

# Cascading Style Sheets



- Événements
- Bordures
- Dégradés
- Transformations
- Animations
- Type d'écran

# Les événements

- Souris au-dessus de l'élément  
⇒ événement : **hover**, s'écrira :

```
nomBalise: hover{  
// style à appliquer pour cet état  
}
```

```
<div class="survol">div avec gestion de passage de souris</div>  
<div class="clique">div avec gestion de clique de souris</div>
```

```
div.survol: hover{  
    background-color: greenyellow;  
}  
  
div.clique: active{  
    background-color: red;  
}
```

- Élément actif (clic gauche sur l'élément)  
□ événement : **active**, s'écrira :

```
nomBalise: active{  
// style à appliquer pour cet état  
}
```

```
div avec gestion de passage de souris  
div avec gestion de clique de souris
```

# Bordures

Il est possible de faire des bordures arrondies, avec une ombre ou avec une image pour bordure.

Exemple d'une bordure arrondie :

```
div{  
  border: 3px ridge green;  
  border-radius: 10px;  
  background-color: yellow;  
  text-align: center;  
}
```

une div avec des bordures arrondies

# Bordures

Une bordure avec ombre s'écrira :

**box-shadow : h v b s c ;**

- h : décalage horizontal en pixel
- v : décalage vertical en pixel
- b : distance de flou en pixel
- s : taille de l'ombre en pixel
- c : couleur de l'ombre

```
div{  
  box-shadow: 10px 10px 20px 5px violet;  
  background-color: yellow;  
  text-align: center;  
  width: 300px;  
}
```

Il est possible d'ajouter le mot clef "inset" pour que l'ombre soit à l'intérieur de la boîte plutôt qu'à l'extérieur.



une div avec une bordure ombrée

# Bordures

Il est possible de mettre une image en tant que bordure d'une boîte. Ceci s'écrira :

**border-image : url(n) s w r**

- n : nom de l'image
- s : taille des "coins" de l'image
- w : largeur du contour
- r : répétition ou pas de l'image, hors "coins" (round ou stretch)



# Bordures

Image en tant que bordure d'une boîte avec répétition « round » :

```
div{  
  border: 15px solid transparent;  
  border-image: url(effetBordure.png) 35 35 round;  
  text-align: center;  
  width: 300px;  
}
```



une div avec une bordure image

Image en tant que bordure d'une boîte avec répétition « stretch »

```
div{  
  border: 15px solid transparent;  
  border-image: url(effetBordure.png) 35 35 stretch;  
  text-align: center;  
  width: 300px;  
}
```



une div avec une bordure image

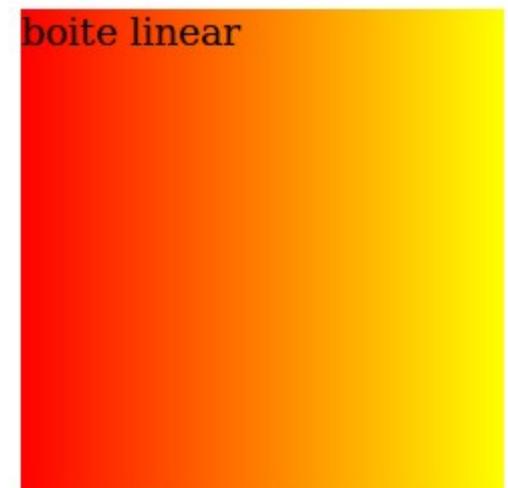
# Dégradés

Il est possible de créer un dégradé linéaire.

Ceci s'écrira : **background : linear-gradient(dir,cd,cf) ;**

- dir : direction du dégradé
- cd : couleur début
- cf : couleur fin

```
div.boiteLinear{  
  background: linear-gradient(to left,yellow,red)  
  width: 200px;  
  height: 200px;  
}
```

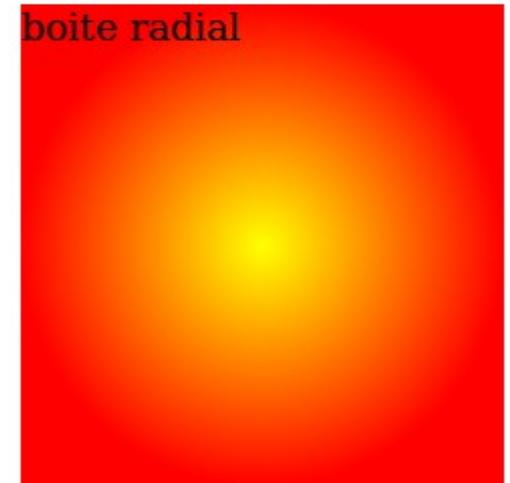


# Dégradés

Il est possible de créer un dégradé radial. Ceci s'écrira : **background : radial-gradient(r,cd,cf) ;**

- r : rayon du dégradé
- cd : couleur début
- cf : couleur fin

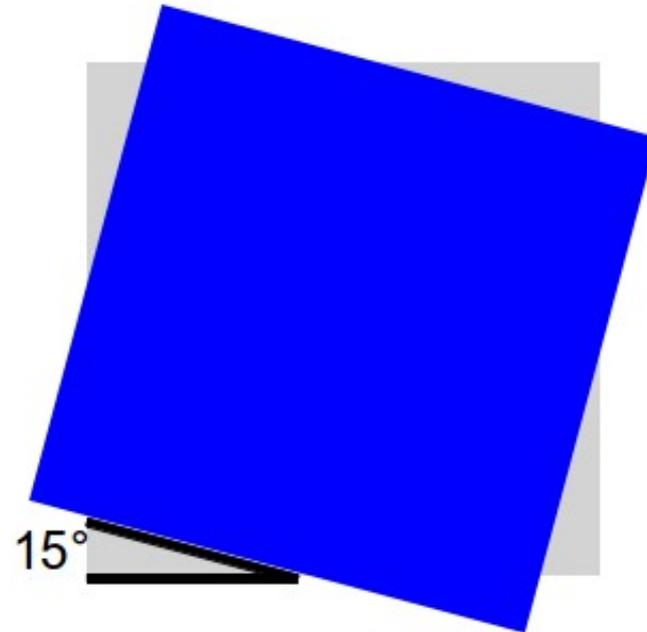
```
div.boiteRadiale{  
  background: radial-gradient(100px,yellow,red);  
  width: 200px;  
  height: 200px;  
}
```



# Transformations

Il est possible de réaliser une rotation d'un élément.  
Ceci s'écrira : **transform : rotate(angleEnDegre) ;**

```
div.pasrot{
  background: lightgray;
  width: 200px;
  height: 200px;
  position: absolute;
  left: 300px;
  top: 300px;
}
div.rot{
  background: blue;
  width: 200px;
  height: 200px;
  position: absolute;
  left: 300px;
  top: 300px;
  transform: rotate(15deg);
}
```

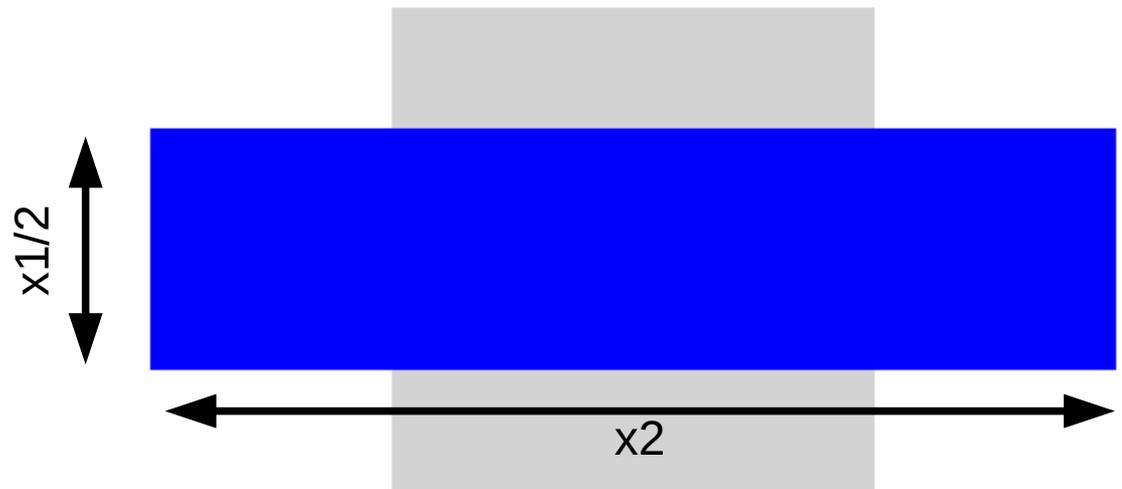


# Transformations

Il est possible de réaliser une mise à l'échelle d'un élément (agrandissement / réduction). Ceci s'écrira :

**transform : scale(ajustementLargeur,ajustementHauteur) ;**

```
div.pascal{
  background: lightgray;
  width: 200px;
  height: 200px;
  position: absolute;
  left: 300px;
  top: 300px;
}
div.scal{
  background: blue;
  width: 200px;
  height: 200px;
  position: absolute;
  left: 300px;
  top: 300px;
  transform: scale(2,0.5);
}
```

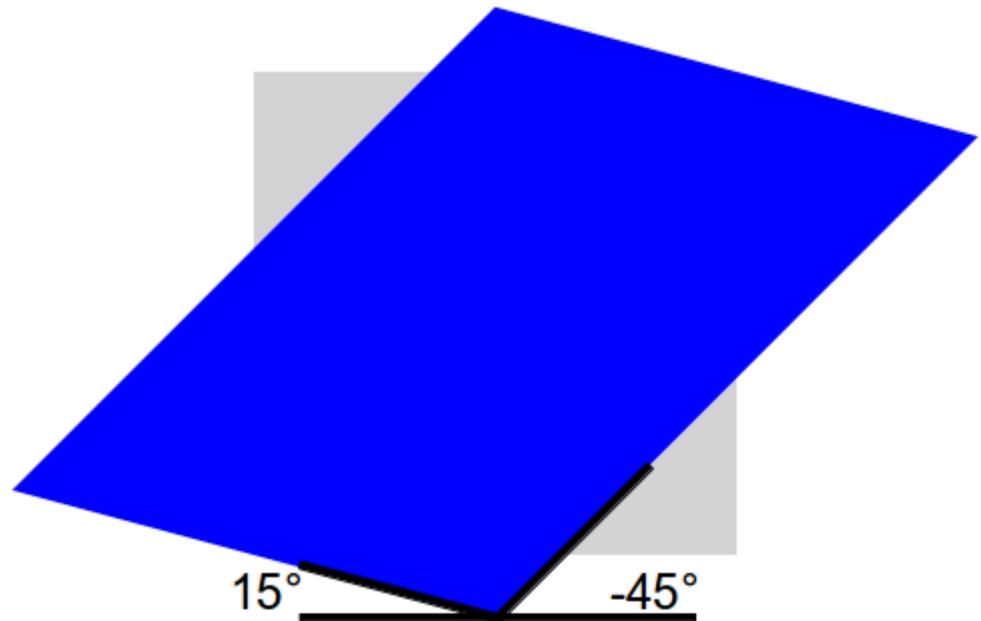


# Transformations

Il est possible de réaliser une déformation oblique d'un élément. Ceci s'écrira :

**transform : skew(angleVertical,angleHorizontal) ;**

```
div.passkew{
  background: lightgray;
  width: 200px;
  height: 200px;
  position: absolute;
  left: 300px;
  top: 300px;
}
div.avecskew{
  background: blue;
  width: 200px;
  height: 200px;
  position: absolute;
  left: 300px;
  top: 300px;
  transform: skew(-45deg,15deg);
}
```



# Animations

Il est possible d'animer des éléments.

Ceci s'écrira : **animation : ani dur type tempo nb re ;**

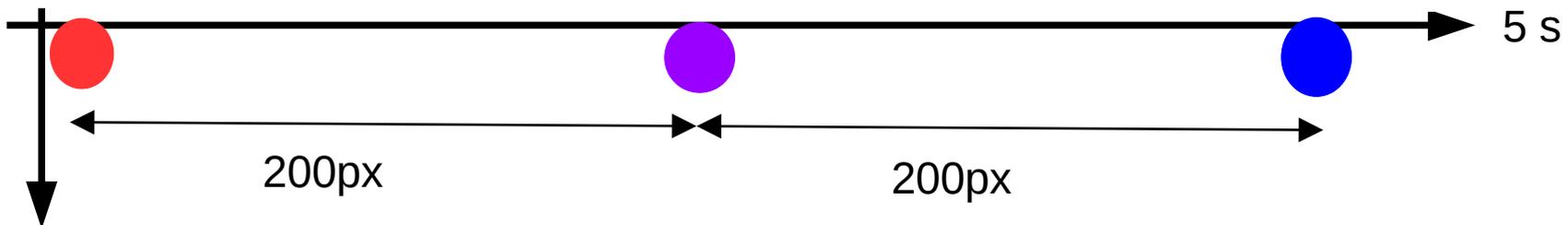
- ani : nom de la liste des points clefs pour les déplacements
- dur : durée de l'animation
- type : vitesse de l'animation (constante ou avec des variations)
- tempo : temporisation avant le début de l'animation
- nb : nombre de fois où l'animation devra être faite (peut être infini)
- re : refaire l'animation en sens inverse ou pas.

# Animations

## Exemple d'une animation :

```
div.bouleAnim
{
  width: 30px;
  height: 30px;
  border-radius: 15px;
  background: red;
  position: relative;
  animation: bougeHorizontal 5s linear 0s infinite alternate;
}
<div class="bouleAnim"></div>
```

```
@keyframes bougeHorizontal
{
  0% {background: red; left: 0px; top: 0px}
  50% {background: violet; left: 200px; top: 0px}
  100% {background: blue; left: 400px; top: 0px}
}
```



# Type d'écran

Il est possible d'appliquer un certain style en fonction des dimensions de l'écran.

Ceci s'écrira : `@media screen and|not|only (media feature)`  
`{`  
`code css ;`  
`}`

- media feature : conditions d'application du css  
(voir : [http://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](http://www.w3schools.com/cssref/css3_pr_mediaquery.asp))

# Type d'écran

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="tstmedia.css" />
  </head>
  <body>
    <div class="bigCarre">carre qui change de couleur et de dimension en
fonction de la taille de l'écran</div>
  </body>
</html>
```

```
@media screen and (min-width:640px)
{
  div.bigCarre{
    position: absolute;
    top: 10px;
    left: 10px;
    width: 620px;
    background-color: yellowgreen;
    height: 320px;
  }
}
```

```
@media screen and (min-width:800px)
{
  div.bigCarre{
    position: absolute;
    top: 10px;
    left: 10px;
    width: 780px;
    background-color: blue;
    height: 320px;
  }
}
```

# Type d'écran

- Attention, toujours définir de la plus petite résolution à la plus grande.
- L'adaptation des pages en fonctions des écrans est désignée sous l'appellation RWD (Responsive Web Design).
- Il existe des bibliothèques css permettant de simplifier la pratique du RWD comme Bootstrap
- Sites de référence :



[http://www.w3schools.com/css/css\\_rwd\\_intro.asp](http://www.w3schools.com/css/css_rwd_intro.asp)