

1. CSS (transitions et animations)

transitions css

Exemple simple :

```
p {
background-color:blue;
transition-property:background-color; /* Transition sur background-color*/
transition-duration:1s; /* Durée 1s */
}
p:hover{background-color:green;}
```

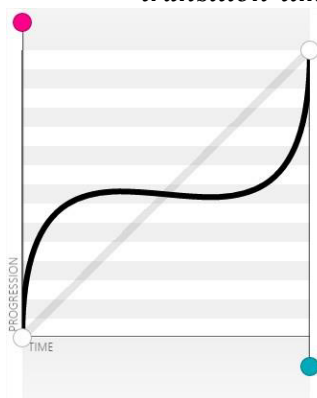
Paramétrages avancés sur une transition :

* **transition-timing-function** (accélérations éventuelles au cours du temps) :

<i>ease</i>	rapide au début, ralenti à la fin
<i>linear</i>	vitesse constante
<i>ease-in</i>	lent à l'approche , accélère à la sortie
<i>ease-out</i>	rapide à l'approche , ralenti à la sortie
<i>ease-in-out</i>	approche et sortie lente
<i>cubic-bezier</i>	courbe de bezier (selon paramétrage)

exemples : *transition-timing-function:ease;*

transition-timing-function:cubic-bezier(0,1.1,1,-0.1);



très rapide au début, légère régression, très rapide à la fin

Syntaxe synthétique : propriété "**transition**" regroupant plusieurs propriétés de transition:

```
{ transition: background-color 1s cubic-bezier(0,1.1,1,-0.1); }
```

et plusieurs transitions en même temps sont possibles :

```
{transition : background-color 2s ease , width 1s linear ;}
```

animations css

Au sein d'un fichier css, une animation se configure en "keyframes" (ayant un nom logique).

On y définit les étapes de l'animation (par exemples à 0 %, 50 %, 100 % du temps) :

@keyframes *monanimation*

```
{
  0% { transform: translateX(0px); }
  50% { transform: translateX(40px) rotate(-15deg) ;}
  100% { transform: translateX(80px) rotate(15deg); }
}
```

@keyframes *monrebond*

```
{
  0% { transform: translateY(0px) translateX(0px);}
  50% { transform: translateY(8px) translateX(2px);}
  100% { transform: translateY(0px) translateX(0px); }
}
```

On applique ensuite une animation via la propriété "animation" :

```
a:hover { color: blue; transition: color 1s linear; animation : monrebond 1s;}
button:hover { color: blue; animation : monrebond 1s;}

p:hover { background-color: #93db83; /* vert*/; animation: monanimation 2s ; }
```