



CSS transition: saiba como usar transições em um elemento!

Última atualização 24 de junho de 2021

O **CSS transition** está entre as propriedades que surgiram com a nova versão da linguagem de [estilo CSS](#), o CSS3. Na prática, ele permite criar animações sem a necessidade de escrever [código JavaScript](#) ou de utilizar recursos externos. Assim, podemos criar inúmeros efeitos de movimento nos elementos e deixar a [aplicação web](#) mais interativa e sofisticada.

Apesar de existirem diversas ferramentas online que criam códigos de animação em CSS rapidamente, é importante compreender o que acontece com os elementos durante esse processo. Para demonstrar como esse recurso funciona, preparamos este post que contém os seguintes tópicos:

Índice

01 | O que é a propriedade transition?



03 | Quais os valores da propriedade transition?

04 | Exemplo de uso da propriedade transition

05 | Qual a compatibilidade com os navegadores?

Continue com a gente e boa leitura!

O que é a propriedade transition?

A propriedade **transition** é utilizada para realizar a modificação do estado de um elemento de forma gradativa e de acordo com as características definidas em seus atributos.

Ao posicionar o mouse sobre um [elemento HTML](#), por exemplo, acontece o estado **hover**, no qual podemos criar um [código CSS](#) para modificar as propriedades do elemento e apresentar modificações, como mudar a cor do texto ou o tamanho da fonte.

Entretanto, as modificações no estado **hover** são realizadas instantaneamente. Com a propriedade **transition CSS hover** podemos fazer com que elas demorem um pouco mais para serem realizadas, o que permite a criação do [efeito de animação](#).

Qual é a sua sintaxe?

A sintaxe da propriedade [CSS transition](#) é:

```
transition: transition-property transition-duration transition-timing-function transition-delay
```



- **transition-property:** corresponde ao nome da propriedade que será modificado quando o processo de transição for iniciado;
- **transition-duration:** determina o tempo para executar a transição;
- **transition-timing-function:** indica o formato da transição entre o estado inicial e o final;
- **transition-delay:** representa, em milissegundos, o tempo de espera para iniciar o processo de transição.

A definição do efeito de transição pode ser feita em uma única linha, conforme a sintaxe acima ou informando cada uma das propriedades de forma individual.

Quais os valores da propriedade transition?

A propriedade **CSS animation transition** contém uma série de propriedades auxiliares que são utilizadas para definir as características da animação. Confira como elas funcionam!

Definindo a qual propriedade a transição será aplicada: transition-property

A propriedade **transition-property** é utilizada para indicar qual ou quais propriedades serão modificadas durante o processo de transição do elemento. Os valores possíveis são os nomes das propriedades separados por vírgula ou valor **"all"**, que é o padrão da propriedade e indica que todas terão o efeito de transição.

Definindo a duração da transição: transition-duration



em segundos. O valor padrão é zero, o que significa que a animação é realizada imediatamente, portanto, sem o efeito de transição gradativa. Por isso, é importante estipular o tempo de duração para ter o comportamento desejado.

Definindo a curva de velocidade do efeito de transição: transition-timing-function

O efeito de transição pode ser aplicado em diversas propriedades CSS. Podemos, por exemplo, deslocar um elemento de um lado para o outro da tela por meio de propriedades de posicionamento ou aumentar o tamanho da fonte com a modificação da propriedade **font-size**. Nesses casos, a propriedade **transition-timing-function** representa a curva de velocidade em que a transição é realizada.

É possível, por exemplo, iniciar o deslocamento de forma lenta, acelerar no meio do caminho e terminar o processo devagar novamente. Para isso, o CSS utiliza a *Easing Functions*, que calcula a velocidade para a transição. Existem algumas predefinidas que podemos utilizar no atributo **transition-timing-function**, são elas:

- **linear**: representa a transição constante do início ao fim;
- **ease**: esse é o valor padrão e indica o efeito de transição com o início lento, durante rápido e no final, lento novamente;
- **ease-in**: indica a transição lenta apenas no início;
- **ease-out**: realiza a transição lenta apenas no final;
- **ease-in-out**: realiza a transição com o início e final lentos;
- **cubic-bezier()**: é uma função para indicar a curva de velocidade e é definida em quatro pontos, nos quais o primeiro e o último representam o início e o fim da transição.

Definindo quando a transição começa: transition-delay

A propriedade CSS **transition-delay** representa um tempo de atraso até que a transição se inicie. Seu valor pode ser definido em



Newsletter da Trybe

Junte-se a mais de 100.000 pessoas da nossa tribo que recebem conteúdos gratuitos e exclusivos em nossa newsletter semanal!

Quero me inscrever!

Exemplo de uso da propriedade transition

Nada melhor que visualizar na prática os efeitos da propriedade **transition**. No código de exemplo abaixo criamos um container com várias caixas feitas com o elemento [HTML <div>](#) e aplicamos os diferentes estilos de transição indicados no tópico sobre a propriedade **transition-timing-function**, que indica a curva de velocidade do efeito de transição.

Perceba que na classe **.box** definimos a propriedade **position** como **relative** para que os elementos sigam a orientação do posicionamento. A seguir, indicamos que a posição inicial deve ser a de **left** igual a zero, ou seja, totalmente posicionados à esquerda.

Já na classe **.container:hover .box**, indicamos que, ao posicionar o mouse sobre o elemento, a propriedade **left** terá o valor de 60%. Assim, cada **box** sofrerá esse deslocamento em relação à posição inicial. A seguir, informamos a propriedade que sofrerá a modificação e o tempo da transição: **transition: left 4s**.



para indicar cada uma delas. Confira como ficou o [código fonte](#).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>CSS Transition</title>
  <style>
    .container{
      display: flex;
      flex-direction: column;
      width:90%;
      height: 600px;
      border: 2px solid green;
      color: #fff;
      margin: 0 auto;
    }
    h1{
      color: green;
      margin:0 auto;
    }
    .box{
      width:30%;
      height: 80px;
      background-color:green;
      margin:10px;
      position:relative;
      left:0;
      flex-wrap: wrap;
    }
    .container:hover .box {
      left: 60%;
      -webkit-transition: left 4s;
      -moz-transition: left 4s;
      -ms-transition: left 4s;
      -o-transition: left 4s;
      transition: left 4s;
    }
    .container .box.linear {
      transition-timing-function: linear;
    }
    .container .box.ease {
```



```
        transition-timing-function: ease-in;
    }
    .container .box.ease-out {
        transition-timing-function: ease-out;
    }
    .container .box.ease-in-out {
        transition-timing-function: ease;
    }
    .container .box.cubic-bezier {
        transition-timing-function: cubic-
bezier(.71,.01,.53,.89);
    }
</style>
</head>
<body>
<div class="container">
    <h1>CSS Transition</h1>
    <div class="box ease">Transition ease - padrão</div>
    <div class="box linear">Transition linear</div>
    <div class="box ease-in">Transition ease-in</div>
    <div class="box ease-out">Transition ease-out</div>
    <div class="box ease-in-out">Transition ease-in-out</div>
    <div class="box cubic-bezier">Transition cubic-bezier()</div>
</div>
</body>
</html>
```

Qual a compatibilidade com os navegadores?

A propriedade **transition** não funciona com esse mesmo nome em todos os navegadores, especialmente os mais antigos. Por isso, a recomendação é utilizar os prefixos antes da propriedade, conforme fizemos no código de exemplo acima ao definir a classe **.container: hover .box**.

Além disso, ao desenvolver [aplicações web](#), é importante conferir se os recursos utilizados são compatíveis com os navegadores. Dessa forma, podemos evitar que [ocorram falhas](#) durante a [utilização da página pela pessoa usuária](#). Confira a partir de qual versão a propriedade **transition** é suportada pelos principais navegadores:



-
- Firefox: 4;
 - Chrome: 3.1;
 - Safari: 5;
 - Opera: 11.5;
 - iOS Safari: 3.2;
 - Android Browser: 2.1;
 - Opera Mobile: 12;
 - Chrome for Android: 88;
 - Firefox for Android: 85;
 - Samsung Internet: 4.

A propriedade **CSS transition** é um poderoso recurso para a criação de efeitos de animação nos elementos [HTML](#). Ele contém uma série de propriedades adicionais que ajudam a tornar o efeito de transição personalizado, como o uso da propriedade **transition-timing-function** associada à função **cubic-bezier()**.

Gostou do nosso conteúdo sobre o uso da propriedade CSS transition? Então, confira nosso [tutorial completo sobre HTML](#)!

Michelle Horn

DEIXE UM COMENTÁRIO

Você precisa fazer o [login](#) para publicar um comentário.

CSS: o que é, guia sobre como usar e vantagens!

CSS Border

CSS Color

CSS Background-image

CSS Display

CSS Gradient Linear e Radial

[CSS Padding](#)[CSS Position](#)[CSS Transition](#)[Currículo](#)[Dúvidas](#)[Trabalhe Conosco](#)[Gerador de CPF](#)[Pague só quando trabalhar](#)[Guia HTML](#)[Guia Javascript](#)[Guia Soft skills](#)[Carreira](#)[Tecnologia](#)[Desenvolvimento Web](#)[Linguagens de Programação](#)[Framework de Programação](#)[TXN](#)[Ferramentas](#)[Desabilitar cookies](#)[Política de Privacidade](#)