

Les transitions en CSS

Les transitions CSS vont nous permettre de modifier la valeur d'une propriété CSS de manière fluide et selon une durée que l'on va pouvoir définir.

Nous allons commencer par étudier les propriétés complètes qui sont les suivantes :

- `transition-property;`
- `transition-duration;`
- `transition-timing-function;`
- `transition-delay.`

1. La propriété `transition-property` va nous permettre de définir quelles propriétés vont être les cibles de nos transitions, c'est-à-dire quelles sont les propriétés dont la valeur va devoir changer progressivement

```
height: 100px;
margin: 20px auto;
padding: 40px;
border: 2px solid black;
box-sizing: border-box;
background-color: orange;
}

.d1{
  transition-property: background-
color;
  transition-duration: 5s;
}
.d2{
  transition-property: background-
color, border;
  transition-duration: 5s;
}
.d3{
  transition-property: all;
  transition-duration: 5s;
}
.d1:hover, .d2:hover, .d3:hover{
  background-color: blue;
  border: 5px solid red;
  color: white;
}
```

Du texte dans le div 1

Du texte dans le div 2

Du texte dans le div 3

2. La propriété `transition-duration` va nous permettre de définir le temps que vont mettre les propriétés passées à `transition-property` pour passer d'une valeur de départ à une valeur d'arrivée. Nous allons pouvoir lui passer un nombre de secondes en valeur.
1. La propriété `transition-timing-function` va nous permettre de choisir la vitesse de la transition au sein de celle ci. Nous allons ainsi pouvoir créer des transitions totalement

linéaires ou, au contraire, créer des transitions qui vont s'accélérer ou ralentir au milieu. Nous allons pouvoir passer les valeurs suivantes à cette propriété :

- **ease** : valeur par défaut. Permet de créer une transition relativement lente au début puis qui s'accélère au milieu et qui se termine lentement ;
- **linear** : permet de créer une transition totalement linéaire c'est-à-dire qui va aller à la même vitesse du début à la fin ;
- **ease-in** : permet de créer une transition avec un départ lent puis qui s'accélère ensuite ;
- **ease-out** : permet de créer une transition qui va ralentir à la fin ;
- **ease-in-out** : permet de créer une transition lente au début puis qui s'accélère au milieu et qui se termine lentement. Ressemble fortement à **transition-timing-function : ease** mais démarre plus lentement ;
- **cubic-bezier(x1,y1,x2,y2)** : sert à créer une transition à la vitesse totalement personnalisé en renseignant une courbe de Bézier.

•

•

```
div{
  width: 50%;
  height: 60px;
  margin: 20px auto;
  border: 2px solid black;
  box-sizing: border-box;
  text-align: center;
  padding: 15px;
}

.d1, .d2, .d3, .d4, .d5, .d6{
  background-color: orange;
  transition-property: width;
  transition-duration: 2s;
}

.d1{
  transition-timing-function: ease;
}
.d2{
  transition-timing-function: ease-in;
}
.d3{
  transition-timing-function: ease-out;
}
.d4{
  transition-timing-function: ease-in-out;
}
.d5{
  transition-timing-function: linear;
}
.d6{
  transition-timing-function: cubic-bezier(0.8, 0, 1, 1);
}
```

transition-timing-function: ease

transition-timing-function: ease-in

transition-timing-function: ease-out

transition-timing-function: ease-in-out

transition-timing-function: linear

transition-timing-function: cubic-bezier(0.8, 0, 1, 1)

- La propriété **transition-delay** va nous permettre de définir quand la transition doit commencer à partir du moment où la nouvelle valeur est passée aux propriétés concernées par la transition. On va pouvoir lui passer une valeur en secondes.

```
div{
  width: 90%;
  height: 100px;
  margin: 20px auto;
  padding: 40px 0px;
  border: 2px solid black;
  box-sizing: border-box;
  background-color: orange;
  text-align: center;
}
.d1,.d2,.d3{
  transition-property: background-
color;
  transition-duration: 2s;
}
}
.d1{
  transition-delay: 0s;
}
.d2{
  transition-delay: 2s;
}
.d3{
  transition-delay: 4s;
}
}
```

transition-delay: 0s

transition-delay: 2s

transition-delay: 4s