

# Transformations, transitions et animations avec CSS

- Animations sans Javascript ni flash
- Rapide, facile à mettre en place
- Supporté par les navigateurs récents

# <STYLES>

## transform, transition et animation

- transform :  
Rotation, translation, effet 3D, etc...
- transition :  
Contrôle le "timing" de modifications de style ou de transformations
- Animation :  
Contrôle le "timing" d'une suite de transformations ou de modifications de style

# Préfixes à utiliser pour la compatibilité

- Utilisez les préfixes

- webkit- pour chrome et safari

- moz- pour mozilla firefox

- ms- pour internet explorer

- o- pour opera

- Exemple :

- ... {-moz-transform : rotate(20deg) ;.... }

## Pour la compatibilité avec les navigateurs

- Ecrire 5 fois avec les mêmes paramètres :

```
#selecteur {-moz-transform : rotate(20deg) ;
```

```
-webkit-transform : rotate(20deg) ;
```

```
-ms-transform : rotate(20deg) ;
```

```
-o-transform : rotate(20deg) ;
```

```
transform : rotate(20deg) ;
```

```
.....
```

```
}
```

- Le navigateur utilise la bonne ligne...

## Transformations 2D et 3D

- Syntaxe : ..{*-prefixe-transformation* : la transformation ;...}

- Les transformations 2D

*rotate(angle)* rotation

*scale(x,y)*, *scaleX(x)* et *scaleY(y)* mise à l'échelle

*skewX(angle)* et *skewY(angle)* pour incliner un élément

*translate(x,y)*, *translateX(x)* et *translateY(y)* translation

*matrix(a, c, b, d, x, y)* matrice de transformation

## Transformations 3D (suite)

- rotateX(angle), rotateY(angle) et rotateZ(angle)
- rotate3d(x,y,z,angle) pour une rotation autour du vecteur 3D x,y,z.
- translateZ(z) pour des translations sur l'axe Z (profondeur).
- translate3d(x,y,z) pour des translations en 3D.
- scaleZ(z) pour des changements d'échelle sur l'axe Z
- scale3d(x,y,z) pour des changements d'échelle en 3D.
- matrix3d() spécifie une matrice de transformation 4x4.
- perspective distance entre l'observateur et la scène
- perspective-origin point de fuite des éléments .
- backface-visibility: visibilité des faces arrières.

## Transitions avec CSS

- Pour contrôler les modification de style ou les transformations
- Etat de départ vers un état d'arrivée avec un timing
- Exemple :

```
#identifiant{ width : 300px ; height :200px ;color : red ;...}
```

```
#identifiant:hover{width:400px ; color:blue ;  
                    -moz-transition : width 5s linear 0s ;}
```

```
#identifiant:active{width:600px ; color:yellow ;  
                    -moz-transform:rotate2D(-20deg) ;  
                    -moz-transition : all 5s linear 0s ;}
```

```
<div id=identifiant >...contenu à animer.....</div>
```

- On peut utiliser d'autres sélecteurs : *.class, img, etc...*

## Paramétrer une transition

- *transition* : regroupe plusieurs paramètres :
- *transition-property* : la(les) propriété(s) CSS à modifier.
- *transition-duration* : durée de la transition (s ou ms)
- *transition-timing-function* : méthode d'interpolation
- *transition-delay* : durée avant le début de la transition



# Animations avec CSS


- Enchaînement de plusieurs transitions
- Consiste à modifier automatiquement les propriétés CSS et à appliquer des transformations 2D ou 3D selon un scénario.
  - On décrit les étapes
  - On décrit l'animation et on l'associe à un sélecteur (*balise, .class, #identifiant ou :hover....*)

# Mise en place d'une animation avec CSS

## 1) Description des étapes avec `@keyframes`


`@keyframes nom_animation1`

```
{ from { color:blue; }  
  33% { color:green; }  
  66% { color:yellow; }  
  to { color:red; }  
}
```



## 2) Description de l'animation (CSS)

`#identifiant {animation: nom_animation1 5s linear 1s;}`



## 3) Utilisation dans le contenu (body)

`<div id=identifiant >.....</div>`

# Paramètres de l'animation

*animation* : raccourci pour regrouper les propriétés ci-dessous

*animation-name* : nom de l'animation à utiliser.

*animation-duration* : durée totale de l'animation.

*animation-timing-function* : méthode d'interpolation

*animation-iteration-count* : nombre de répétition de l'animation.  
Utilisez *infinite* pour une animation en continu.

*animation-direction* : permet de jouer une animation en sens inverse

*animation-play-state* : mettre en pause l'animation.

*animation-delay* : Temps avant que l'animation ne démarre.

*animation-fill-mode* : conserver l'état de l'animation

## Pour la compatibilité écrire 5 fois...

```
@keyframes nom_animation1{ ... }
```

```
@-webkit-keyframes nom_animation1{ ... }
```

```
@-moz-keyframes nom_animation1{ ... }
```

```
@-ms-keyframes nom_animation1{ ... }
```

```
@-o-keyframes nom_animation1{ ... }
```

```
#identifiant {animation: nom_animation1 5s linear 1s;
```

```
    -webkit-animation: nom_animation1 5s linear 1 s;
```

```
    -moz-animation: nom_animation1 5s linear 1s;
```

```
    -ms-animation: nom_animation1 5s linear 1s;
```

```
    -o-animation: nom_animation1 5s linear 1s;
```

```
}
```

# Transformations 2D et 3D avec CSS

- Démonstrations avec des explications :
  - <http://www.css3create.com/animation>
  - <http://www.html5-css3.fr/css3/transformations-3d-css3>
  - <http://www.w3.org/TR/css3-3d-transforms/>
  - Etc...

## Travail à faire

- Réaliser une page avec :
  - un menu escamotable
  - un bloc contenant du texte et une photo qui tourne de 360 deg au survol de la souris, la transition devra se faire en 5 secondes
  - une animation personnalisée avec @keyframes
- Mettre la page en ligne et envoyer un mail avec le lien à l'enseignant lorsque ça fonctionne (jacky.senlis@u-picardie.fr)