



# PROGRAMMATION WEB

## HTML5/CSS 3

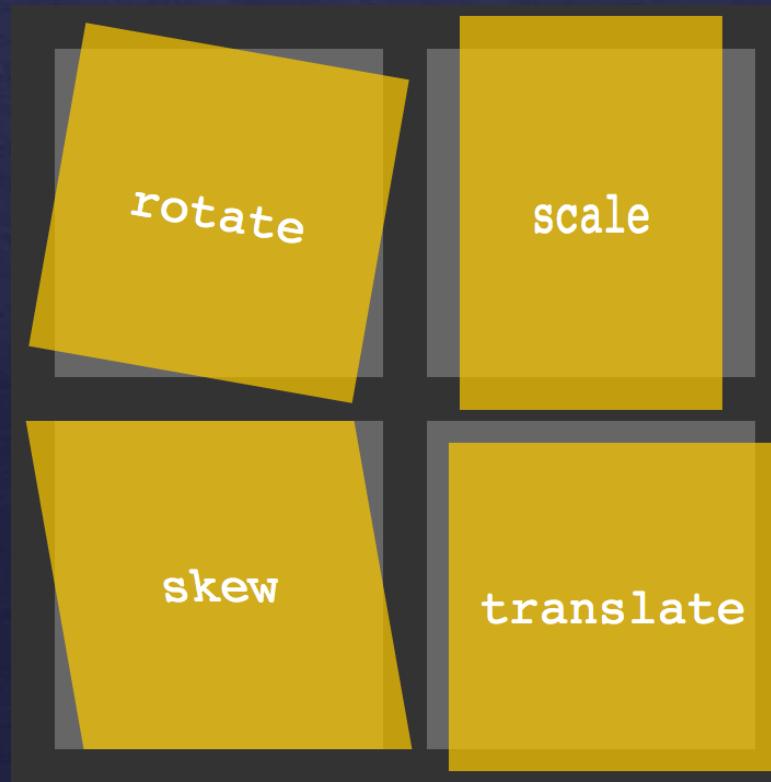
# Plan

- ❖ Les transformations
- ❖ Les transitions
- ❖ Les animations
- ❖ Exercice

# Les transformations

# Les transformations

- ◊ CSS3 permet de manipuler un élément HTML en 2 dimensions (sur les axes X et Y (horizontal et vertical)) grâce à des fonctions diverses de transformation graphique.



# Les transformations

- ❖ Deux propriétés majeures sont utilisées pour définir les transformations CSS : **transform** et **transform-origin**
- ❖ **transform-origin:** Spécifie la position de l'origine. Par défaut, celle-ci est dans le coin supérieur gauche de l'élément et peut être déplacée. Elle est utilisée par de nombreuses transformations, qui nécessitent un point spécifique pour paramètre.
- ❖ **Transform :** Spécifie la transformation à appliquer à l'élément. C'est une liste de transformations séparée par des espaces, qui sont appliquées les unes après les autres, comme requis par l'opération de composition. Les transformations qui sont composées entre elles sont appliquées dans l'ordre, de droite à gauche.

# Les transformations

## ❖ Fonction de zoom: **scale()**

```
#photo  
{  
    -webkit-transform: scale(0.5); /* Safari, Chrome */  
    -ms-transform: scale(0.5); /* IE */  
    -moz-transform: scale(0.5); /* Firefox */  
    -o-transform: scale(0.5); /* Opera */  
    transform: scale(0.5);  
}
```

# Les transformations

- ❖ Les fonctions **scaleX()** et **scaleY()** existent aussi et permettent d'appliquer la modification uniquement sur un des axes

```
#photo
{
  -moz-transform: scaleX(-1);
  -webkit-transform: scaleX(-1);
  -o-transform: scaleX(-1);
  transform: scaleX(-1);
}
```

# Les transformations

- ❖ La fonction de rotation: **rotate()**

```
#photo  
{  
-moz-transform: rotate(45deg);  
-webkit-transform: rotate(45deg);  
-o-transform: rotate(45deg);  
transform: rotate(45deg);  
}
```

# Les transformations

- ❖ Pour modifier l'origine de la transformation

```
#photo
{
    -webkit-transform-origin: 0 0; /* Safari, Chrome */
    -ms-transform-origin: 0 0; /* IE */
    -moz-transform-origin: 0 0; /* Firefox */
    -o-transform-origin: 0 0; /* Opera */
    transform-origin: 0 0; /* La transformation partira du coin en haut à gauche */
    transform: rotate(10deg);
}
```

# Les transformations

- ❖ La fonction de translation: **translate()**

```
#photo
{
    -moz-transform: translate(100px, 100px);
    -webkit-transform: translate(100px, 100px);
    -o-transform: translate(100px, 100px);
    transform: translate(100px, 100px);
}
```

# Les transformations

- ❖ La transformation oblique: "étire" un des côtés du bloc pour en faire un losange. On utilise pour cela la fonction **skew(ax)** ou **skew(ax, ay)**

#photo

```
{  
-moz-transform: skew(-25deg, 10deg);  
-webkit-transform: skew(-25deg, 10deg);  
-o-transform: skew(-25deg, 10deg);  
transform: skew(10deg);  
}
```

# Les transformations

- ❖ Combinez les transformations

```
{  
-moz-transform: scaleX(-1) rotate(45deg) translate(5px,10px);  
-webkit-transform: scaleX(-1) rotate(45deg) translate(5px,10px);  
-o-transform: scaleX(-1) rotate(45deg) translate(5px,10px);  
transform: scaleX(-1) rotate(45deg) translate(5px,10px);  
}
```

# Les transitions

# Les transitions

- ❖ Le principe de base d'une transition CSS 3 est de permettre un passage en douceur de l'ancienne vers la nouvelle valeur d'une propriété CSS lorsqu'un événement est déclenché
- ❖ Les transitions permettent de choisir :
  - ❖ les propriétés à animer en les listant explicitement
  - ❖ le début de l'animation
  - ❖ la durée de l'animation
  - ❖ la façon dont la transition s'exécutera

# Les transitions

- ❖ Les transitions CSS sont généralement contrôlées grâce à la propriété raccourcie **transition** , Syntaxe:

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

- ❖ **transition-property** : Seules les propriétés listées ici seront sujettes aux transitions. Les modifications appliquées aux autres propriétés seront instantanées.
- ❖ **transition-duration**: définit la durée de la transition. On peut définir une durée pour toutes les transitions ou une durée pour chacune des propriétés.
- ❖ **transition-timing-function**: définit une fonction qui décrit la façon dont les valeurs intermédiaires sont calculées. On utilise pour cela des fonctions de temporisation.
- ❖ **transition-delay**: indique le temps à attendre entre le moment où la propriété est modifiée et le début de la transition.

# Les transitions

- ❖ Propriété à animer : **transition-property**
- ❖ **all** (valeur par défaut) : toutes les propriétés possibles seront animées ;
- ❖ **propriété** : le nom d'une propriété pouvant être animée ;
- ❖ **none** : aucune propriété ne sera animée.

# Les transitions

- ❖ Durée de l'animation : **transition-duration**
- ❖ Les deux unités de temps acceptées sont:
  - ❖ **s** : la seconde ;
  - ❖ **ms** : la milliseconde.
- ❖ Si plusieurs propriétés ont été définies à l'aide de transition-property, il est possible de préciser leurs valeurs en les séparant d'une virgule.

# Les transitions

- ❖ Accélération : **transition-timing-function** :
  - ❖ **linear** (valeur par défaut) : vitesse constante sur toute la durée de l'animation ;
  - ❖ **ease** : rapide sur le début et ralenti sur la fin ;
  - ❖ **ease-in** : lent sur le début et accélère de plus en plus vers la fin ;
  - ❖ **ease-out** : rapide sur le début et décélère sur la fin ;
  - ❖ **ease-in-out** : le départ et la fin sont lents.

# Les transitions

## ❖ Exemple: menu de navigation avec transition

- En HTML:

```
<nav>  
  <a href="#">Accueil</a>  
  <a href="#">À propos</a>  
  <a href="#">Contact</a>  
  <a href="#">Liens</a>  
</nav>
```

# Les transitions

- En CSS:

```
a {  
    color: #fff;  
    background-color: #333;  
    transition: all 1s ease-out;  
}
```

```
a:hover,a:focus {  
    color: #333;  
    background-color: #fff;  
}
```

# Les animations

# Les animations

- ◊ Les animations CSS permettent de créer des transitions entre deux états de mise en forme.
- ◊ Une animation est décrite par deux choses : des propriétés propres à l'animation d'une part et un ensemble d'étapes (keyframes) qui indiquent l'état initial, final et éventuellement des états intermédiaires d'autre part.

# Les animations

- ❖ La déclaration:

```
@keyframes nomAnim{  
from {  
    color:blue;  
}  
35% {  
    color:green;  
}  
to {  
    color: red;  
}  
}
```

# Les animations

- ❖ Les propriétés:

- ❖ **animation-name** : le nom de l'animation à utiliser.
- ❖ **animation-duration** : le temps total de l'animation.
- ❖ **animation-timing-function** : la méthode d'interpolation (accélération, décélération).
  - **linear**,
  - **ease**,
  - **ease-in**,
  - **ease-out**,
  - **ease-in-out**,
  - **cubic-bezier( p1, p2, p3, p4)**

# Les animations

- ❖ Les propriétés:
  - ❖ **animation-iteration-count** : le nombre de répétition de l'animation. La valeur infinite permet de jouer une animation en continu.
  - ❖ **animation-direction** : permet de jouer une animation en sens inverse (en fonction du cycle).
    - Normal
    - Reverse
    - Alternate
    - Alternate-reverse
  - **animation-play-state** : mettre en pause l'animation.
    - En cours d'étude de suppression par le W3C.
    - Paused
    - running
  - ❖ **animation-delay** : le temps avant que l'animation ne démarre.

# Les animations

- ❖ La propriété animation:

```
div{  
    animation: nomAnim 5s linear 2s;  
}
```

- ❖ Exemple:

```
@-webkit-keyframes slideIn {  
    from { top:0px; }  
    to { top:10PX; }  
}
```

```
p {  
    animation: 3s infinite alternate  
    slideIn;  
}
```

# Les animations

- ❖ La compatibilité:

```
@-webkit-keyframes nomAnim{ ... }
```

```
@-moz-keyframes nomAnim{ ... }
```

```
@-ms-keyframes nomAnim{ ... }
```

```
@-o-keyframes nomAnim{ ... }
```

```
@keyframes nomAnim{ ... }
```

Exemple:

```
div{  
  -webkit-animation: nomAnim 5s linear 2s;  
  -moz-animation: nomAnim 5s linear 2s;  
  -ms-animation: nomAnim 5s linear 2s;  
  -o-animation: nomAnim 5s linear 2s;  
  animation: nomAnim 5s linear 2s;  
}
```

# Exercice

## Création d 'un slideshow automatique