ISEN ALL IS DIGITAL! OUEST | yncréa

Développement Web
- CSS-

Groupe des étudiants : CIR1

Transition

Transition

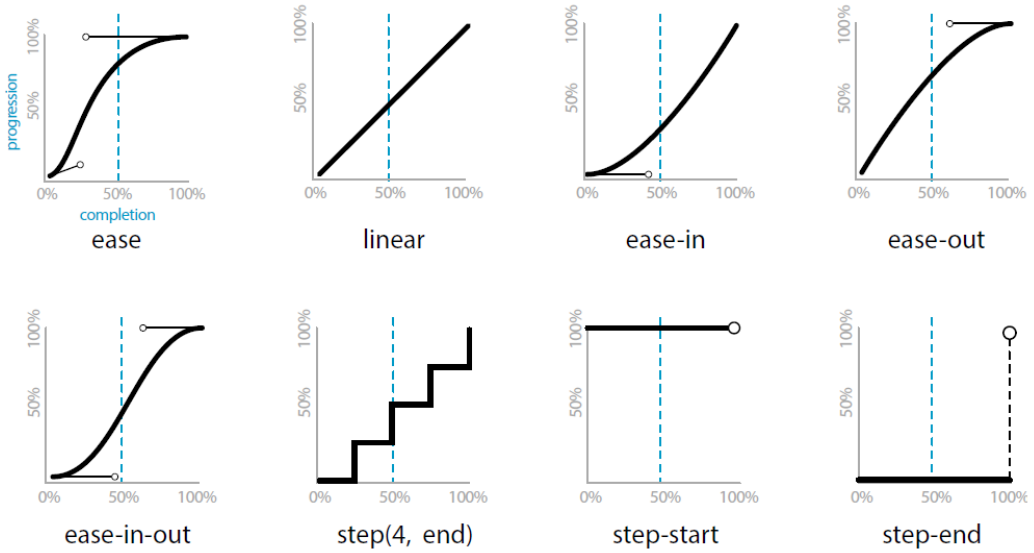
- Pour appliquer une transition en CSS, il faut spécifier :
  a. la propriété CSS à laquelle l'effet va être appliqué : transition-property
     - **valeurs** : property-name | all | none
     - **valeur par défaut** : all
     - **appliqué aux** : à tous les éléments et aux pseudo-éléments "::before" et "::after"

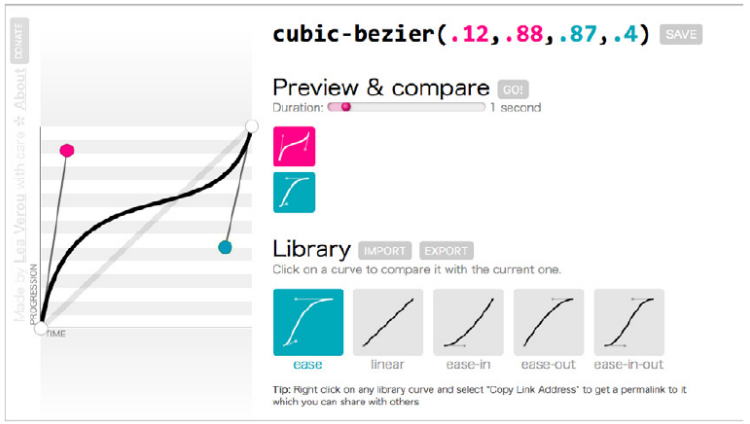
Transition

- Pour appliquer une transition en CSS, il faut spécifier :
  a. la propriété CSS à laquelle l'effet va être appliqué : transition-property
  b. la durée de l'effet : transition-duration
     - **valeurs** : time (en seconde)
     - **valeur par défaut** : 0s
     - **appliqué aux** : à tous les éléments et aux pseudo-éléments "::before" et "::after"

Transition

- Pour appliquer une transition en CSS, il faut spécifier :
  a. la propriété CSS à laquelle l'effet va être appliqué : transition-property
  b. la durée de l'effet : transition-duration
  c. accélération de la transition : transition-timing-function
     - **valeurs** : ease | linear | ease-in | ease-out | ease-in-out | step-start | step-end | steps | cubic-bezier(#,#,#,#)
     - **valeur par défaut** : ease
     - **appliqué aux** : à tous les éléments et aux pseudo-éléments "::before" et "::after"





https://cubic-bezier.com/

Transition

- Pour appliquer une transition en CSS, il faut spécifier :
  a. la propriété CSS à laquelle l'effet va être appliqué : transition-property
  b. la durée de l'effet : transition-duration
  c. accélération de la transition : transition-timing-function
  d. le temps à attendre entre le moment où la propriété est modifiée et le début de la transition : delay
     - **valeurs** : time en secondes
     - **valeur par défaut** : all
     - **appliqué aux** : à tous les éléments et aux pseudo-éléments "::before" et "::after"

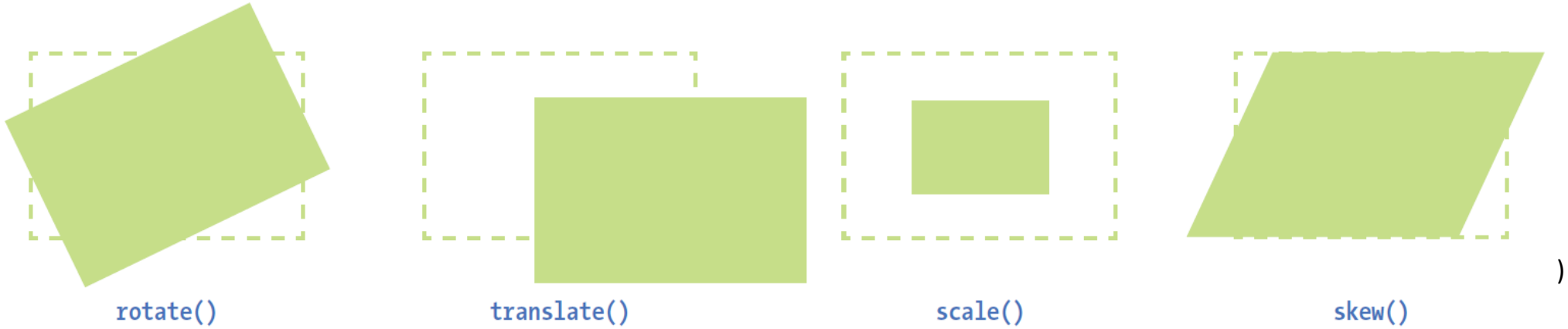
- **Exemple** :

https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition4

Transformation

Transformation 2D

- 4 transformations sont possibles :



- Propriété CSS :
  - **valeurs** : rotat)
    | skewY() | nc
  - **valeur par dé**
  - **appliqué aux** : éléments transformables

Rotation 2D

- Changement de l'origine du repère pour les opérations : translate
  - **valeurs** : percentage | length | left | center | right | top | bottom
  - **valeur par défaut** : 50% 50%
  - **appliqué aux** : éléments transformables

transform: rotate(degree);

```
transform-origin: center top;
```

Translation 2D

- **Syntaxe :**

```
transform: translate(translateX, translateY);
```

- **Exemple :**

```
transform: translate(90px, 60px);
```

Changement d'échelle

- **Syntaxe :**

```
transform: scale(scaleX, scaleY);
```

- **Exemple :**

```
transform: scale(1.5, .5);
```

Distortion

- **Syntaxe :**

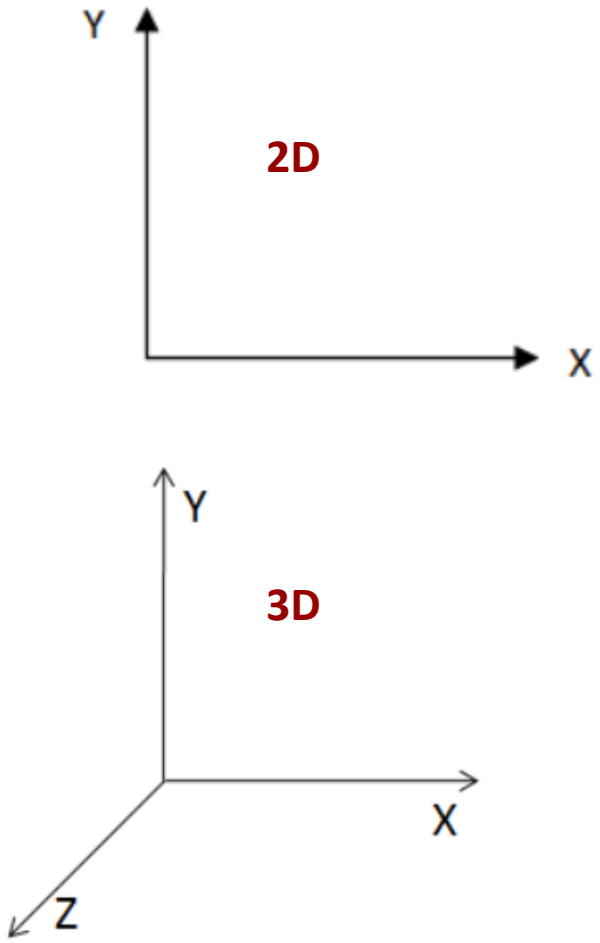
```
transform: skew(skewX, skewY);
```

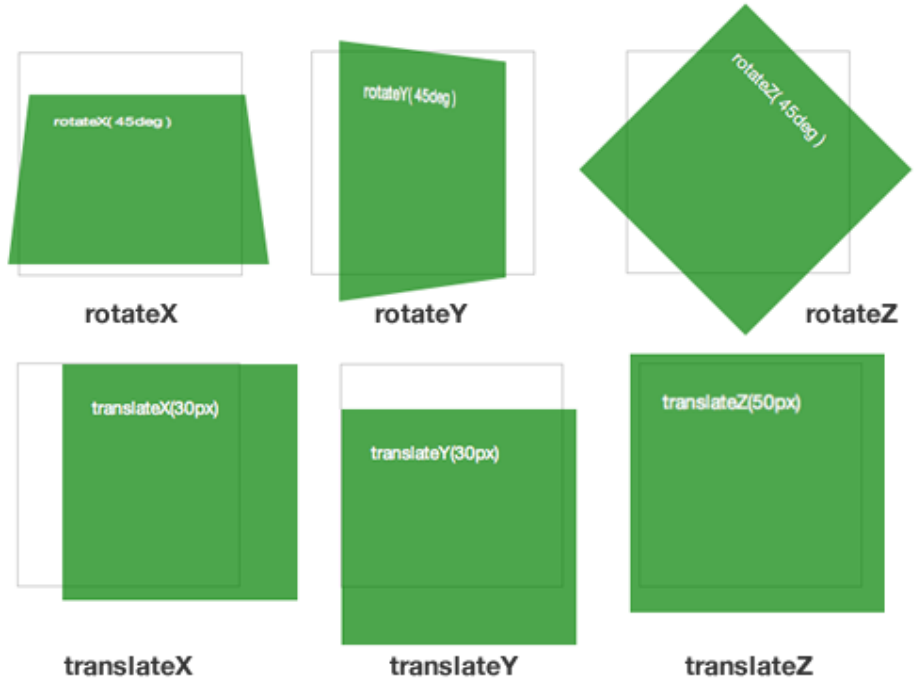
- **Exemple :**

```
transform: skew(15deg, 30deg);
```

Transformation 3D

- Les transformations 2D sont applicable aussi en 3D en ajoutant l'axe Z

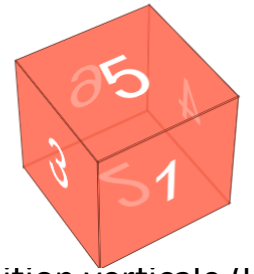




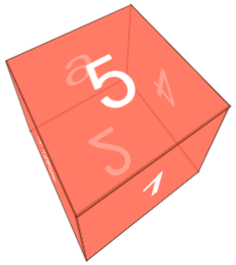
Perspective en 3D

- **Définition** : la perspective est la distance entre le plan d'équation z = 0 et la position de l'utilisateur => notion de profondeur de la scène 3D
  - **propriété** : perspective
  - **valeurs** : longueur
  - **valeur par défaut** : none
  - **appliqué aux** : éléments transformables

`perspective: 1000px`



`perspective: 250px`



- Position depuis laquelle on regarde la scène 3D
  - **propriété** : perspective-origin
  - **valeurs** : position horizontale (left, center, right, longueur, pourcentage), position verticale (left, center, right, longueur, pourcentage)
  - **valeur par défaut** : 50% 50%
  - **appliqué aux** : éléments transformables

`left`

`center`

`right`

Animation

Keyframes

- Pour appliquer une animation en CSS, il faut spécifier :
  - les règles keyframe avec @keyframes
  - la propriété CSS à laquelle l'animation va être appliquée
- Syntaxe :

```
@keyframes animation-name {
    keyframe { property: value; }
    /* d'autres keyframes */
}
```

- Exemple :

```
div {
  width: 100px;
  height: 100px;
  animation-name: colors;
  animation-duration: 20s;
}

@keyframes colors {
    0% { background-color: red; }
    20% { background-color: orange; }
    40% { background-color: yellow; }
    60% { background-color: green; }
    80% { background-color: blue; }
    100% { background-color: purple; }
}
```

Les propriété d'animation

animation-name	nom de l'animation
animation-duration	durée de l'animation
animation-timing-function	effet sur la progression de l'animation
animation-delay	durée de la pause avant l'exécution de l'animation
animation-iteration-count	nombre de répétition de l'animation
animation-direction	sens de lecture de l'animation
animation-fill-mode	moment de l'animation où les effets sont visibles

Exemple : https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation5