

# AWA1

## ANIMATION WEB

# Animations CSS

# Contenu

- ➔ 1. Transformations
- ➔ 2. Transitions
- ➔ 3. Animation
- ➔ 4. Annexe

# Browser support

## CSS Animation - WD

Global 89.54% + 0.06% = 89.61%








unprefixed: 65.16%

Complex method of animating certain properties of an element

Current aligned

Usage relative

Show all

| IE | Edge * | Firefox | Chrome   | Safari  | Opera | iOS Safari *  | Opera Mini * | Android Browser *   | Chrome for Android |
|----|--------|---------|--|---|-------|---|--------------|---|--------------------|
|    |        |         | 31  |   |       |   |              | 4.1    |                    |
| 8  |        | 38      | 43   |   |       |   |              | 4.3    |                    |
| 9  |        | 39      | 44   |   |       |   |              | 4.4    |                    |
| 10 |        | 40      | 45   | 8  |       | 8.4  |              | 4.4.4  |                    |
| 11 | 12     | 41      | 46   | 9   | 32    | 9   | 8            | 44  | 45                 |
|    | 13     | 42      | 47   |   | 33    |   |              |   |                    |
|    |        | 43      | 48   |   | 34    |   |              |   |                    |
|    |        | 44      | 49   |   |       |   |              |   |                    |

Notes

Known issues (6)

Resources (4)

Feedback

<sup>1</sup> Partial support in Android browser refers to buggy behavior in different scenarios.

# 1. Transformations CSS

# 1. Transformations CSS

**transform:scale(X,Y)** : Change la taille de l'élément, (hauteur,largeur).

Exemple:

```
transform.scale(2);
```

Multiplie la taille de la div par 2 (en X et en Y), la font grandira aussi.

scaleX() et scaleY(): Permettent d'agrandir l'élément sur un axe uniquement.

# 1. Transformations CSS (suite)

**transform:translate(X,Y)** : Déplace un élément (gauche-droite,haut-bas)

Exemple:

```
transform.translate(100px);
```

Déplace l'élément de 100px vers la droite.

translateX() ou translateY()

Permet de préciser l'axe de déplacement.

translate3d(x, y, z)

L'élément fait une translation sur 3 axes.

# 1. Transformations CSS (suite)

**transform:rotate():** Tourne l'élément (dans le sens des aiguilles d'une montre)

Exemple:

```
transform.rotate(25deg);  
transform.rotateZ(25deg);
```

Tourne l'élément de 20 degrés.

rotateX() ou rotateY()

Permet de faire une rotation sur l'axe X ou Y.

rotate3d(x, y, z)

Permet de faire une rotation sur 3 axes.

# 1. Transformations CSS (suite)

**transform:skewX() ou skewY()** : Penche l'élément, (parallélogramme)

Exemple:

```
transform.skewX(25deg);
```

Transforme l'élément en parallélogramme horizontal.



# 1. Transformations CSS (suite)

## **transform:matrix() :**

Généré par le navigateur quand il y a plusieurs transformations.

Exemple:

```
transform:matrix(0.7071067811865475, 0.7071067811865476, -  
0.7071067811865476, 0.7071067811865475, -0.7071067811865497,  
34.648232278140824)
```

## 2. Transitions CSS

## 2. Transitions CSS

Une transition CSS peut être ajustée avec 4 paramètres :

|  |                              |
|--|------------------------------|
| <code>transition-property: height;</code>      | // propriété CSS qui change  |
| <code>transition-duration: 2s;</code>          | // temps de la transition    |
| <code>transition-timing-function: ease;</code> | // accélération              |
| <code>transition-delay: 1s;</code>             | // délai avant la transition |

**ou :**

```
transition: height 2s ease 1s;
```

# Préfixes (vendor prefixes) ☹️

```
#madv{  
-webkit-transition: all 1s ease-in-out;  
-moz-transition: all 1s ease-in-out;  
-o-transition: all 1s ease-in-out;  
transition: all 1s ease-in-out;  
}
```

- Pour éviter d'alourdir le CSS en précisant pour chaque style les préfixes pour chaque navigateur, une méthode simple consiste à inclure dans la page, le script [prefixfree.min.js](https://raw.githubusercontent.com/LeaVerou/prefixfree/gh-pages/prefixfree.min.js) disponible ici :  
<https://raw.githubusercontent.com/LeaVerou/prefixfree/gh-pages/prefixfree.min.js>
- Ce script s'occupe d'ajouter les préfixes nécessaires pendant l'exécution.
- On pourrait aussi utiliser les préprocesseurs CSS : LESS ou SASS.

## 2. Transitions CSS (suite)

### transition property + transition duration

```
div {  
  width: 100px;  
  height: 200px;  
  background: red;  
  transition: width 2s, transform 4s;  
}  
  
div:hover {  
  width: 300px;  
  transform: rotate(360deg);  
}
```

- La propriété largeur de la div passe de 100px à 300px en 2secondes.
- La propriété transform tourne de 360 deg en 4 secondes.
- **Important:** La transition n'est pas spécifiée dans la div:hover, mais dans la div par défaut.

## 2. Transitions CSS (suite)

### transition delay

```
div {  
  width: 100px;  
  height : 200px;  
  background: red;  
  transition: all 2s;  
  transition-delay: 1s;  
}  
  
div:hover {  
  width: 300px;  
  background-image: url("paper.gif");  
}
```

Utilité d'un délai ?

# transition-timing-function

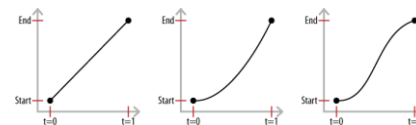
```
div {
  width: 100px;
  height : 200px;
  background: red;
  transition: all 2s;
}

#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}

div:hover {
  width: 300px;
  background: blue;
}
```

Le code html comprend 5 DIVS avec un id différent.

transition-timing-function décrit la courbe d'accélération.



Pour chaque div la durée des transitions est la même, mais l'accélération est différente

## 2. Transitions CSS (suite)

### Exemples:

<http://codepen.io/julienborrel/pen/yqwGA>

(Slide diagonal)

<http://codepen.io/yy/pen/Fqbks>

(Slide hexagonal x 4 )



# 3. Animation CSS

# 3. Animation CSS

## @keyframes

```
@keyframes example {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}
```

Définition d'une animation

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
}
```

Elément auquel on applique  
l'animation

## 3. Animation CSS (suite)

### @keyframes avec %

```
@keyframes example {  
  0%   {background-color: red; left:0px; top:0px;}  
  25%  {background-color: yellow; left:200px; top:0px;}  
  50%  {background-color: blue; left:200px; top:200px;}  
  75%  {background-color: green; left:0px; top:200px;}  
  100% {background-color: red; left:0px; top:0px;}  
}
```

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
}
```

Questquispasse?

# 3. Animation CSS (suite)

## Propriétés de l'animation

|                |  |                            |
|----------------|--|----------------------------|
| Nom >          | <code>animation-name: example;</code>                            |                            |
| Durée >        | <code>animation-duration: 5s;</code>                             |                            |
| Acceleration > | <code>animation-timing-function: linear;</code>                  | <code>//ease-in-out</code> |
| Délai >        | <code>animation-delay: 2s;</code>                                |                            |
| Répétitions >  | <code>animation-iteration-count: 3;</code>                       | <code>//infinite</code>    |
| Sens >         | <code>animation-direction: alternate;</code>                     | <code>//reverse</code>     |
|                | <code>}</code>   |                            |
|                | <br><code>div {</code>   |                            |
|                | <code>animation: example 5s linear 2s infinite alternate;</code> |                            |
|                | <code>}</code>   |                            |

## 3. Animation CSS (suite)

### Exemples:

<http://codepen.io/Venerons/pen/BvHbK>

(Animations circulaires)

<https://daneden.github.io/animate.css/>

(Animate.css : librairie d'animations pur CSS)

# 4. Annexe

## 4. Annexe : Formes CSS

```
#cercle {  
width: 100px;  
height: 100px;  
background: red;  
border-radius: 50%;  
}
```



```
#ovale {  
width: 200px;  
height: 100px;  
background: red;  
border-radius: 100% / 50%;  
}
```



## 4. Annexe : Formes CSS (suite)

```
#triangle-up {  
width: 0;  
height: 0;  
border-left: 50px solid transparent;  
border-right: 50px solid transparent;  
border-bottom: 100px solid red;  
}
```



```
// Pour #triangle-bas : border-top: 100px solid red;
```



## 4. Annexe : Formes CSS (suite)

```
#bulle {  
width: 120px;  
height: 80px;  
background: red;  
position: relative;  
border-radius: 10px;  
}
```



```
#bulle:before {  
content:"";  
position: absolute;  
right: 100%;  
top: 26px;  
width: 0;  
height: 0;  
border-top: 13px solid transparent;  
border-bottom: 13px solid transparent;  
border-right: 26px solid red;  
}
```

## 4. Annexe : Formes CSS (suite)

```
#hexagone { width: 100px; height: 55px; background: red; position: relative; }
```

```
#hexagone:before{  
  content: "";  
  position: absolute;  
  top: -25px;  
  left: 0;  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 25px solid red;  
}
```

```
#hexagone:after{  
  content: "";  
  position: absolute;  
  bottom: -25px;  
  left: 0;  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-top: 25px solid red;  
}
```



EXERCICE ➡