



# Support de cours CSS3

Jérôme AMBROISE

Next Formation  
Semaine 27/02/2017 - 01/03/2017

# Sommaire (1/2)

## Une bordure en image

- A. Gestion des navigateurs
- B. La super-propriété
- C. Source
- D. Décalage
- E. Epaisseur de l'image
- F. Marge
- G. Répétition
- H. Conclusion
- I. Les différents navigateurs

## Les dégradés de couleurs

- A. Gradient linéaire
- B. Gradient radial

## Fond d'écrans

- A. Fond d'écrans multiples
- B. Taille de l'image de fond d'écran
- C. Origine de l'image de fond d'écran

## Les polices de caractères

- A. Les polices de caractères

## Les ombres

- A. Les ombres de texte
- B. Les ombres d'éléments

## Transformations

- A. Translation
- B. Rotation
- C. Mise à l'échelle
- D. Déformation
- E. Matrix

## Transition

- A. Présentation
- F. Courbe de vitesse
- G. Délai
- H. Super-propriété

## Animations

- A. Présentation
- B. Définition de l'animation
- C. Appel de l'animation
- D. Etats intermédiaires
- E. Délai, itérations
- F. Ordre, courbe de vitesse
- G. Super-propriété

# 1.

## Une bordure en image

Gestion des navigateurs

La super-propriété

Source

Décalage

Epaisseur de l'image

Marge

Répétition

Conclusion

# Une bordure en image

## Gestion des navigateurs

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

- Ces "propriétés propriétaires" doivent être préfixées d'un tiret et d'un code correspondant au moteur les exploitant :
  - -o- pour Opera
  - -moz- pour Gecko (Mozilla)
  - -webkit- pour Webkit (Chrome, Safari, Android...)
  - -ms- pour Microsoft (Internet Explorer)
  - -khtml- pour KHTML (Konqueror)

# Une bordure en image

## Gestion des navigateurs

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.

Quelques inconvénients font surface :

- Si l'on souhaite profiter en avance de ces styles, il faudra user de plusieurs propriétés pour une déclaration, avec chacun des préfixes.
- Cela peut être perçu comme une autre façon d'écrire des hacks spécifiques à chaque navigateur, c'est pourquoi certains puristes réclament l'utilisation d'un préfixe commun, tel que -beta-.
- Accessoirement, la syntaxe ou les propriétés peuvent changer, la spécification CSS étant toujours en cours de rédaction pour certains modules.

# Une bordure en image

## La super-propriété

L'objectif est de définir une image en tant que bordure pour un élément.

- Super-propriété : `border-image`
- Propriétés à fournir :
  - ◆ `border-image-source: none;`
  - ◆ `border-image-slice: 100%`
  - ◆ `border-image-width: 1`
  - ◆ `border-image-outset: 0s`
  - ◆ `border-image-repeat: stretch`

# Une bordure en image

## Source

Premièrement il faut définir la source de l'image :

→ Propriété : border-image-source

→ Valeurs : url(...)

### Exemple :

```
.border-rd {  
    border-image-source: url("../img/border-image-movie.png")  
}
```

# Une bordure en image

## Décalage

Maintenant que l'image est définie, il va falloir la découper. Ca tombe bien, slice veut dire « tranche » en anglais. C'est donc cet attribut qui va s'en charger

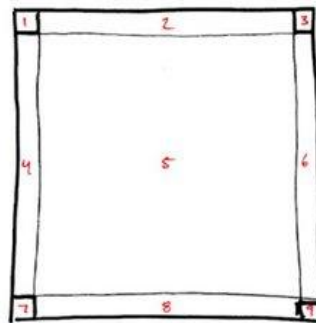
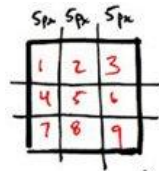
- Propriété : border-image-slice
- Valeurs : unité de mesures (1, 2 ou 4 valeurs)

### Exemple :

```
.border-rd {  
    border-image-slice: 23;  
}
```

Exemple de generateur :

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Background\\_and\\_Borders/Border-image\\_generator](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Background_and_Borders/Border-image_generator)





# Une bordure en image

## Epaisseur de l'image

Il est possible de définir la taille de l'image de bordure. Elle peut-être différente de l'épaisseur de la bordure.

- Propriété : `border-image-width`
- Valeurs : unité de mesure

### Exemple :

```
.border-rd {  
    border-image-width: 0.75;  
}
```

# Une bordure en image

## Marge

Avant-dernière propriété, elle permet d'introduire une marge intérieure supplémentaire séparant la bordure du contenu.

- Propriété : `border-image-outset`
- Valeurs : unité de mesure

### Exemple :

```
.border-rd {  
    border-image-outset: 35px;  
}
```

# Une bordure en image

## Répétition

Dernière propriété : la répétition de l'image.

- Propriété : border-image-repeat
- Valeurs : repeat, stretch, round, space

### Exemple :

```
.border-rd {  
    border-image-outset: repeat;  
}
```

# Une bordure en image

## Conclusion

Finalement, préférez utiliser la syntaxe de la super-propriété, et ajustez si besoin avec une propriété en particulier.

Note : “outset” n’est pas supporté par tous les navigateurs.

### Exemple :

```
.border-rd {  
    border-image: url("../img/border-image-movie.png") 106 36 stretch;  
    border-image-width: 0.75;  
}
```

# Une bordure en image

## Les différents navigateurs

Pour être supportable par la plupart des navigateurs :

### Exemple :

```
.border-rd {  
    -webkit-border-image: url("../img/border-image-movie.png") 106 36 stretch ;  
    -moz-border-image: url("../img/border-image-movie.png") 106 36 stretch ;  
    -o-border-image: url("../img/border-image-movie.png") 106 36 stretch ;  
    border-image: url("../img/border-image-movie.png") 106 36 stretch ;  
    border-image-width: 0.75;  
}
```

# 2.

## Les dégradés de couleurs

Gradient linéaire  
Gradient radial

# Les dégradés de couleurs

## Gradient linéaire

CSS3 permet de définir des dégradés de couleurs :

→ Propriété : background

→ Valeur : linear-gradient()

Dans sa forme minimale, la valeur “linear-gradient” accepte 2 arguments : couleur de départ, couleur d’arrivée.

Le dégradé s’établit de haut en bas.

Exemple :

```
#gradient-simple
{
    width:50%;
    height: 15rem;
    background: linear-gradient(#8319CA, #9B7B10);
}
```

# Les dégradés de couleurs

## Gradient linéaire

Il est possible de définir le point de départ du gradient (par défaut “haut”), selon les navigateurs, l’appel est différent.

### Exemple : Dégradé de gauche à droite

```
#gradient-left-right
{
  width:50%;
  height:15rem;
  background: blue; /* Si le navigateur n'accepte pas les dégradés */
  background: -webkit-linear-gradient(left, #000364 , #8F91B7); /* Safari */
  background: -o-linear-gradient(right, #000364, #8F91B7); /* Opera */
  background: -moz-linear-gradient(right, #000364, #8F91B7); /* Firefox */
  background: linear-gradient(to right, #000364 , #8F91B7); /* Syntaxe standardisée */
}
```



# Les dégradés de couleurs

## Gradient linéaire

Il est aussi possible de commencer à partir d'un coin de la balise.

### Exemple : Dégradé du coin haut-gauche au coin bas-droite

```
#gradient-from-top-left
{
  width:50%;
  height:15rem;
  background: red; /* Si le navigateur n'accepte pas les dégradés */
  background: -webkit-linear-gradient(left, #000364 , #8F91B7); /* Safari */
  background: -o-linear-gradient(right, #000364, #8F91B7); /* Opera */
  background: -moz-linear-gradient(right, #000364, #8F91B7); /* Firefox */
  background: linear-gradient(to right, #000364 , #8F91B7); /* Syntaxe standardisée */
}
```

# Les dégradés de couleurs

## Gradient linéaire

Pour définir un dégradé à plusieurs couleurs, il suffit de les lister.

### Exemple :

```
#gradient-colors
{
  width:50%;
  height:15rem;
  background: green; /* Si le navigateur n'accepte pas les dégradés */
  background: -webkit-linear-gradient(left, #000364 , #8F91B7); /* Safari */
  background: -o-linear-gradient(right, #000364, #8F91B7); /* Opera */
  background: -moz-linear-gradient(right, #000364, #8F91B7); /* Firefox */
  background: linear-gradient(to right, #000364 , #8F91B7); /* Syntaxe standardisée */
}
```

# Les dégradés de couleurs

## Gradient linéaire

Pour aller plus loin, on peut répéter plusieurs fois le dégradé dans une même balise.

Propriété : `repeating-linear-gradient()`

Valeur : (couleur de départ), (couleur d'arrivée + pourcentage)

### Exemple :

```
#gradient-multiple
{
    width:50%;
    height:15rem;
    background: green;
    background: -webkit-repeating-linear-gradient(#016508, #83C688 33.3%);
    background: -o-repeating-linear-gradient(#016508, #83C688 33.3%);
    background: -moz-repeating-linear-gradient(#016508, #83C688 33.3%);
    background: repeating-linear-gradient(#016508, #83C688 33.3%);
}
```

# Les dégradés de couleurs

## Gradient radial

Les dégradés circulaires sont définis de la même manière que les gradients linéaires mais avec une autre propriété :

Propriété : radial-gradient()

Valeur : (couleur de départ), (couleur d'arrivée + pourcentage)

### Exemple :

```
#gradient-multiple
{
    width:50%;
    height:15rem;
    background: green;
    background: -webkit-repeating-linear-gradient(#016508, #83C688 33.3%);
    background: -o-repeating-linear-gradient(#016508, #83C688 33.3%);
    background: -moz-repeating-linear-gradient(#016508, #83C688 33.3%);
    background: repeating-linear-gradient(#016508, #83C688 33.3%);
}
```

Remarque : Pour obtenir un cercle au lieu d'une ellipse, définir en premier paramètre "circle".

# Les dégradés de couleurs

## Gradient radial

Le paramètre taille définit la taille du dégradé. Il peut prendre quatre valeurs:

- closest-side
- farthest-side
- closest-corner
- Farthest-corner

### Exemple :

```
background: radial-gradient(closest-side at 60% 55%, red, yellow, black);
```

# Les dégradés de couleurs

## Gradient radial

Pour définir plusieurs radiants circulaires, c'est la même façon de procéder que pour les gradients linéaires mais avec une autre propriété :

### Exemple :

```
#gradient-radial-multiple
{
    width:50%;
    height:15rem;
    background: purple;
    background: -webkit-repeating-radial-gradient(circle, #321D56, #4400BA 25%, #D0BCF3 25%);
    background: -o-repeating-radial-gradient(circle, #321D56, #4400BA 25%, #D0BCF3 25%);
    background: -moz-repeating-radial-gradient(circle, #321D56, #4400BA 25%, #D0BCF3 25%);
    background: repeating-radial-gradient(circle, #321D56, #4400BA 25%, #D0BCF3 25%);
}
```

# 3.

## Fond d'écrans

Fond d'écrans multiples  
Taille de l'image de fond d'écran  
Origine de l'image de fond d'écran

# Fond d'écrans

## Fond d'écrans multiples

La super-propriété “background” permet de définir plusieurs images de fond d'écran :

### Exemple :

```
background : url("img/top.jpg") top left no-repeat,  
             url("img/bottom.jpg") bottom left no-repeat,  
             url("img/middle.jpg") left no-repeat;
```



# Fond d'écrans

## Taille de l'image de fond d'écran

Il est possible de définir la taille de l'image de fond d'écran pour gérer plus finement son affichage.

### Exemple :

```
#bg-css3
{
    background: url(../img/bg-arbre.jpg) right no-repeat;
    background-size: 200px;
}
```

# Fond d'écrans

## Taille de l'image de fond d'écran

La propriété “background-size” possède 2 mots clés qui peuvent être utile :

- “contain” : redimensionne l'image au maximum sans dépasser du contenant (il peut donc avoir des zones où l'image ne couvre pas le contenant.
- “cover” : redimensionne l'image pour couvrir complètement le contenant (des parties de l'images peuvent ne pas être visibles)

# Fond d'écrans

## Origine de l'image de fond d'écran

La propriété “background-origin” définit où peindre l'arrière-plan: à travers l'élément entier, à l'intérieur de la bordure ou à l'intérieur de la marge intérieure.

Propriété : background-origin

Value : border-box, padding-box, content-box

#bg-origin

{

background: url("../img/bg-example.jpg") no-repeat;

background-origin: border-box;

}

# Fond d'écrans

## Origine de l'image de fond d'écran

La propriété “background-origin” définit où peindre l'arrière-plan: à travers l'élément entier, à l'intérieur de la bordure ou à l'intérieur de la marge intérieure.

Propriété : background-origin

Value : border-box, padding-box, content-box

#bg-origin

{

background: url("../img/bg-example.jpg") no-repeat;

background-origin: border-box;

}

# 4.

## Les polices de caractères

# Formatage du texte

## Les polices de caractères

CSS3 permet d'utiliser votre propres polices de caractères en tant que ressources (stockée sur votre serveur).

Comme pour les médias, les navigateurs ne supportent pas tous les mêmes formats :

→ TTF/OTF, WOFF, WOFF2, SVG et EOT

# Formatage du texte

## Les polices de caractères

- L'utilisation des polices de caractères s'utilise avec la règle "@font-face" pour définir l'emplacement de votre police.
- Vous pouvez ensuite l'appeler comme une police de caractère classique

# Formatage du texte

## Les polices de caractères

### Exemple :

```
@font-face {  
  font-family: maFont;  
  src: url(../fonts/mapolice.woff);  
}
```

```
div {  
  font-family: maFont;  
}
```



# Formatage du texte

## Les polices de caractères

### Exemple :

```
@font-face {  
    font-family: 'yananeska';  
    src: url('../fonts/yananeska/yananeska.woff2') format('woff2'),  
        url('../fonts/yananeska/yananeska.woff') format('woff'),  
        url('../fonts/yananeska/yananeska.ttf') format('truetype');  
}  
  
#custom-font  
{  
    font-family: 'yananeska';  
    font-style: normal ;  
}
```

# Formatage du texte

## Les polices de caractères

Particularité d'internet explorer :

Le format “eot” est un peu particulier dans sa déclaration, pour permettre à internet explorer de parser :

```
url(yananeska.eot?) format('eot'),
```

# Formatage du texte

## Les polices de caractères

Certaines polices de caractères utilisent le “flat”, ils fournissent donc :

- Une police régulière “regular”
- Une police italique “italic”
- Une police avec mise en gras “bold”
- Éventuellement une police italique et gras “italic gras”

# Formatage du texte

## Les polices de caractères

Pour les polices dérivées, le nom “font-family” sera le même mais il faudra ajouter des propriétés pour préciser (italic, gras, ... ):

### Exemple :

```
@font-face {  
  font-family: yananaska;  
  src: url(yananaska_bold.woff);  
  font-weight: bold;  
}
```

Remarque : “font-style: italic;” pour l’italique

# 5.

## Les ombres

Les ombres de texte  
Les ombres d'éléments

# Les ombres

## Les ombres de textes

Il est possible de définir des ombres à du texte ou à des éléments de manière générale. La propriété pour définir une ombre à un texte est “text-shadow”.

Dans sa version avec 3 arguments, elle permet simplement de créer un double du texte et de le décaler selon l’axe X et Y avec une couleur différente.

### Exemple :

```
#shadow h1
{
  text-shadow: 2px 2px #7683e1;
}
```

# Les ombres

## Les ombres de textes

La version de “text-shadow” avec un 4ème argument introduit la notion de “flou”.

### Exemple :

```
#shadow h1
{
  text-shadow: 2px 2px 5px #7683e1;
}
```

# Les ombres

## Les ombres de textes

Il est possible de déclarer plusieurs ombres.

### Exemple :

```
#shadow h1
{
color: #0f1330;
text-shadow: 1px 1px 2px #efb1b1, 0 0 5px #cfcff0, 0 0 25px #b4a327;
}
```



# Les ombres

## Les ombres d'éléments

Pour définir des ombres à des blocs, la propriété est “box-shadow”, elle possède les mêmes valeurs que “text-shadow”

### Exemple :

```
#shadow aside
{
    width:20rem;
    height:10rem;
    background-color:navy;
    box-shadow: 10px 10px 5px grey;
}
```

# Les ombres

## Les ombres d'éléments

Comme avec “text-shadow” on peut définir plusieurs ombres.

### Exemple :

```
#shadow aside
{
    width: 20rem;
    height: 10rem;
    background-color: #dbdbdb;
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.44), 0 6px 20px 0
    rgba(0, 0, 0, 0.38);
    border: 1px solid darkgrey;
}
```

# 6.

## Transformations

Translation  
Rotation  
Mise à l'échelle  
Déformation  
Matrix

# Transformations

## Présentation

Les transformations CSS3 vous permettent de translater, de faire pivoter, de mettre à l'échelle et d'incliner des éléments.

Une transformation est un effet qui permet à un élément de changer de forme, de taille et de position.

# Transformations

## Translation

Une translation permet de “décaler” un élément selon les axes (x, y).

Propriété : `translate()`

Valeurs : (unité de mesure axe X) (unité de mesure axe Y)

### Exemple :

```
#translation
{
    width:10rem;
    height:10rem;
    background: olive;
    -ms-transform: translate(5rem, 7.5rem); /* IE 9 */
    -webkit-transform: translate(5rem, 7.5rem); /* Safari */
    transform: translate(5rem, 7.5rem);
}
```

# Transformations

## Rotation

La rotation fait tourner un élément dans le sens des aiguilles d'une montre selon un degré donné.

Propriété : rotate()

Valeurs : (degrés positifs ou négatifs)

### Exemple :

```
#rotation
{
    width:10rem;
    height:10rem;
    background: mediumpurple;
    -ms-transform: rotate(-175deg); /* IE 9 */
    -webkit-transform: rotate(-175deg); /* Safari */
    transform: rotate(-175deg);
}
```

# Transformations

## Mise à l'échelle

La mise à l'échelle augmente ou diminue la taille d'un élément (selon les paramètres donnés pour la largeur et la hauteur).

Propriété : `scale()`

Valeurs : (coefficient multiplicateur X) (coefficient multiplicateur Y)

### Exemple :

```
#echelle
{
    width:10rem;
    height:10rem;
    background: goldenrod;
    -ms-transform: scale(2.5, 0.5); /* IE 9 */
    -webkit-transform: scale(2.5, 0.5); /* Safari */
    transform: scale(2.5, 0.5);
    Margin:auto; /* Pour voir l'élément en entier */
}
```

# Transformations

## Déformation

Il est aussi possible de faire une déformation de l'élément selon les axes X et Y.

Propriété : skew()

Valeurs : (degrés axe X ) (degrés axe Y )

### Exemple :

```
#deformation-axes
{
    width:10rem;
    height:10rem;
    background: lightskyblue;
    -ms-transform: skew(15deg, 25deg); /* IE 9 */
    -webkit-transform: skew(15deg, 25deg); /* Safari */
    transform: skew(15deg, 25deg);
    margin:auto;
}
```



# Transformations

## Matrix

La méthode `matrix()` prend six paramètres, contenant des fonctions mathématiques, ce qui vous permet de faire pivoter, de mettre à l'échelle, de déplacer et d'incliner des éléments.

Propriété : `skew()`

Valeurs : `scaleX()`, `skewY()`, `skewX()`, `scaleY()`, `translateX()`, `translateY()`

### Exemple :

```
#matrix
{
    width:10rem;
    height:10rem;
    background: navy;
    -ms-transform: matrix(0.7, 0.5, 0, 0.7, 0, 0); /* IE 9 */
    -webkit-transform: matrix(1.5, 0.5, 0, 0.7, 0, 0); /* Safari */
    transform: matrix(0.7, 0.5, 0, 0.7, 0, 0);
    margin:auto;
}
```

# Transformations

## Transformation 3D

Il est possible de faire des rotations selon les 3 axes.

La méthode rotateX() fait tourner un élément autour de son axe X à un degré donné.

### Exemple :

```
#rotate-x
{
    -webkit-transform: rotateX(180deg); /* Safari */
    transform: rotateX(180deg); /* Standard syntax */
}
```

# Transformations

## Transformation 3D

La méthode rotateY() fait tourner un élément autour de son axe X à un degré donné.

### Exemple :

```
#rotate-y
{
    -webkit-transform: rotateY(180deg); /* Safari */
    transform: rotateY(180deg); /* Standard syntax */
}
```

# Transformations

## Transformation 3D

La méthode rotateZ() fait tourner un élément autour de son axe X à un degré donné.

### Exemple :

```
#rotate-z
{
    -webkit-transform: rotateZ(180deg); /* Safari */
    transform: rotateZ(180deg); /* Standard syntax */
}
```

# 7.

## Transition

Présentation  
Courbe de vitesse  
Délai  
Super-propriété

# Transition

## Présentation

Une transition permet de définir une durée entre 2 changements de propriétés.

Il faut définir :

- La propriété auquel la transition s'applique (multiple possible)
- La durée de la transition

# Transition

## Présentation

### Exemple :

```
#transition-bgcolor
{
    width:10rem;
    height:10rem;
    background: tan;
    -webkit-transition: background 2s; /* Safari */
    transition: background 2s;
}
#transition-bgcolor:hover
{
    background: teal;
}
```

# Transition

## Courbe de vitesse

La propriété “transition-timing-function” indique la courbe de vitesse de l'effet de transition :

- ease - spécifie un effet de transition avec un démarrage lent, puis rapide, puis finissent lentement (ce qui est par défaut)
- linear - spécifie un effet de transition avec la même vitesse du début à la fin
- ease-in - spécifie un effet de transition avec un démarrage lent
- ease-out - spécifie un effet de transition avec un final lent
- ease-in-out - spécifie un effet de transition avec un démarrage lent et rapide à la fin
- cubic-bezier(n,n,n,n) - vous permet de définir vos propres valeurs dans une fonction cubique-Bézier



# Transition

## Délai

La propriété “transition-delay” spécifie un délai (en secondes) pour l'effet de transition.

### Exemple :

```
#transition-easeout-delay
{
  width:10rem;
  background: rgba(111, 25, 25, 1);
  -webkit-transition: background 2s; /* Safari */
  transition: background 2s;
  transition-timing-function: ease-out;
  transition-delay: 0.5s;
}
#transition-easeout-delay:hover
{
  background: rgba(111, 25, 25, 0.3);
}
```

# Transition

## Super-propriété

La super-propriété “transition” permet de raccourcir l’écriture d’une transition.

### Exemple :

```
#transition-short
{
  width:10rem;
  background: rgba(111, 25, 25, 1);
  -webkit-transition: background 2s ease-out 0.5s; /* Safari */
  transition: background 2s ease-out 0.5s;
}
#transition-short:hover
{
  background: rgba(111, 25, 25, 0.3);
}
```

# 8.

## Animations

Présentation  
Définition de l'animation  
Appel de l'animation  
Etats intermédiaires  
Délai, itérations  
Ordre, courbe de vitesse  
Super-propriété

# Animations

## Présentation

Les animations sont similaires aux transitions dans la manière de changer de propriétés.

Seulement l'animation n'a pas besoin d'être déclenchée par une action utilisateur.

# Animations

## Définition de l'animation

Les animations introduisent la directive “@keyframes” qui permet de définir un nom à l'animation.

### Exemple :

```
@keyframes animationTest
{
    ...
}
```

# Animations

## Définition de l'animation

Lors de la définition de votre animation, 2 propriétés (au minimum) doivent être définies :

→ from

→ to

### Exemple :

```
@keyframes animationTest
{
    from{color:gainsboro ;}
    to{color:indigo;}
}
```

# Animations

## Appel de l'animation

Pour appeler l'animation, il y a deux élément à fournir :

- Le nom de l'animation
- La durée de l'animation

### Exemple :

```
#animation-simple p
{
  width: 10rem;
  height: 10rem;
  color: gainsboro ;
  animation-name: animationTest;
  animation-duration: 6s;
}
```

# Animations

## Etats intermédiaires

Dans la définition de l'animation “from” signifie 0% et “to” 100%. Il est possible de définir des état intermédiaires entre les deux avec des pourcentages :

### Exemple :

```
@keyframes animationMultiple
{
  0%{background-color:darkorchid;}
  33%{background-color:indianred;}
  66%{background-color:lightpink;}
  100%{background-color:MediumSpringGreen ;}
}
```

```
#animation-composee
{
  width: 10rem;
  height: 10rem;
  background-color: darkorchid ;
  animation-name: animationMultiple;
  animation-duration: 6s;
}
```



# Animations

## Délai, itérations

- Comme pour les transitions, il est possible de définir un délai :

Propriété : animation-delay

Valeur : (nombre en secondes)

- Il est aussi possible de définir le nom d'itérations que l'animation doit effectuer.

Propriété : animation-iteration-count

Valeurs : (nombre) ou infinite

# Animations

## Délai, itérations

### Exemple :

```
@keyframes animationRedimension
{
  0%{width:10rem;}
  50%{width:30rem;}
  100%{width:20rem;}
}
```

```
#animation-redimension
{
  width: 10rem;
  height: 10rem;
  background-color: darkorchid ;
  animation-name: animationRedimension;
  animation-duration: 5s;
  animation-delay: 3s;
  animation-iteration-count: 5;
}
```

# Animations

## Ordre, courbe de vitesse

- Une autre propriété permet de changer l'ordre d'exécution de l'animation (exprimé en pourcentages dans l'animation).

Propriété: animation-direction

Valeurs: reverse, alternate

- Enfin, il est possible de définir, comme pour les transitions, la courbe de vitesse de l'animation

Propriété: animation-timing-function

Valeurs: linear, ease, ease-in, ease-out, ease-in-out

# Animations

## Ordre, courbe de vitesse

### Exemple :

```
@keyframes animationComple
```

```
{
```

```
  0%{left:0;top:0;}
```

```
  12%{left:10rem;top:0;}
```

```
  25%{left:20rem;top:0;}
```

```
  37%{left:20rem;top:-10rem;}
```

```
  50%{left:20rem;top:-20rem;}
```

```
  62%{left:10rem;top:-20rem;}
```

```
  75%{left:0rem;top:-20rem;}
```

```
  87%{left:0rem;top:-10rem;}
```

```
  100%{left:0rem;top:0rem;}
```

```
}
```

```
#animation-redimension
```

```
{
```

```
  width: 10rem;
```

```
  height: 10rem;
```

```
  position: relative;
```

```
  -webkit-border-radius: 50%;
```

```
  -moz-border-radius: 50%;
```

```
  border-radius: 50%;
```

```
  background-color: OrangeRed;
```

```
  animation-name: animationComple;
```

```
  animation-duration: 5s;
```

```
  animation-delay: 3s;
```

```
  animation-iteration-count: 5;
```

```
  animation-direction: alternate;
```

```
  animation-timing-function: ease-out;
```

```
}
```

# Animations

## Super-propriété

- La super-propriété “animation” permet de réduire l’écriture pour appeler une animation avec ses 6 propriétés :

### Exemple :

```
#animation-redimension
```

```
{
```

```
    animation: animationComplet 5s 3s 5 alternate ease-out;
```

```
}
```

Remarque : L’ordre peut être changé mais quelques contraintes :

- Le nom de l’animation apparaît en premier
- La durée d’animation apparaît avant le délai

Merci de votre attention !

=)