

[◀ VOLTAR](#)

# Animações e Transições

Neste tópico iremos aprender como criar animações, transições e transformações utilizando HTML5 e CSS3.

NESTE TÓPICO



Marcar  
tópico



## Criando animações.

Até pouco tempo atrás eram necessárias diversas ferramentas e muita programação para se criar animações e transições para nossos sites. Mas isto mudou com as novas implementações advindas da chegada do CSS3 que nos permitem criar animações e transições sem utilizar o javascript e o Flash.

Uma animação nada mais é do que um elemento que muda gradualmente de um estilo para outro, sendo que podemos alterar quantas propriedades quisermos e diversas vezes.

Para criar animações nós devemos criar quadros-chave para determinar os estilos que os elementos terão em cada momento da animação ou transição.

## O seletor CSS @keyframes

Quando especificamos em CSS um seletor `@keyframes` estamos informando ao navegador de onde a animação começa com o `from` (a partir de) até onde ela vai com o `to` (para).

Para que uma animação funcione corretamente temos de vincular a animação a um elemento e para isto precisamos utilizar a propriedade em CSS `animation`.

Vejamos o código abaixo para entender melhor quais são estes comandos e para que servem.

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Animações</title>
6.     <style>
7.       /* Código da animação */
8.       @keyframes teste { /* criação do @keyframes com nome teste */
9.         from {background-color: #00F;float: left;}/* inicio da animação
10.        */
11.        to {background-color: #F00;float:right;}/* fim da animação */
12.      }
13.      /* Elemento que vamos aplicar a animação */
14.      div {
15.        width: 300px;
16.        height: 200px;
17.        background-color: #00F;
18.        animation-name: teste; /* vinculação ao @keyframes pelo nome */
19.        animation-duration: 10s; /* tempo da animação */
20.      }
21.    </style>
22.  </head>
23.  <body>
24.    <div></div>
25.
26.  </body>
27. </html>
```



No caso de nosso exemplo acima temos que o elemento div chamado teste iniciará com a cor azul e estará na esquerda. Depois de 10 segundos ele se modificará para vermelho e mudará de posição para a direita da tela.

## VAMOS FAZER UM EXERCÍCIO?

Faça um exercício de fixação seguindo os seguintes passos:

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

## VAMOS TREINAR?

Treine a criação de animação como a do exemplo acima [clicando aqui](#).  
([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation1](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation1))

## IMPORTANTE:

Observe que em todos os exemplos disponibilizados nos exercícios do "Vamos treinar?" há a inserção dos mesmos comandos de animação antecedidos por -webkit-. Isto ocorre para que as animações sejam corretamente visualizadas no navegador Safari ok?



Abaixo temos os prefixos de cada navegador:

- Google Chrome e Safari: -webkit-
- Mozilla Firefox: -moz-
- Internet Explorer: -ms-
- Opera: -o-

*Lembrando que esses prefixos são temporários até que o W3C valide os comandos CSS3. Devido a isto você poderá encontrar nos exemplos e exercícios online estes prefixos antecedendo as propriedades de forma que elas sejam duplicadas como, por exemplo, no exercício acima:*

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    -webkit-animation-name: example; /* Safari 4.0 - 8.0 */  
    -webkit-animation-duration: 4s; /* Safari 4.0 - 8.0 */  
    animation-name: example;  
    animation-duration: 4s;  
}  
  
/* Safari 4.0 - 8.0 */  
@-webkit-keyframes example {  
    from {background-color: red;}  
    to {background-color: yellow;}  
}  
  
/* Standard syntax */  
@keyframes example {  
    from {background-color: red;}  
    to {background-color: yellow;}  
}
```

*Nos nossos exemplos deste tópico utilizaremos a formatação direta, sem a utilização destes prefixos ou seja, utilizaremos apenas o prefixo padrão /\* Standard syntax \*/ para que as explicações fiquem mais claras e fáceis de entender, ok?*

## Utilizando porcentagens

Nós também podemos utilizar porcentagens para fazer com que as mudanças ocorram diversas vezes. Vamos ver como isto é aplicado no exemplo abaixo.

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Animações</title>
6.     <style>
7.       /* Código da animação */
8.       @keyframes teste { /* criação do @keyframes com nome teste */
9.         0%   {background-color:#FF0; left:0px; top:0px;}/* posição inicial */
10.        25%  {background-color:#F00; left:400px; top:0px;}
11.        50%  {background-color:#0F0; left:400px; top:400px;}
12.        75%  {background-color:#F0F; left:0px; top:400px;}
13.        100% {background-color:#FF0; left:0px; top:0px;}/* volta a posição inicial */
14.      }
15.
16.      /* Elemento que vamos aplicar a animação */
17.      div {
18.        width: 300px;
19.        height: 200px;
20.        position: relative; /* definição da posição do elemento */
21.        background-color: #00F;
22.        animation-name: teste; /* vinculação ao @keyframes pelo nome */
23.        animation-duration: 10s; /* tempo da animação */
24.      }
25.    </style>
26.  </head>
27.  <body>
28.    <div></div>
29.
30.  </body>
31. </html>
```



No exemplo acima temos uma div que se movimenta pela tela ao mesmo tempo que muda de cor.

### VAMOS FAZER UM EXERCÍCIO?

Faça um exercício de fixação seguindo os seguintes passos:

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

## Atrasando uma animação

Quando precisamos que uma animação demore um pouco para começar utilizamos a propriedade animation-delay.

Vamos ver esta propriedade aplicada a nossa animação.



### VAMOS TREINAR?

Treine esta animação clicando aqui.  
([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation3](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation3))

```

1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.      <head>
4.          <meta charset="utf-8"/>
5.          <title>Animações</title>
6.          <style>
7.              /* Código da animação */
8.              @keyframes teste { /* criação do @keyframes com nome teste */
9.                  0% {background-color:#FF0; left:0px; top:0px;}/* posição inicial */
10.                 25% {background-color:#F00; left:400px; top:0px;}
11.                 50% {background-color:#0F0; left:400px; top:400px;}
12.                 75% {background-color:#F0F; left:0px; top:400px;}
13.                 100% {background-color:#FF0; left:0px; top:0px;}/* volta a posição inicial */
14.             }
15.
16.             /* Elemento que vamos aplicar a animação */
17.             div {
18.                 width: 300px;
19.                 height: 200px;
20.                 position: relative; /* definição da posição do elemento */
21.                 background-color: #00F;
22.                 animation-name: teste; /* vinculação ao @keyframes pelo nome */
23.                 animation-duration: 10s; /* tempo da animação */
24.                 animation-delay: 5s; /* tempo de espera para iniciar a animação */
25.             }
26.         </style>
27.     </head>
28.     <body>
29.         <div></div>
30.
31.     </body>
32. </html>
```

### VAMOS FAZER UM EXERCÍCIO?

Faça um exercício de fixação seguindo os seguintes passos:

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).



## VAMOS TREINAR?

Treine a utilização do animation-delay clicando [aqui](#).  
[https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation\\_delay](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_delay)

## Definindo a repetição da animação

Nós podemos definir quantas vezes uma animação irá se repetir, para isto utilizamos a propriedade CSS animation-iteration-count.

Vamos ver como ela funciona no exemplo abaixo:

```

1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.      <head>
4.          <meta charset="utf-8"/>
5.          <title>Animações</title>
6.          <style>
7.              /* Código da animação */
8.                  @keyframes teste { /* criação do @keyframes com nome teste */
9.                      0% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* posição inicial */
10.                     25% {background-color:#F00; left:400px; top:0px;width:200px;height:100px;}
11.                     50% {background-color:#0F0; left:400px; top:400px;width:100px;height:50px;}
12.                     75% {background-color:#F0F; left:0px; top:400px;width:200px;height:100px;}
13.                     100% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* volta a posição inicial */
14.                 }
15.
16.             /* Elemento que vamos aplicar a animação */
17.             div {
18.                 width: 300px;
19.                 height: 200px;
20.                 position: relative; /* definição da posição do elemento */
21.                 background-color: #00F;
22.                 animation-name: teste; /* vinculação ao @keyframes pelo nome */
23.                 animation-duration: 10s; /* tempo da animação */
24.                 animation-delay: 5s; /* tempo de espera para iniciar a animação
25.                 */
26.             }
27.         </style>
28.     </head>
29.     <body>
30.         <div></div>
31.
32.     </body>
33. </html>

```



Observe que no exemplo acima, além da repetição com a propriedade `animation-iteration-count`, também alteramos o tamanho da div em cada momento da animação.

## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

## VAMOS TREINAR?

Treine a utilização da propriedade animation-iteration-count [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_count2) ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation\\_count2](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_count2)).



## Definindo o sentido de uma animação

Você definir o sentido de uma animação utilizando a propriedade animation-direction. Podemos inverter a posição (sentido) com o valor reverse ou definir que ele deve ir pra frente e para trás com o valor alternate.

Vamos ver com esta propriedade funciona no exemplo abaixo em que utilizamos o valor alternate.

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Animações</title>
6.     <style>
7.       /* Código da animação */
8.       @keyframes teste { /* criação do @keyframes com nome teste */
9.         0% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* posição inicial */
10.        25% {background-color:#F00; left:400px; top:0px;width:200px;height:100px;}
11.        50% {background-color:#0F0; left:400px; top:400px;width:100px;height:50px;}
12.        75% {background-color:#F0F; left:0px; top:400px;width:200px;height:100px;}
13.        100% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* volta a posição inicial */
14.      }
15.
16.      /* Elemento que vamos aplicar a animação */
17.      div {
18.        width: 300px;
19.        height: 200px;
20.        position: relative; /* definição da posição do elemento */
21.        background-color: #00F;
22.        animation-name: teste; /* vinculação ao @keyframes pelo nome */
23.        animation-duration: 10s; /* tempo da animação */
24.        animation-delay: 5s; /* tempo de espera para iniciar a animação
*/
25.        animation-iteration-count: 4; /* define o número de vezes que a animação ocorrerá. Caso deseje que ela funcione infinitamente utilize o valor infinito */
26.        animation-direction: alternate; /* define o sentido da animação. Pode ser reverse ou alternate */
27.      }
28.    </style>
29.  </head>
30.  <body>
31.    <div></div>
32.
33.  </body>
34. </html>
```



## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

## VAMOS TREINAR?

Treine a utilização do animation-direction: alternate [clicando aqui.](#)  
([https://www.w3schools.com/css/tryit.asp?  
filename=trycss3\\_animation\\_direction2](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_direction2))

Treine a utilização do animation-direction: reverse [clicando aqui.](#)  
([https://www.w3schools.com/css/tryit.asp?  
filename=trycss3\\_animation\\_direction](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_direction))



## Definindo diferentes tempos de execução para a animação

Podemos definir uma "curva" de tempo para a animação, ou seja, definir como ela irá se comportar durante o tempo que definimos com a propriedade animation-duration.

Para isto utilizamos a propriedade animation-timing-function. Segundo o W3schools (ONLINE) esta propriedade pode receber os seguintes valores:

- ease - define uma animação com um início lento, então rápido, e depois termina devagar (padrão);
- linear - define uma animação com a mesma velocidade do começo ao fim;
- ease-in - define uma animação com um início lento;
- ease-out - define uma animação com uma extremidade lenta;
- ease-in-out - define uma animação com um início e fim lentos;
- cubic-bezier(n,n,n,n) - permite que você defina seus próprios valores em uma função cúbico-bezier.

Vamos ver sua aplicação em nosso exemplo, utilizando a propriedade animation-timing-function: ease-in:

```

1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.      <head>
4.          <meta charset="utf-8"/>
5.          <title>Animações</title>
6.          <style>
7.              /* Código da animação */
8.                  @keyframes teste { /* criação do @keyframes com nome teste */
9.                      0% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* posição inicial */
10.                     25% {background-color:#F00; left:400px; top:0px;width:200px;height:100px;}
11.                     50% {background-color:#0F0; left:400px; top:400px;width:100px;height:50px;}
12.                     75% {background-color:#F0F; left:0px; top:400px;width:200px;height:100px;}
13.                     100% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* volta a posição inicial */
14.                 }
15.
16.             /* Elemento que vamos aplicar a animação */
17.             div {
18.                 width: 300px;
19.                 height: 200px;
20.                 position: relative; /* definição da posição do elemento */
21.                 background-color: #00F;
22.                 animation-name: teste; /* vinculação ao @keyframes pelo nome */
23.                 animation-duration: 10s; /* tempo da animação */
24.                 animation-delay: 5s; /* tempo de espera para iniciar a animação
*
25.                 animation-iteration-count: 4; /* define o número de vezes que a animação ocorrerá. Caso deseje que ela funcione infinitamente utilize o valor infinito */
26.                 animation-direction: alternate; /* define o sentido da animação. Pode ser reverse ou alternate */
27.                 animation-timing-function: ease-in; /*define uma animação com um início lento*/
28.             }
29.
30.         </style>
31.     </head>
32.     <body>
33.
34.         <div></div>
35.
36.     </body>
37. </html>

```



## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).



## VAMOS TREINAR?

Treine a utilização da propriedade animation-timing-function [clicando aqui.](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_speed)  
[https://www.w3schools.com/css/tryit.asp?  
filename=trycss3\\_animation\\_speed](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_speed)

## Abreviando a propriedade animation

Nós podemos abreviar a linha de comando da propriedade animation da seguinte forma:

1. `animation: animation-name animation-duration animation-timing-function animation-delay animation-iteration-count animation-direction;`

O código de nossa animação, seguindo esta ordem seria:

```

1. div {
2.     width: 300px;
3.     height: 200px;
4.     position: relative; /* definição da posição do elemento */
5.     background-color: #00F;
6.     animation-name: teste; /* vinculação ao @keyframes pelo nome */
7.     animation-duration: 10s; /* tempo da animação */
8.     animation-timing-function: ease-in; /* define uma animação com um
   início lento*/
9.     animation-delay: 5s; /* tempo de espera para iniciar a animação
   */
10.    animation-iteration-count: 4; /* define o número de vezes que a a
   nimação ocorrerá. Caso deseje que ela funcione infinitamente utilize o valor infinito */
11.    animation-direction: alternate; /* define o sentido da animação.
   Pode ser reverse ou alternate */
12. }
```

Nossa linha de comando abreviada para a propriedade animation seria então:

```

1. div {
2.     width: 300px;
3.     height: 200px;
4.     position: relative; /* definição da posição do elemento */
5.     background-color: #00F;
6.     animation: teste 10s ease-in 5s 4 alternate;
7. }
```

O código completo de nossa página HTML com a linha de comando da propriedade animation resumida seria.

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Animações</title>
6.     <style>
7.       /* Código da animação */
8.       @keyframes teste { /* criação do @keyframes com nome teste */
9.         0% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* posição inicial */
10.        25% {background-color:#F00; left:400px; top:0px;width:200px;height:100px;}
11.        50% {background-color:#0F0; left:400px; top:400px;width:100px;height:50px;}
12.        75% {background-color:#F0F; left:0px; top:400px;width:200px;height:100px;}
13.        100% {background-color:#FF0; left:0px; top:0px;width:300px;height:200px;}/* volta a posição inicial */
14.      }
15.
16.      /* Elemento que vamos aplicar a animação */
17.      div {
18.        width: 300px;
19.        height: 200px;
20.        position: relative; /* definição da posição do elemento */
21.        background-color: #00F;
22.        animation: teste 10s ease-in 5s 4 alternate; /* linha de comando
da propriedade animation abreviada */
23.      }
24.
25.    </style>
26.  </head>
27.  <body>
28.
29.    <div></div>
30.
31.  </body>
32. </html>
```



## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

## VAMOS TREINAR?

Treine a utilização abreviada da propriedade animation [clicando aqui](#) ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation5](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation5)).

## Criando transições

Quando criamos transições estamos na verdade mudando os valores da propriedade de forma suave durante um período de tempo.



Para criar transições utilizando o HTML e o CSS precisamos lembrar de especificar duas coisas, sendo elas:

- A propriedade em CSS que desejamos adicionar um efeito;
- A duração do efeito.

Definidas estas duas informações podemos então, iniciar a criação de nossas transições.

## A propriedade transition

Para a criação de transições utilizamos a propriedade transition que definirá o que queremos que seja alterado em nosso elemento.

Vamos ver um exemplo abaixo em que um elemento div muda de tamanho quando o mouse for colocado sobre ele, obedecendo uma duração de 4 segundos (observe a linha de comando da propriedade transition no CSS da div):

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Transições</title>
6.     <style>
7.
8.       /* Elemento que vamos aplicar a transição */
9.       div {
10.         width: 300px;
11.         height: 200px;
12.         background-color: #00F;
13.         transition: width 4s; /* propriedade transition informando que será
14.                     alterada a largura no tempo de 4 segundos*/
15.       }
16.       /* elemento div com o estado hover - mouse sobre - que ativará a transição
17.                     até 600px de largura*/
18.       div:hover {
19.         width: 600px;
20.       }
21.     </style>
22.   </head>
23.   <body>
24.
25.     <div></div>
26.   </body>
27.
28. </html>
```

Observe que quando tiramos o mouse do elemento div, ela volta a seu tamanho original.

## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).



## VAMOS TREINAR?

Treine a utilização da propriedade transition [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition1).  
([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transition1](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition1))

## IMPORTANTE:

Observe que em todos os exemplos disponibilizados nos exercícios do "Vamos treinar?" há a inserção dos mesmos comandos de transição antecedidos por -webkit-. Isto ocorre para que as transições sejam corretamente visualizadas no navegador Safari , ok?

## Criando transições com outros valores de propriedade

Nós podemos criar outras transições alterando os valores das propriedades do elemento.

Vamos ver um exemplo com alteração da largura e da altura de um elemento div. Observe que, neste exemplo, para que a transição ocorra você deve clicar na div pois estamos utilizando o estado active, ok?

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Transições</title>
6.     <style>
7.
8.       /* Elemento que vamos aplicar a transição */
9.       div {
10.         width: 300px;
11.         height: 200px;
12.         background-color: #00F;
13.         transition: width 2s, height 4s; /* propriedade transition informando que será alterada a largura e altura com tempos diferentes*/
14.       }
15.       /* elemento div com o estado active - clicado - que ativará a transição até 600px de largura e 300px de altura*/
16.       div:active
17.     {
18.       width: 600px;
19.       height: 300px;
20.     }
21.
22.     </style>
23.   </head>
24.   <body>
25.
26.     <div></div>
27.
28.   </body>
29. </html>
```



## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](#) (<https://codepen.io/pen/?editors=1100>).

## VAMOS TREINAR?

Treine esta transformação de largura e de altura de um elemento [clicando aqui](#). ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transition2](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition2))

## DICA LEGAL:

Altere o elemento para treinar as transições! Veja o exemplo abaixo com uma transição aplicada a uma imagem (img). Para funcionar, salve uma imagem na sua máquina no mesmo local em que salvou o arquivo .html e substitua pelo exemplo.jpg, ok?



```

1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.      <head>
4.          <meta charset="utf-8"/>
5.          <title>Transições</title>
6.          <style>
7.
8.              /* Elemento que vamos aplicar a transição */
9.              img {
10.                  width: 300px;
11.                  height: 200px;
12.                  transition: width 2s, height 4s; /* propriedade transition informando que será alterada a largura e altura com tempos diferentes*/
13.              }
14.              /* elemento img com o estado hover - mouse sobre - que ativará a transição até 600px de largura e 300px de altura*/
15.              img:hover
16.              {
17.                  width: 600px;
18.                  height: 300px;
19.              }
20.
21.          </style>
22.      </head>
23.      <body>
24.
25.          
26.
27.      </body>
28.  </html>

```

## Definindo diferentes tempos de execução para a transição

Podemos definir uma "curva" de tempo para a transição, ou seja, definir como ela irá se comportar durante o tempo que definimos com a propriedade transition.

Para isto utilizamos a propriedade transition-timing-function. Segundo o W3schools (ONLINE) esta propriedade pode receber os seguintes valores:

- ease - define um efeito de transição com um início lento, então rápido, e depois termina devagar (padrão);
- linear - define um efeito de transição com a mesma velocidade do começo ao fim;

- ease-in - define um efeito de transição com um início lento;
- ease-out - define um efeito de transição com um fim lento;
- ease-in-out - define um efeito de transição com um início e fim lentos;
- cubic-bezier(n,n,n,n) - permite que você defina seus próprios valores em uma função cúbico-bezier.

Vamos ver sua aplicação em nosso exemplo, utilizando a propriedade transition-timing-function: ease-out:

```

1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.      <head>
4.          <meta charset="utf-8"/>
5.          <title>Transições</title>
6.          <style>
7.
8.              /* Elemento que vamos aplicar a transição */
9.              div {
10.                  width: 300px;
11.                  height: 200px;
12.                  background-color: #00F;
13.                  transition: width 2s, height 4s; /* propriedade transition informando que será alterada a largura e altura com tempos diferentes*/
14.                  transition-timing-function: ease-out; /*controla a forma como a transição funcionará, neste caso com um fim mais lento*/
15.              }
16.              /* elemento div com o estado houve - mouse sobre - que ativará a transição até 600px de largura e 300px de altura*/
17.              div:hover {
18.                  width: 600px;
19.                  height: 300px;
20.              }
21.
22.          </style>
23.      </head>
24.      <body>
25.
26.
27.      <div></div>
28.
29.  </body>
30. </html>

```



## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código e altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen clicando aqui (<https://codepen.io/pen/?editors=1100>).

## VAMOS TREINAR?

Treine a propriedade transition-timing-function clicando aqui ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transition\\_speed](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition_speed)).



## Atrasando o início da transição

Nós podemos atrasar o início da transição utilizando a propriedade transition-delay. Esta propriedade atrasa o inicio e o fim da transição.

Veja um exemplo de aplicação abaixo:

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Transições</title>
6.     <style>
7.
8.       /* Elemento que vamos aplicar a transição */
9.       div {
10.         width: 300px;
11.         height: 200px;
12.         background-color: #00F;
13.         transition: width 2s, height 4s; /* propriedade transition informando que será alterada a largura e altura com tempos diferentes*/
14.         transition-timing-function: ease-out; /*controla a forma como a transição funcionará, neste caso com um fim mais lento*/
15.         transition-delay: 3s; /*atrasa o funcionamento da transição em 3 segundos, no inicio e no fim*/
16.       }
17.       /* elemento div com o estado houve - mouse sobre - que ativará a transição até 600px de largura e 300px de altura*/
18.       div:hover {
19.         width: 600px;
20.         height: 300px;
21.       }
22.
23.
24.     </style>
25.   </head>
26.   <body>
27.
28.     <div></div>
29.
30.   </body>
31. </html>
```

## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).



## VAMOS TREINAR?

Treine a utilização da propriedade transition-delay [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition_delay) ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transition\\_delay](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition_delay)).

# Criando transformações

Nós podemos criar transformações em um elemento HTML utilizando a propriedade transform.

## A propriedade transform

A propriedade transform aplica uma transformação 2D ou 3D a um elemento. Com a propriedade transform podemos girar, dimensionar, mover, inclinar e fazer outras transformações.

Para fazer estas transformações utilizamos os seguintes métodos:

- **translate()** - utilizado para mover um determinado elemento HTML da sua posição atual (de acordo com os parâmetros dados para o eixo X e o eixo Y). Veja um exemplo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_translate). ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transform\\_translate](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_translate))
- **rotate()** - utilizado para rotacionar um determinado elemento HTML no sentido horário ou anti-horário de acordo com um determinado grau. Veja um exemplo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_rotate). ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transform\\_rotate](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_rotate))
- **scale()** - utilizado para aumentar ou diminuir o tamanho de um elemento HTML. Veja um exemplo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_scale). ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transform\\_scale](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_scale))

**filename=trycss3\_transform\_scale)**

- **skewX()** - utilizado para distorcer um elemento ao longo do eixo X, de acordo com um ângulo informado. Vejo um exemplo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_skewx) ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transform\\_skewx](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_skewx)).
- **skewY()** - utilizado para distorcer um elemento ao longo do eixo Y, de acordo com um ângulo informado. Vejo um exemplo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_skewy) ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transform\\_skewy](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_skewy)).
- **skew()** - utilizado para distorcer um elemento ao longo do eixo X e Y, de acordo com um ângulo informado. Vejo um exemplo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_skew) ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transform\\_skew](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_skew)).
- **matrix ()** - utilizado para unificar todos os métodos de transformação 2D em um único método. Veja um exemplo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_matrix1) ([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transform\\_matrix1](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transform_matrix1)).



### As transformações 2D

Na tabela abaixo temos as explicações do site do MSDN Microsoft (ONLINE) com os valores aplicáveis para a propriedade transform para 2D :

| Função  | Descrição  |
|---|--|
| <b>matrix(<i>a, b, c, d, e, f</i>)</b><br>( <a href="https://msdn.microsoft.com/pt-br/library/jj200270(v=vs.85).aspx">https://msdn.microsoft.com/pt-br/library/jj200270(v=vs.85).aspx</a> ) | Especifica uma transformação 2D sob a forma de uma matriz de transformação de seis valores.  |
| <b>rotate(<i>angle</i>)</b><br>( <a href="https://msdn.microsoft.com/pt-br/library/jj200276(v=vs.85).aspx">https://msdn.microsoft.com/pt-br/library/jj200276(v=vs.85).aspx</a> )            | Especifica uma rotação 2D pelo ângulo especificado no parâmetro sobre a origem do elemento.  |
| <b>scale(<i>sx,sy</i>)</b><br>( <a href="https://msdn.microsoft.com/pt-br/library/jj200281(v=vs.85).aspx">https://msdn.microsoft.com/pt-br/library/jj200281(v=vs.85).aspx</a> )             | Especifica uma operação de escala 2D pelo vetor escalar [ <i>sx,sy</i> ] que é descrito por dois parâmetros.   |
| <b>scaleX(<i>sx</i>)</b><br>( <a href="https://msdn.microsoft.com/pt-br/library/jj200278(v=vs.85).aspx">https://msdn.microsoft.com/pt-br/library/jj200278(v=vs.85).aspx</a> )               | Especifica uma operação de escala usando o vetor escalar [ <i>sx,1</i> ], onde <i>sx</i> é dado como o parâmetro.  |
| <b>scaleY(<i>sy</i>)</b><br>( <a href="https://msdn.microsoft.com/pt-br/library/jj200279(v=vs.85).aspx">https://msdn.microsoft.com/pt-br/library/jj200279(v=vs.85).aspx</a> )               | Especifica uma operação de escala usando o vetor escalar [ <i>1,sy</i> ], onde <i>sy</i> é dado como o parâmetro.  |
| <b>skew(<i>angleX,angleY</i>)</b><br>( <a href="https://msdn.microsoft.com/pt-br/library/jj200284(v=vs.85).aspx">https://msdn.microsoft.com/pt-br/library/jj200284(v=vs.85).aspx</a> )      | Especifica uma transformação de inclinação ao longo dos eixos <i>x</i> e <i>y</i> . O primeiro parâmetro de ângulo especifica a inclinação no eixo <i>x</i> . O segundo parâmetro de ângulo especifica a inclinação no eixo <i>y</i> . |

| Função   | Descrição   |
|--|---|
| <b>skewX(angle)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200282(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200282(v=vs.85).aspx)</a>     | Especifica uma transformação de inclinação ao longo do eixo x pelo ângulo dado.   |
| <b>skewY(angle)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200283(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200283(v=vs.85).aspx)</a>     | Especifica uma transformação de inclinação ao longo do eixo y pelo ângulo dado.   |
| <b>translate(tx,ty)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200290(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200290(v=vs.85).aspx)</a> | Especifica uma conversão 2D pelo vetor [tx,ty], onde tx é o primeiro parâmetro de valor de conversão e ty é o segundo parâmetro opcional do valor de conversão. |
| <b>translateX(tx)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200287(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200287(v=vs.85).aspx)</a>   | Especifica uma conversão por quantidade dada na direção x.  |
| <b>translateY(ty)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200288(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200288(v=vs.85).aspx)</a>   | Especifica uma conversão por quantidade dada na direção y.  |

#### Funções de transformação 2D

**Fonte:** MSDN Microsoft. Dê vida à sua página com transformações CSS.

Disponível em <[https://msdn.microsoft.com/pt-br/library/jj665791\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/jj665791(v=vs.85).aspx)> ([https://msdn.microsoft.com/pt-br/library/jj665791\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/jj665791(v=vs.85).aspx)). Acesso em 04 jun. 2017.



### PARA SABER MAIS:

Clique nos links disponíveis nas funções para ser direcionado a explicações no site da MSDN Microsoft (em inglês).

### As transformações 3D

Na tabela abaixo temos as explicações do site do MSDN Microsoft (ONLINE) com os valores aplicáveis para a propriedade transform para 3D :

| Função   | Descrição                                     |
|--|---|
| <b>rotate3d(x, y, z, angle)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200272(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200272(v=vs.85).aspx)</a> | Especifica uma rotação 3D em sentido horário. |

|  |   |
|--|---|
| <b>rotateX(angle)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200273(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200273(v=vs.85).aspx)</a>   | Especifica uma rotação no sentido horário pelo ângulo fornecido sobre o eixo x.   |
| <b>rotateY(angle)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200283(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200283(v=vs.85).aspx)</a>   | Especifica uma rotação no sentido horário pelo ângulo fornecido sobre o eixo y.   |
| <b>rotateZ(angle)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200290(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200290(v=vs.85).aspx)</a>   | Especifica uma rotação no sentido horário pelo ângulo fornecido sobre o eixo z.   |
| <b>scale3d(sx,sy,sz)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200277(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200277(v=vs.85).aspx)</a>  | Especifica uma operação de escala 3D pelo vetor escalar [sx,sy,sz] descrito pelos três parâmetros.  |
| <b>scaleZ(sz)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200280(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200280(v=vs.85).aspx)</a>   | Especifica uma operação de escala usando o vetor escalar [1,1,sz], onde sz é dado como o parâmetro.   |
| <b>translate3d(tx, ty, tz)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200286(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200286(v=vs.85).aspx)</a>                                  | Especifica uma conversão 3D pelo vetor [tx,ty,tz], onde tx, ty e tz são o primeiro, o segundo e o terceiro parâmetros de valor de conversão, respectivamente. |
| <b>translateZ(tz)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200289(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200289(v=vs.85).aspx)</a>   | Especifica uma conversão por quantidade dada na direção z.  |
| <b>matrix3d(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p)</b><br><a href="https://msdn.microsoft.com/pt-br/library/jj200269(v=vs.85).aspx">(https://msdn.microsoft.com/pt-br/library/jj200269(v=vs.85).aspx)</a> | Especifica uma transformação 3D como uma matriz homogênea $4 \times 4$ de 16 valores na ordem principal das colunas.  |



**Fonte:** MSDN Microsoft. Dê vida à sua página com transformações CSS.

Disponível em <[https://msdn.microsoft.com/pt-br/library/jj665791\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/jj665791(v=vs.85).aspx)> ([https://msdn.microsoft.com/pt-br/library/jj665791\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/jj665791(v=vs.85).aspx)). Acesso em 04 jun. 2017.

## PARA SABER MAIS:

Clique nos links disponíveis nas funções para ser direcionado a explicações no site da MSDN Microsoft (em inglês).

Veja abaixo os exemplos de utilização de cada um destes valores da propriedade transform:

- **rotate** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_rotate&preval=none](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_rotate&preval=none));

- **matrix;** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_matrix](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_matrix))
- **translate;** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_translate](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_translate))
- **translateX(x);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_translatex](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_translatex));
- **translateY(y);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_translatey](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_translatey))
- **scale(x,y);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_scale](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_scale))
- **scaleX(x);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_scalex](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_scalex))
- **scaleY(y);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_scaley](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_scaley))
- **rotate(*angle*);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_rotate](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_rotate))
- **rotateX(*angle*);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_rotatex](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_rotatex))
- **rotateY(*angle*);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_rotatey](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_rotatey))
- **rotateZ(*angle*);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_rotatez](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_rotatez))
- **skew(*x-angle,y-angle*);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_skew](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_skew))
- **skewX(*angle*);** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_skewx](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_skewx))
- **skewY(*angle*).** ([https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_transform\\_skewy](https://www.w3schools.com/cssref/playit.asp?filename=playcss_transform_skewy))



## Criando transições com transformações

Para criar uma transição que ocorra ao mesmo tempo com uma transformação utilizamos:

- o valor transform na propriedade transition;
- a propriedade transform no elemento em que ocorrerá a transição com a transformação.

Veja abaixo um exemplo da aplicação de uma transição com transformação.

```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Transições</title>
6.     <style>
7.
8.       /* Elemento que vamos aplicar a transição */
9.       div {
10.         width: 300px;
11.         height: 200px;
12.         background-color: #00F;
13.         transition: width 2s, height 4s, transform 2s; /* propriedade transition informando que será alterada a largura e altura com tempos diferentes e também com uma transformação*/
14.         transition-timing-function: ease-out; /*controla a forma como a transição funcionará, neste caso com um fim mais lento*/
15.         transition-delay: 3s; /*atrasa o funcionamento da transição em 3 segundos, no inicio e no fim*/
16.       }
17.       /* elemento div com o estado houve - mouse sobre - que ativará a transição até 600px de largura e 300px de altura e a transformação rotate*/
18.       div:hover {
19.         width: 600px;
20.         height: 300px;
21.         transform: rotate(180deg); /*propriedade transform que determina o tipo de transformação, neste caso rotate, o 180deg seria como um giro com ângulo de 180 graus*/
22.       }
23.     }
24.
25.   </style>
26. </head>
27. <body>
28.
29.   <div></div>
30.
31. </body>
32. </html>
```



## VAMOS FAZER UM EXERCÍCIO?

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Altere o código transform: rotate(180deg) trocando os métodos da propriedade transform;
5. Salve o arquivo com a extensão .html;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

## VAMOS TREINAR?

Treine a realização de transições com transformações [clicando aqui.](#)  
([https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_transition\\_transform](https://www.w3schools.com/css/tryit.asp?filename=trycss3_transition_transform))

Dica: Altere o código transform: rotate(180deg) trocando os métodos da propriedade transform.



## Ferramentas e exemplos

Criar animações, transições e transformações utilizando HTML5 e CSS3 nem sempre é uma tarefa fácil, mas há ótimas ferramentas de apoio que servem para facilitar este trabalho disponíveis na internet.

Abaixo seguem algumas sugestões:

### Ferramentas para Animação em HTML5

- <https://www.hippani.com/> (<https://www.hippani.com/>)
- <https://www.google.com/webdesigner/>  
(<https://www.google.com/webdesigner/>)
- <https://www.animatron.com/> (<https://www.animatron.com/>)
- <http://html5maker.com/> (<http://html5maker.com/>)
- <http://www.createjs.com/#!/TweenJS>  
(<http://www.createjs.com/#!/TweenJS>)
- <https://www.aquafadas.com//motioncomposer/?lang=en>  
(<https://www.aquafadas.com//motioncomposer/?lang=en>)
- <http://tumult.com/hype/> (<http://tumult.com/hype/>)
- <https://greensock.com/gsap> (<https://greensock.com/gsap>)
- <https://www.mugeda.com/index.php> (<https://www.mugeda.com/index.php>)
- <http://radiapp.com/> (<http://radiapp.com/>)
- <http://bly.sk/#about> (<http://bly.sk/#about>)
- <http://www.nodefire.com/> (<http://www.nodefire.com/>)
- <https://modernizr.com/> (<https://modernizr.com/>)

### Exemplos:

- <https://ianlunn.github.io/Hover/> (<https://ianlunn.github.io/Hover/>)
- <http://all-animation.github.io/> (<http://all-animation.github.io/>)
- <http://leaverou.github.io/animatable/#text-indent>  
(<http://leaverou.github.io/animatable/#text-indent>)

- <https://1stwebdesigner.com/css-effects/> (<https://1stwebdesigner.com/css-effects/>)

Edite os códigos diretamente no codepen

- <https://dcrazed.com/css3-animation-examples/> (<https://dcrazed.com/css3-animation-examples/>)
- <https://webdesign.tutsplus.com/pt/articles/15-inspiring-examples-of-css-animation-on-codepen--cms-23937> (<https://webdesign.tutsplus.com/pt/articles/15-inspiring-examples-of-css-animation-on-codepen--cms-23937>)



## O QUE VOCÊ PRECISA TER ENTENDIDO NESTE TÓPICO

- O que são e como fazer animações utilizando HTML5 e CSS3;
- O que são e como fazer transições utilizando HTML5 e CSS3;
- O que são e como fazer transformações utilizando HTML5 e CSS3.

## Quiz

Exercício Final

Animações e Transições

INICIAR ➤

## Referências

DEVELOPER MOZILLA. animation. Disponível em <<https://developer.mozilla.org/en-US/docs/Web/CSS/animation?v=control>> Acesso em 04 jun. 2017.

DEVELOPER MOZILLA. @keyframes. Disponível em <<https://developer.mozilla.org/en-US/docs/Web/CSS/@keyframes>> Acesso em 04 jun. 2017.

DEVELOPER MOZILLA. transition. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/CSS/transition>> Acesso em 04 jun. 2017.

DEVELOPER MOZILLA. transform. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/CSS/transform>> Acesso em 04 jun. 2017.

MSDN Microsoft. HTML5 - Adicionar personalidade com transições e animações de CSS3. Disponível em <<https://msdn.microsoft.com/pt-br/library/jj944012.aspx>> (https://msdn.microsoft.com/pt-br/library/jj944012.aspx) Acesso em 04 jun. 2017.

MSDN Microsoft. Dê vida à sua página com transformações CSS. Disponível em <[https://msdn.microsoft.com/pt-br/library/jj665791\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/jj665791(v=vs.85).aspx)> (https://msdn.microsoft.com/pt-br/library/jj665791(v=vs.85).aspx) Acesso em 04 jun. 2017.

SILVA, Murício Samy. *HTML5: A linguagem de marcação que revolucionou a web.* 1. ed. São Paulo: Novatec Editora, 2011.

TABLELESS. All Animation CSS3 ↗ Criar animações CSS3 nunca foi tão fácil. Disponível em <<https://tableless.com.br/animation-css3-criar-animacoes-css3-nunca-foi-tao-facil>> (https://tableless.com.br/animation-css3-criar-animacoes-css3-nunca-foi-tao-facil) Acesso em 04 jun. 2017.

TERUEL, Evandro Carlos. *HTML5: Guia prático.* 1. ed. São Paulo: Erica, 2011.

W3SCHOOLS. CSS3 Animations. Disponível em <[https://www.w3schools.com/css/css3\\_animations.asp](https://www.w3schools.com/css/css3_animations.asp)> (https://www.w3schools.com/css/css3\_animations.asp) Acesso em 04 jun. 2017.

W3SCHOOLS. CSS3 Transitions. Disponível em <[https://www.w3schools.com/css/css3\\_transitions.asp](https://www.w3schools.com/css/css3_transitions.asp)> (https://www.w3schools.com/css/css3\_transitions.asp) Acesso em 04 jun. 2017.

W3SCHOOLS. CSS3 2D Transforms. Disponível em <[https://www.w3schools.com/css/css3\\_2dtransforms.asp](https://www.w3schools.com/css/css3_2dtransforms.asp)> (https://www.w3schools.com/css/css3\_2dtransforms.asp) Acesso em 04 jun. 2017.



Avalie este tópico



ANTERIOR  
Formulários

≡  
Índice

Biblioteca  
(<https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)  
Portal Uninove  
(<http://www.uninove.br>)  
Mapa do Site

® Todos os direitos reservados

Ajuda?  
PRÓXIMO:  
(<https://ava.uninove.br/seu/AVA/topico/topico.php>)  
Som eídosos=)

