

10 : LES TRANSITIONS

Explications :

Les transitions CSS permettent de contrôler la vitesse d'animation lorsque les propriétés CSS sont modifiées. Plutôt que le changement soit immédiat, on peut l'étaler sur une certaine période. Ainsi, si on souhaite passer un élément de blanc à noir, on pourra utiliser les transitions CSS afin que cette modification soit effectuée progressivement, selon une courbe d'accélération donnée.

Les animations qui utilisent des transitions entre deux états sont souvent appelées transitions implicites car l'état initial et l'état final sont définis implicitement par le navigateur.

Les transitions CSS vous permettent de choisir :

- Les propriétés à animer en les listant explicitement
- Le début de l'animation
- La durée de l'animation
- La façon dont la transition s'exécutera

[\[Liste des propriétés CSS que l'on peut animer\]](#)

Les propriétés CSS relatives aux transitions :

Les transitions CSS sont généralement contrôlées grâce à la propriété raccourcie `transition`. Les différents composants d'une transition CSS peuvent être décrits dans le détail grâce aux propriétés détaillées suivantes :

`transition-property` :

Cette propriété définit le nom des propriétés CSS pour lesquelles on veut appliquer des transitions. Seules les propriétés listées ici seront sujettes aux transitions. Les modifications appliquées aux autres propriétés seront instantanées.

`transition-duration` :

Cette propriété définit la durée de la transition. On peut définir une durée pour toutes les transitions ou une durée pour chacune des propriétés.

Ex : `transition-duration: 0.5s`

`transition-timing-function` :

Cette propriété définit une fonction qui décrit la façon dont les valeurs intermédiaires sont calculées. On utilise pour cela des fonctions de temporisation : `steps`, `linear`, `ease-in`, `ease-out`, `ease-in-out` et `step-end`.

Ex : `transition-timing-function: ease`

[\[Fonctions de temporisation\]](#)

`transition-delay` :

Cette propriété indique le temps à attendre entre le moment où la propriété est modifiée et le début de la transition.

Ex : transition-delay: 0.5s

La syntaxe de la propriété raccourcie transition est :

transition: <property> <duration> <timing-function> <delay>;

Le rôle de la taille des listes de valeurs :

Si la liste des valeurs pour une propriété est plus courte qu'une autre, les valeurs de la liste la plus courte seront répétées pour que la longueur réelle corresponde. Ainsi :

```
style.css

div {
  transition-property: opacity, left, top, height;
  transition-duration: 3s, 5s;
}
```

Sera équivalent à :

```
style.css

div {
  transition-property: opacity, left, top, height;
  transition-duration: 3s, 5s, 3s, 5s;
}
```

De même, si la liste est trop longue par rapport à transition-property, elle sera tronquée. Ainsi :

```
style.css

div {
  transition-property: opacity, left;
  transition-duration: 3s, 5s, 2s, 1s;
}
```

Sera équivalent à :

```
style.css

div {
  transition-property: opacity, left;
  transition-duration: 3s, 5s;
}
```

Utiliser les transitions pour accentuer les éléments pour un menu :

On utilise parfois CSS pour mettre en avant les éléments d'un menu lorsque l'utilisateur les survole avec sa souris. On peut facilement utiliser les transitions CSS pour améliorer l'effet obtenu.

Tout d'abord, on définit le menu en HTML :

```
index.html

<nav>
  <a href="#">Accueil</a>
  <a href="#">À propos</a>
  <a href="#">Contact</a>
  <a href="#">Liens</a>
</nav>
```

On construit le CSS pour définir l'apparence du menu :

```
style.css

a {
  color: #fff;
  background-color: #333;
  transition: all 1s ease-out;
}
a:hover,
a:focus {
  color: #333;
  background-color: #fff;
}
```

Ainsi, lorsque la souris survole l'élément, la couleur du texte et de l'arrière-plan change.

Détecter le début et la fin d'une transition :

L'évènement `transitionend` est déclenché lorsqu'une transition est terminée. C'est un objet `TransitionEvent` qui possède deux propriétés supplémentaires qu'un `Event` :

propertyName :

Une chaîne de caractères qui indique le nom de la propriété CSS pour laquelle la transition est terminée.

elapsedTime :

Un nombre flottant qui indique le nombre de secondes durant lesquelles la transition s'est déroulée. Cette valeur n'est pas modifiée par la valeur de `transition-delay`.

Comme pour les différents évènements, on pourra utiliser `eventtarget.addEventListener()` pour « écouter » cet événement :

```
el.addEventListener('transitionend', updateTransition, true);
```

Pour détecter le début d'une transition, on pourra utiliser l'évènement `transitionrun` qui est déclenché avant tout retardement et l'évènement `transitionstart` qui est déclenché après tout retardement :

```
el.addEventListener('transitionrun', signalStart, true);  
el.addEventListener('transitionstart', signalStart, true);
```

Note : L'évènement `transitionend` n'est pas déclenché si la transition est interrompue avant la fin de la transition si `display: none` ou si la valeur de la propriété est modifiée.