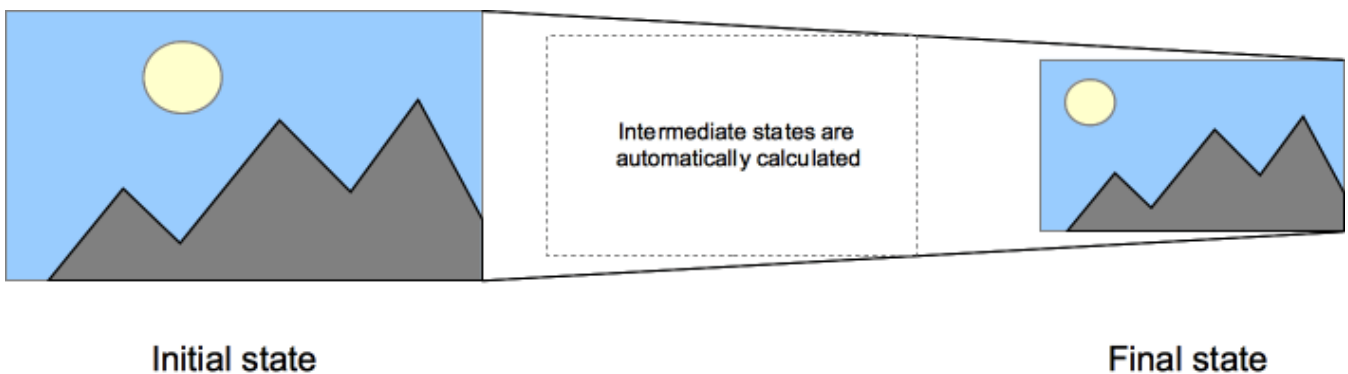




Utiliser les transitions CSS

Les transitions CSS permettent de contrôler la vitesse d'animation lors de la modification des propriétés CSS. Au lieu que les modifications de propriété prennent effet immédiatement, vous pouvez faire en sorte que les modifications d'une propriété aient lieu sur une période de temps. Par exemple, si vous changez la couleur d'un élément du blanc au noir, le changement est généralement instantané. Lorsque les transitions CSS sont activées, les changements se produisent à des intervalles de temps qui suivent une courbe d'accélération, qui peuvent tous être personnalisés.

Les animations qui impliquent une transition entre deux états sont souvent appelées *transitions implicites* car les états entre les états de début et de fin sont définis implicitement par le navigateur.



Les transitions CSS vous permettent de décider quelles propriétés animer (en les *répertoriant explicitement*), quand l'animation démarrera (en définissant un *délai*), combien de temps durera la transition (en définissant une *durée*) et comment la transition s'exécutera (en définissant une *fonction de temporisation*, par exemple linéairement ou rapide au début, lente à la fin).

Quelles propriétés CSS peuvent être transférées ?

L'auteur Web peut définir quelle propriété doit être animée et de quelle manière. Cela permet la création de transitions complexes. Comme cela n'a pas de sens d'animer certaines propriétés, la [liste des propriétés animables](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions) est limitée à un ensemble fini.

Remarque : L'ensemble des propriétés pouvant être animées évolue au fur et à mesure que la spécification évolue.

Remarque : La `auto` valeur est souvent un cas très complexe. La spécification recommande de ne pas animer de et vers `auto`. Certains agents utilisateurs, comme ceux basés sur Gecko, implémentent cette exigence et d'autres, comme ceux basés sur WebKit, sont moins stricts. L'utilisation d'animations avec `auto` peut conduire à des résultats imprévisibles, selon le navigateur et sa version, et doit être évitée.

Définition des transitions

Les transitions CSS sont contrôlées à l'aide de la [transition](#) propriété raccourcie. C'est la meilleure façon de configurer les transitions, car cela permet d'éviter plus facilement les paramètres de synchronisation, ce qui peut être très frustrant de devoir passer beaucoup de temps à déboguer en CSS.

Vous pouvez contrôler les composants individuels de la transition avec les sous-propriétés suivantes :

[transition-property](#)

Spécifie le ou les noms des propriétés CSS auxquelles les transitions doivent être appliquées. Seules les propriétés répertoriées ici sont animées pendant les transitions ; les modifications de toutes les autres propriétés se produisent instantanément comme d'habitude.

[transition-duration](#)

Spécifie la durée pendant laquelle les transitions doivent se produire. Vous pouvez spécifier une durée unique qui s'applique à toutes les propriétés pendant la transition, ou plusieurs valeurs pour permettre à chaque propriété de passer sur une période de temps différente.

[transition-timing-function](#)

Spécifie une fonction pour définir comment les valeurs intermédiaires des propriétés sont calculées. *Les fonctions de synchronisation* déterminent comment les valeurs intermédiaires de la transition sont calculées. La plupart [des fonctions de synchronisation](#) peuvent être spécifiées en fournissant le graphique de la fonction

correspondante, tel que défini par quatre points définissant un Bézier cubique. Vous pouvez également choisir l'assouplissement de

[Aide-mémoire sur les fonctions d'assouplissement](#) .

transition-delay

Définit le temps d'attente entre le moment où une propriété est modifiée et le début de la transition.

La syntaxe CSS abrégée s'écrit comme suit :

```
div {  
  transition: <property> <duration> <timing-function> <delay>;  
}
```

Exemples

Exemple simple

Cet exemple effectue une transition de taille de police de quatre secondes avec un délai de deux secondes entre le moment où l'utilisateur passe la souris sur l'élément et le début de l'effet d'animation :

```
#delay {  
  font-size: 14px;  
  transition-property: font-size;  
  transition-duration: 4s;  
  transition-delay: 2s;  
}  
  
#delay:hover {  
  font-size: 36px;  
}
```

Exemple de propriétés animées multiples

Contenu CSS

```
.box {  
  border-style: solid;  
  border-width: 1px;  
  display: block;  
  width: 100px;  
  height: 100px;  
  background-color: #0000FF;
```

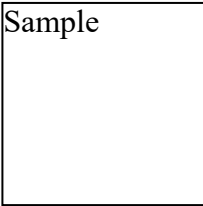
```
transition: width 2s, height 2s, background-color 2s, transform 2s;
}

.box:hover {
  background-color: #FFCCCC;

  width: 200px;
  height: 200px;
  transform: rotate(180deg);
}
```

The box below combines transitions for: width, height, background-color, transform. Hover over the box to see these properties animated.

Sample



Lorsque les listes de valeurs de propriété sont de longueurs différentes

Si la liste de valeurs d'une propriété est plus courte que les autres, ses valeurs sont répétées pour les faire correspondre. Par exemple:

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



Ceci est traité comme s'il s'agissait :

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



De même, si la liste de valeurs d'une propriété est plus longue que celle de [transition-property](#), elle est tronquée, donc si vous avez le CSS suivant :

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

Ceci est interprété comme :

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

Utilisation de transitions lors de la mise en surbrillance des menus

Une utilisation courante de CSS consiste à mettre en surbrillance des éléments dans un menu lorsque l'utilisateur passe le curseur de la souris dessus. Il est facile d'utiliser des transitions pour rendre l'effet encore plus attrayant.

Tout d'abord, nous configurons le menu en HTML :

```
<nav>  
  <a href="#">Home</a>  
  <a href="#">About</a>  
  <a href="#">Contact Us</a>  
  <a href="#">Links</a>  
</nav>
```

Ensuite, nous construisons le CSS pour implémenter l'apparence de notre menu :

```
nav {  
  display: flex;  
  gap: 0.5rem;  
}  
  
a {  
  flex: 1;  
  background-color: #333;  
  color: #fff;  
  border: 1px solid;  
  padding: 0.5rem;  
  text-align: center;  
  text-decoration: none;  
  transition: all 0.5s ease-out;  
}
```

```
}  
  
a:hover,  
a:focus {  
    background-color: #fff;  
  
    color: #333;  
}
```

Ce CSS établit l'apparence du menu, avec les couleurs d'arrière-plan et de texte changeant toutes les deux lorsque l'élément est dans son état [:hover](#) et [:focus](#) indique :

[Home](#)[About](#)[Contact Us](#)[Links](#)

Exemples JavaScript

Remarque : Des précautions doivent être prises lors de l'utilisation d'une transition immédiatement après :

- ajouter l'élément au DOM en utilisant `.appendChild()`
- supprimer la `display: none;` propriété d' un élément .

Ceci est traité comme si l'état initial ne s'était jamais produit et que l'élément était toujours dans son état final. Le moyen le plus simple de surmonter cette limitation consiste à appliquer une `window.setTimeout()` poignée de millisecondes avant de modifier la propriété CSS vers laquelle vous souhaitez effectuer la transition.

Utiliser des transitions pour rendre la fonctionnalité JavaScript fluide

Les transitions sont un excellent outil pour rendre les choses beaucoup plus fluides sans avoir à faire quoi que ce soit à votre fonctionnalité JavaScript. Prenons l'exemple suivant.

```
<p>Click anywhere to move the ball</p>  
<div id="foo" class="ball"></div>
```



En utilisant JavaScript, vous pouvez faire en sorte que la balle se déplace vers une certaine position :

```
var f = document.getElementById('foo');
document.addEventListener('click', function(ev){
    f.style.transform = 'translateY('+(ev.clientY-25)+'px)';
    f.style.transform += 'translateX('+(ev.clientX-25)+'px)';
}, false);
```

Avec CSS, vous pouvez le rendre fluide sans aucun effort supplémentaire. Ajoutez une transition à l'élément et tout changement se fera en douceur :

```
.ball {
    border-radius: 25px;
    width: 50px;
    height: 50px;
    background: #c00;
    position: absolute;
    top: 0;
    left: 0;
    transition: transform 1s;
}
```

Click anywhere to move the ball

Détecter le début et la fin d'une transition

Vous pouvez utiliser l' [transitionend](#) événement pour détecter qu'une animation est terminée. Il s'agit d'un [TransitionEvent](#) objet, qui a deux propriétés supplémentaires au-delà d'un [Event](#) objet typique :

propertyName

Une chaîne indiquant le nom de la propriété CSS dont la transition s'est terminée.

elapsedTime

Flottant indiquant le nombre de secondes d'exécution de la transition au moment où l'événement s'est déclenché. Cette valeur n'est pas affectée par la valeur de [transition-delay](#).

Comme d'habitude, vous pouvez utiliser la [addEventListener\(\)](#) méthode pour surveiller cet événement :

```
e1.addEventListener("transitionend", updateTransition, true);
```

Vous détectez le début d'une transition en utilisant [transitionrun](#) (se déclenche avant tout délai) et [transitionstart](#) (se déclenche après tout délai), de la même manière :

```
e1.addEventListener("transitionrun", signalStart, true);  
e1.addEventListener("transitionstart", signalStart, true);
```

Remarque : L' [transitionend](#) événement ne se déclenche pas si la transition est abandonnée avant la fin de la transition, car l'élément est créé ou la valeur de la propriété d'animation est modifiée. [display](#) : none

Caractéristiques

spécification	Statut	Commenter
Transitions CSS	Document de travail	Définition initiale

Voir également

- L' [TransitionEvent](#) interface et l' [transitionend](#) événement.
- [Utiliser des animations CSS](#)

Dernière modification : 5 novembre 2021, [par les contributeurs MDN](#)