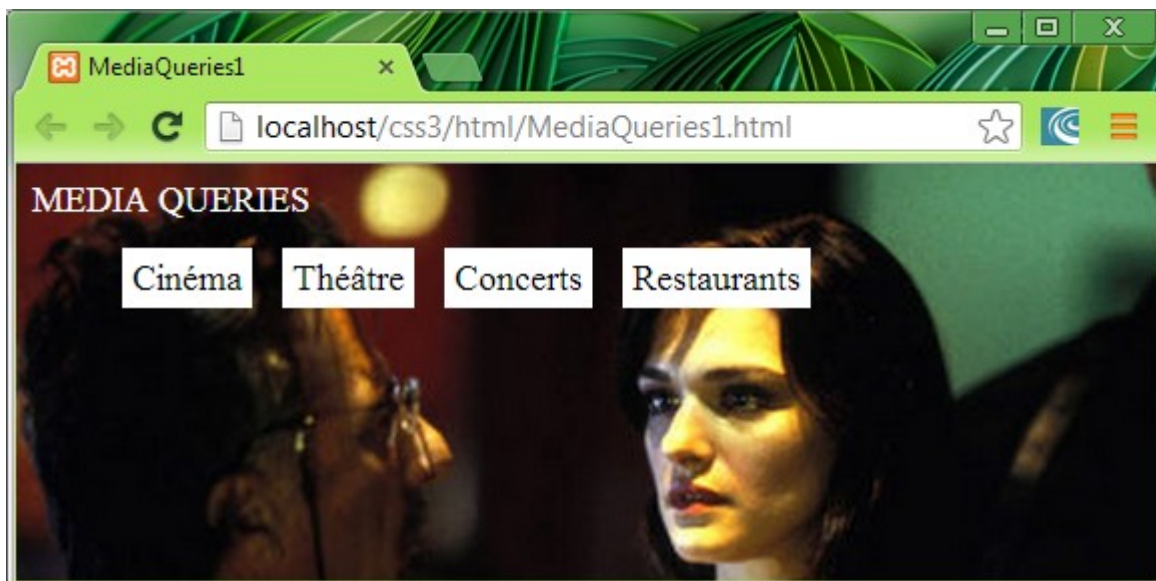
```
/*
    Document    : MediaQueries1.css
*/

/*
* Par défaut : grand écran
*/
body {
    color: red;
    background-image: url(http://localhost/images/cinema/paysage.jpg);
    background-repeat: no-repeat;
    background-size: 100%;
    font-size: large;
}

/*
* Si petit écran
*/
@media only screen and (max-width: 480px) {
    body {
        color: white;
        background-image: url(http://localhost/images/cinema/portrait.jpg);
        background-repeat: no-repeat;
        background-size: 100%;
        font-size: small;
    }
}
```

#### 1.10.4 - Exercice

Modifier le feuille de style pour obtenir le résultat suivant :



## 1.11 - Menu Hamburger



Du HTML  
Une image  
Du CSS  
Du JavaScript  
Du jQuery

## Du HTML :

```
<!DOCTYPE html>
<!-- Pates.html -->

<html>

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link rel="shortcut icon" type="image/x-icon"
href=" ../icones/favicon.ico" />
    <link rel="stylesheet" type="text/css" href=" ../css/main.css">
    <script src=" ../ ../lib/jquery/jquery.js"></script>
    <title>La pizza de Venise</title>
  </head>

  <body>

    <header class="entete">
      <script src="partials/HeaderView.js"></script>
    </header>

    <nav class="sommaire">
      <script src="partials/Nav.js"></script>
    </nav>

    <article id="articlePates" class="pCentre">

      <h1>Les pâtes</h1>

      <p>
        <label id="lblMessage"></label>
      <p>

      <dl id="listePates">
      </dl>

    </article>

    <script src=" ../js/main.js"></script>
    <script src=" ../js/Pates.js"></script>
  </body>
</html>
```

## Une image :

| hamburger\_blanc.png

## Du CSS :

```
.entete{
    position: fixed;
    top: 0px;
    width: 100%;
    height: 50px;
    padding: 5px;

    background-color: green;
    color: white;
    font-size: large;
    font-weight: bold;
}

.sommaire{
    background-color: green;
    margin-top: 60px !important;
}
```

## Du JavaScript :

### Nav.js

```
/*
 * Nav.js
 */
var sommaire = "";
sommaire += "<ul>";
sommaire += "<li><a href='Accueil.html'>Accueil</a></li>";
sommaire += "<li><a href='Pizzas.html'>Pizzas</a></li>";
sommaire += "<li><a href='Salades.html'>Salades</a></li>";
sommaire += "<li><a href='Pates.html'>P&acirc;tes</a></li>";
sommaire += "<li><a href='Desserts.html'>Desserts</a></li>";
sommaire += "<li><a href='Vins.html'>Vins</a></li>";
sommaire += "<li><a href='SodasEauxEtc.html'>Sodas, eaux, ...</a></li>";
sommaire += "<li><a href='Acces.html'>Acc&egrave;s</a></li>";
sommaire += "<li><a href='Contact.html'>Je contacte la pizzeria !</a></li>";
sommaire += "<li><a href='Aide.html'>Aide</a></li>";
sommaire += "<li><a href='APropos.html'>A propos</a></li>";
sommaire += "</ul>";

$("nav").html(sommaire);
```

### HeaderView.js

```
/*
 * HeaderView.js
 */
var entete = "";

entete += "<p>";
```

```

entete += "<a href='#'>";
entete += "<img src='../icones/hamburger_blanc.png' alt='menu' width='40' id='hamburger'/>";
entete += "</a>&nbsp;&nbsp;&nbsp;";

entete += "<a href='https://www.yelp.fr/biz/pizz%C3%A9ria-de-venise-paris-2' target='_blank'>";
entete += "<img src='../icones/yelp.jpg' alt='Yelp' title='Yelp' width='40'/>";
entete += "</a>";

entete += "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Pizza de Venise";

entete += "</p>";

$("header").html(entete);

```

## Main.js

```

/*
 * main.js
 */
var lbMenuAffiche = false;

$("#hamburger").click(afficherMasquerSommaire);

/**
 *
 * @returns {undefined}
 */
function afficherMasquerSommaire() {
    if (lbMenuAffiche) {
        masquerSommaire();
    }
    else {
        afficherSommaire();
    }
    lbMenuAffiche = !lbMenuAffiche;
} /// afficherMasquerSommaire

/**
 *
 * @returns {undefined}
 */
function afficherSommaire() {

    var hauteur = $(window).height();

    /**
     *
     */
    $(".sommaire").css("position", "absolute");
    $(".sommaire").css("left", "-300px");
    $(".sommaire").css("top", "0px");
    $(".sommaire").css("width", "200px");
    $(".sommaire").css("opacity", "0");
    $(".sommaire").css("height", hauteur + "px");
    $(".sommaire").css("display", "block");
    $(".sommaire").animate({left: 0, opacity: 1}, 1000);

} /// afficherSommaire

```

```
/**
 *
 * @returns {undefined}
 */
function masquerSommaire() {
    $(".sommaire").fadeOut(1000);
} /// masquerSommaire

/*
 *
 */
$(".sommaire").hide();
```