

Les animations

A quoi sert une animation ?

Une animation css sert à animer des elements html sans avoir besoin d'utiliser js ou flash.

Les animations sont supportées par la plupart des navigateurs modernes.

Les proprietes des animations

- ✓ @keyframes
- ✓ animation-name
- ✓ animation-duration
- ✓ animation-delay
- ✓ animation-iteration-count
- ✓ animation-direction
- ✓ animation-timing-function
- ✓ animation-fill-mode
- ✓ animation

Le principe de l'animation

Une animation est le fait qu'un element change de style progressivement au cours d'un laps de temps.

Pendant l'animation on peut changer autant de propriétés qu'on veut.

Pour pouvoir utiliser une animation, il faut spécifier les keyframes (points cles) de celle-ci.

Un keyframe est un l'état de l'element à un moment donné du deroulement de l'animation

La propriete keyframe

1- definir les points cles de l'animation et lui attribuer un nom : variant 1 : debut et fin

```
@keyframes changer-couleur {  
  from {background-color: greenyellow;}  
  to {background-color: skyblue;}  
}
```

Variant 2 : % cad étapes de deroulement de l'animation

```
@keyframes anime-degrade {  
  0% {background-color: rgb(136, 255, 0);}  
  25% {background-color: greenyellow;}  
  50% {background-color: skyblue;}  
  100% {background-color: rgb(23, 93, 173);}  
}
```

2-spécifier le nom de l'animation et les autres propriétés qui vont avec

```
selecteur {  
  width: 100px;  
  height: 100px;  
  background-color: blue;  
  
  animation-name: anime-degrade;  
  animation-duration: 4s;  
  animation-delay: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: normal;  
  animation-timing-function : ease-in;  
  animation-fill-mode: forwards;  
}
```

animation-name : le nom de l'animation qu'on a défini au préalable (avant ou après)

animation-duration : la durée de l'animation en secondes

animation-delay : le délai après lequel l'animation s'exécute

NB : le délai peut être négatif, si c'est le cas, l'animation va se comporter comme si elle a été lancée après ce délai

Exp : animation-delay : -2s ; // comme si l'animation a été lancée après 2s

animation-iteration-count : n ou infinie. Si n, l'animation va s'exécuter n fois, si infinie, elle va être exécutée pour l'infini.

animation-direction : le sens de l'exécution de l'animation : à l'endroit, à l'envers ou en alternance.

animation-timing-function : spécifie le mode d'accélération de la vitesse de l'exécution.

animation-fill-mode : le comportement avant et après l'animation : spécifie si ça revient à son état d'origine ou au dernier de l'animation (par défaut : none, ça revient à l'état d'origine).

NB : on peut écrire les lignes précédentes dans sur une seule d'une façon condensée

```
selecteur {  
    animation: anime-degrade 4s 2s infinite normal ease-in forwards;  
}
```

L'ordre n'a pas d'importance, à l'exception de la durée et du délai, la durée vient avant.

Avec js, il se peut qu'on ait besoin de jouer ou arrêter l'animation d'une façon interactive, pour ce faire, on dispose de la propriété animation-play-state et qui a les valeurs running/paused

Lien de doc complète sur les animations

https://www.w3schools.com/css/css3_animations.asp

Les animations et la performance

L'animation est un grand plus au côté esthétique du produit mais il a un coût sur la performance.

Donc il faut l'utiliser avec modération.

Les transitions

Une transition est le fait qu'un élément de style suite au changement d'état d'une façon fluide.

Pour définir une transition, il faut définir deux éléments essentiels :

1- la propriété à laquelle on effectue l'effet

2- la durée

Exemple :

```
p {  
    width: 200px;  
    height: 200px;  
    background: skyblue;  
    transition: width 2s;  
}  
p:hover {  
    width: 300px;  
}
```

NB : si on change une propriété autre que celle qu'on a définie dans transition, ça ne donne pas d'effet de transition (la prop change mais il n'y aura pas de transition).

On peut combiner les prop avec les mêmes ou différents délais

```
transition: width 1s, background 4s;
```

pour donner l'effet de transition à toutes les prop qui subissent un changement, on peut mettre all

```
transition: all 2s;
```

Autres propriétés de la transition

```
transition-timing-function: ease-in; // La vitesse de la transition
transition-delay: 1s; // délai
```

écriture concise

```
transition: all 2s ease-in 1s;
```

Les transformations 2D

Les transformations 2D permettent de traduire, pivoter, mettre à l'échelle et déformer des éléments.

```
transform: translate(200px); /* translation de 200px vers la droite */
transform: translate(50px, 100px); /* translation de 50px vers la droite et 100px vers le bas */
```

NB : il est possible d'utiliser des %. Pour faire la translation dans l'autre sens on peut utiliser des valeurs négatives pour la gauche et le haut.

```
transform: rotate(20deg); /* rotation de 20 degrés dans le sens des aiguilles */
```

NB : il est possible d'utiliser des valeurs négatives pour faire la rotation dans l'autre sens.

```
transform: scale(2, 3); /* agrandit la largeur de 2x et la hauteur de 3x */
```

NB : il est possible d'utiliser des valeurs décimales ou négatives. Les valeurs négatives vont être utilisées comme suit : la taille (width et height) aura la valeur absolue mais le sens du texte sera inversé (effet miroir).

```
transform: skew(20deg, 10deg); /* déformation de 20 deg vers la gauche et 10 deg vers le haut */
```

NB : il est possible d'utiliser des valeurs négatives pour faire une déformation dans l'autre sens.

Regrouper toutes les propriétés précédentes en une seule ligne

```
transform: matrix(1, -0.3, 0, 1, 0, 0); /*
matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY()) */
```

Les transformations 3D

```
transform: rotateX(150deg); /* rotation d'un élément de 150 degrés de l'axe des abscisses dans le sens des aiguilles */
```

NB : il est possible d'utiliser des valeurs négatives pour faire une rotation dans l'autre sens.

```
transform: rotateY(150deg); /* rotation d'un élément de 150 degrés de l'axe des ordonnées dans le sens des aiguilles */
```

```
transform: rotateZ(220deg); /* rotation verticale d'un selon l'axe des z */
```

Les effets de transition et de transformation peuvent être utilisés au sein des animations.

Bibliothèques d'animations

<https://freefrontend.com/css-animation-libraries/>

<https://animate.style/>

<https://www.toptal.com/developers/css3maker/css3-animation.html>

<https://speckyboy.com/css-animation/>

<https://animista.net/>

<https://webcode.tools/generators/css/keyframe-animation>

<https://www.theappguruz.com/tag-tools/web/CSSAnimations/>

<https://fr.makingcss.com/>

Animations avancées

<https://angrytools.com/css/animation/>

AOS

<https://github.com/michalsnik/aos>

demo :

<http://michalsnik.github.io/aos/>