

CSS

transition, animation

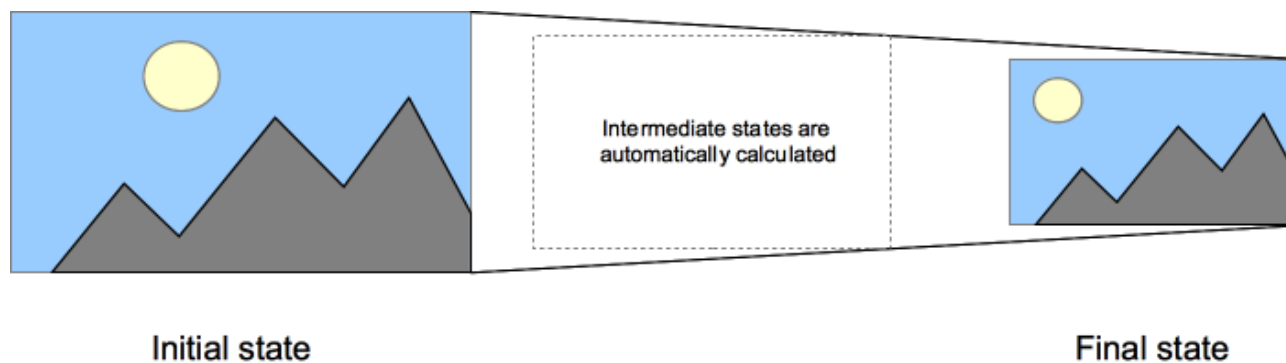




Transições CSS

Permitem **alterar os valores das propriedades CSS suavemente**, durante um determinado período.

Utilizada para dar alguns efeitos de animação sobre as páginas web.



Animações que envolvem transições entre dois estados são chamadas de ***transições implícitas***, como os estados entre o começo e o fim são implicitamente definidos pelo browser



Características

Animações CSS permitem:

- a) Definir **quais propriedades animar** (listando explicitamente);
- b) Determinar **quando a animação deve começar** (definindo o delay);
- c) Estabelecer a **duração da animação**; e
- d) Definir **como a transição ocorrerá** (definindo o tempo da função, linear ou rápido no início, lento no final).



Como usar CSS transitions?

Para criar uma transição, é necessário especificar duas coisas:

- A **propriedade CSS** que se deseja adicionar um efeito; e
- A **duração** do efeito.



Se a duração não for especificada, a transição não terá efeito, **já que o valor padrão é 0 (zero).**

```
div {  
  width: 150px;  
  height: 150px;  
  background: darkgreen;  
  transition: width 2s;  
}
```

Seleciona e estiliza uma <div>
quando o mouse passar sobre ela

```
div:hover {  
  width: 300px;  
}
```

tempo de duração

Propriedade CSS

[Exemplo.html](#)



Propriedade **transition-property**

Especifica a propriedade CSS para a qual o efeito de transição será aplicado.

O efeito de transição começa quando a propriedade CSS especificada for alterada.

Se a duração não for especificada, a transição não terá efeito, **já que o valor padrão é 0 (zero).**

```
div {  
    width: 100px;  
    height: 100px;  
    background: darkgreen;  
    transition-property: width;  
    transition-duration: 2s;  
}  
  
div:hover {  
    width: 300px;  
}
```

[Exemplo.html](#)



Propriedade **transition-duration**

Especifica a **duração** em que as transições devem ocorrer.

É possível especificar um único tempo de duração que se aplica a todas as propriedades durante a transição ou **vários valores para permitir que cada propriedade faça a transição em um período de tempo diferente.**

transition-duration: 2s

[Exemplo.html](#)



Propriedade **transition-delay**

Especifica **um atraso** (em segundos) para o início do efeito de transição.

```
div {  
  width: 100px;  
  height: 100px;  
  background: darkgreen;  
  transition-property: width;  
  transition-duration 3s;  
  transition-delay: 1s;  
}
```

```
div:hover {  
  width: 450px;  
}
```

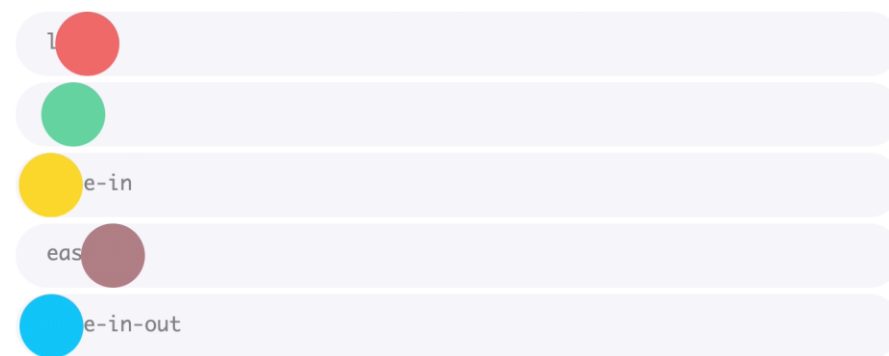
[Exemplo.html](#)

↑
----- Tempo de atraso (*delay*)

Propriedade **transition-timing-function**



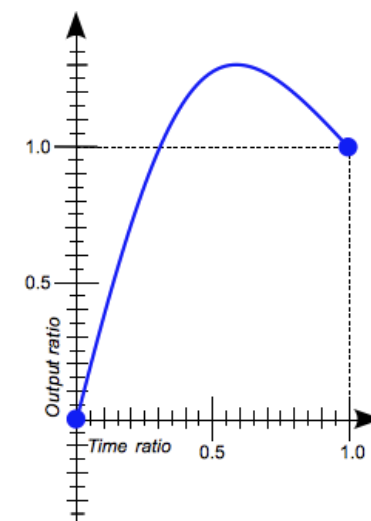
Especifica a **curva de velocidade** do efeito de transição.



[Exemplo.html](#)

Valores possíveis:

- **linear** - especifica um efeito de transição com a mesma velocidade do início ao fim;
- **ease** - especifica um efeito de transição com um início lento, depois rápido e, em seguida, termina lentamente (padrão);
- **easy-in** - especifica um efeito de transição com um início lento;
- **easy-out** - especifica um efeito de transição com um final lento;
- **easy-in-out** - especifica um efeito de transição com início e fim; e
- **cubic-bezier(n,n,n,n)** - permite definir seus próprios valores em uma função cúbica-bezier.





Propriedade abreviada **transition**

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

```
div {  
    width: 100px;  
    height: 100px;  
    background: darkgreen;  
    transition-property: width;  
    transition-duration: 2s;  
    transition-timing-function: linear;  
    transition-delay: 1s;  
}
```

As propriedades **transition**
podem ser especificadas uma a uma.

```
div {  
    width: 100px;  
    height: 100px;  
    background: darkgreen;  
    transition: width 2s linear 1s;  
}
```

Ou de forma abreviada.

[Exemplo.html](#)



Resumo das propriedades **transition**

Propriedade	Descrição
transition	Abreviação para definir as quatro propriedades de transição em uma única propriedade transition
transition-delay	Especifica um atraso (em segundos) para o efeito de transição
transition-duration	Especifica quantos segundos ou milissegundos um efeito de transição leva para ser concluído
transition-property	Especifica qual propriedade CSS sofrerá o efeito de transição
transition-timing-function	Define a velocidade da transição

CSS --- transform





transition + transform

Uma **CSS transform** permite mover, girar, dimensionar e inclinar elementos.

Aplica uma transformação 2D ou 3D a um elemento.

```
div {  
  width: 100px;  
  height: 100px;  
  background: darkgreen;  
  transition: width 2s, height 2s, transform 2s;  
}
```

Propriedade
transition

```
div:hover {  
  width: 300px;  
  height: 300px;  
  transform: rotate(180deg);  
}
```

[Exemplo.html](#)



Valores da propriedade **transform**

Propriedade	Descrição
none	Define que não deve haver transformação
matrix(<i>n,n,n,n,n,n</i>)	Define uma transformação 2D, usando uma matriz de seis valores
matrix3d(<i>n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n</i>)	Define uma transformação 3D, usando uma matriz 4x4 de 16 valores
translate(<i>x,y</i>)	Define uma translação 2D
translate3d(<i>x,y,z</i>)	Define uma translação 3D
translateX(<i>x</i>)	Define uma translação, usando apenas o valor para o eixo X
translateY(<i>y</i>)	Define uma translação, usando apenas o valor para o eixo Y
translateZ(<i>z</i>)	Define uma translação, usando apenas o valor para o eixo Z
scale(<i>x,y</i>)	Define uma transformação de escala 2D
scale3d(<i>x,y,z</i>)	Define uma transformação de escala 3D



Propriedade	Descrição
scaleX(x)	Define uma transformação de escala fornecendo um valor para o eixo X
scaleY(y)	Define uma transformação de escala fornecendo um valor para o eixo Y
scaleZ(z)	Define uma transformação de escala fornecendo um valor para o eixo Z
rotate(ângulo)	Define uma rotação 2D, o ângulo é especificado no parâmetro
rotate3d(x, y, z, ângulo)	Especifica uma rotação 3D.
rotateX(ângulo)	Define uma rotação 3D ao longo do eixo X
rotateY(ângulo)	Define uma rotação 3D ao longo do eixo Y
rotateZ(ângulo)	Define uma rotação 3D ao longo do eixo Z
skew(ângulo-x-, ângulo-y)	Define uma transformação de inclinação 2D ao longo dos eixos X e Y
skewX(ângulo)	Define uma transformação de inclinação 2D ao longo do eixo X
skewY(ângulo)	Define uma transformação de inclinação 2D ao longo do eixo y
perspective(n)	Define uma vista em perspectiva para um elemento transformado em 3D

CSS --- animation



CSS animations



Uma animação CSS permite que um **elemento mude gradualmente de um estilo para outro.**

Para usar animações é preciso **especificar quadros (keyframes) para a animação.**



Os **keyframes** **contêm os estilos** que o elemento terá em determinados momentos.



CSS Animations

Formado por dois componentes:

1. Um **estilo descrevendo a animação**; e
2. Um **conjunto de keyframes** que indicam o estado final e inicial do estilo da animação, bem como possíveis pontos intermediários ao longo do caminho.

Vantagens

- ✓ **Fácil utilização** para animações simples;
- ✓ As animações executam bem, mesmo **sobre moderada carga do sistema**;
- ✓ Deixar o navegador controlar a sequência de animação permite a **otimização da performance e eficiência**.



Regra @keyframe

Ao **especificar estilos em uma @keyframes**, a animação muda gradualmente do estilo atual para o novo estilo em determinados momentos.

Para fazer uma animação funcionar, é **necessário vincular a animação a um elemento**.

2) vincular a animação **exemplo** ao elemento <div>

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: darkgreen;  
  animation-name: exemplo;  
  animation-duration: 4s;  
}
```

3) a animação dura 4 segundos

1) Definir a regra: alterar a cor de fundo do elemento <div> de "verde" para "amarelo":

```
@keyframes exemplo {  
  from {  
    background-color: darkgreen;  
  }  
  to {  
    background-color: yellow;  
  }  
}
```

[Exemplo.html](#)



No exemplo anterior, o estilo é alterado usando as palavras-chave **from** e **to**, as quais representam, respectivamente, 0% (início) e 100% (animação completa).

É possível também definir porcentagens, para adicionar quantas alterações de estilo forem necessárias.

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: exemplo;  
  animation-duration: 4s;  
}
```

```
@keyframes exemplo {  
  0% {  
    background-color: red;  
  }  
  25% {  
    background-color: yellow;  
  }  
  50% {  
    background-color: blue;  
  }  
  100% {  
    background-color: green;  
  }  
}
```



Incrementando a animação

Altera a cor de fundo e a **posição do elemento <div>** quando a animação estiver 25% concluída, 50% concluída e novamente quando a animação estiver 100% concluída.

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: exemplo;  
  animation-duration: 4s;  
}
```

```
@keyframes exemplo {  
  0% {  
    background-color: red;  
    left: 0px;  
    top: 0px;  
  }  
  
  25% {  
    background-color: yellow;  
    left: 200px;  
    top: 0px;  
  }  
  
  50% {  
    background-color: blue;  
    left: 200px;  
    top: 200px;  
  }  
  
  75% {  
    background-color: green;  
    left: 0px;  
    top: 200px;  
  }  
  
  100% {  
    background-color: red;  
    left: 0px;  
    top: 0px;  
  }  
}
```

[Exemplo.html](#)



Propriedade animation-delay

Permite **especificar um atraso** para o início de uma animação.

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  position: relative;  
  animation-name: exemplo;  
  animation-duration: 4s;  
  animation-delay: 2s;  
}
```

```
@keyframes exemplo {  
  0%   {background-color:red; left:0px; top:0px;}  
  25%  {background-color:yellow; left:200px; top:0px;}  
  50%  {background-color:blue; left:200px; top:200px;}  
  75%  {background-color:green; left:0px; top:200px;}  
  100% {background-color:red; left:0px; top:0px;}  
}
```

[Exemplo.html](#)



Propriedade animation-delay

É possível especificar **valores negativos**.

[Exemplo.html](#)

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  position: relative;  
  animation-name: exemplo;  
  animation-duration: 4s;  
  animation-delay: -2s;  
}
```

```
@keyframes exemplo {  
  0%   {background-color:red; left:0px; top:0px;}  
  25%  {background-color:yellow; left:200px; top:0px;}  
  50%  {background-color:blue; left:200px; top:200px;}  
  75%  {background-color:green; left:0px; top:200px;}  
  100% {background-color:red; left:0px; top:0px;}  
}
```

Ao usar valores negativos, a animação começa como se já tivesse sido reproduzida por N segundos.



Propriedade **animation-iteration-count**

Define a **quantidade de vezes** que uma animação deve rodar.

[Exemplo.html](#)

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  position: relative;  
  animation-name: exemplo;  
  animation-duration: 4s;  
  animation-iteration-count: 5;  
}
```

```
@keyframes exemplo {  
  0%   {background-color:red; left:0px; top:0px;}  
  25%  {background-color:yellow; left:200px; top:0px;}  
  50%  {background-color:blue; left:200px; top:200px;}  
  75%  {background-color:green; left:0px; top:200px;}  
  100% {background-color:red; left:0px; top:0px;}  
}
```

Animação em loop infinito.

animation-iteration-count: infinite;



Propriedade `animation-direction`

Especifica **como uma animação deve ser reproduzida** (para frente, para trás ou em ciclos alternados).

[Exemplo.html](#)

Pode assumir os seguintes valores:

- **normal** - a animação é reproduzida para frente (padrão);
- **reverse** - a animação é reproduzida na direção contrária (para trás);
- **alternate** - a animação é reproduzida primeiro para a frente e depois para trás; e
- **alternate-reverse** - a animação é reproduzida primeiro para trás e depois para a frente.

```
div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
    animation-direction: reverse;  
}
```




Propriedade **animation-timing-function**

Especifica a **curva de velocidade** da animação

[Exemplo.html](#)

Pode assumir os seguintes valores:

- **ease** - animação tem um início lento, depois rápido e, termina lentamente (padrão);
- **linear** - animação com a mesma velocidade do início ao fim;
- **easy-in** - animação com um início lento;
- **easy-out** - animação com um final lento;
- **easy-in-out** - animação com início e fim lentos; e
- **cubic-bezier(n, n, n, n)** - permite definir seus próprios valores em uma função cúbica-bezier.

```
#div1 {  
  animation-timing-function: linear;  
}  
  
#div2 {  
  animation-timing-function: ease;  
}  
  
#div3 {  
  animation-timing-function: ease-in;  
}
```



Propriedade **animation-fill-mode**

As animações CSS não afetam um elemento **antes do primeiro keyframe ser reproduzido** ou **após o último keyframe ser reproduzido**.

A propriedade **animation-fill-mode** permite substituir esse comportamento.

Propriedade pode assumir os seguintes valores:

- **none** - valor padrão. A animação não aplicará nenhum estilo ao elemento antes ou depois de sua execução;
- **forward** - elemento manterá os valores de estilo definidos pelo último keyframe;
- **backward** - O elemento obterá os valores de estilo que são definidos pelo primeiro keyframe e os manterá durante o período de atraso da animação; e
- **both** - a animação seguirá as regras para frente e para trás, estendendo as propriedades da animação em ambas as direções.



Propriedade **animation**

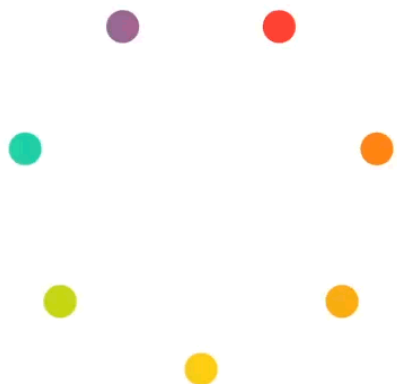
Usa seis das propriedades de animação, de forma abreviada.

```
div {  
    animation: example 5s linear 2s infinite alternate;  
}  
  
div {  
    animation-name: example;  
    animation-duration: 5s;  
    animation-timing-function: linear;  
    animation-delay: 2s;  
    animation-iteration-count: infinite;  
    animation-direction: alternate;  
}
```



Resumo das propriedades de **animation**

@keyframes	Especifica o código de animação
animation	Propriedade abreviada para definir todas as propriedades de animação
animation-delay	Especifica um atraso para o início de uma animação
animation-direction	Especifica se uma animação deve ser reproduzida para frente, para trás ou em ciclos alternados
animation-duration	Especifica quanto tempo uma animação deve levar para completar um ciclo
animation-fill-mode	Especifica um estilo para o elemento quando a animação não está sendo reproduzida (antes de começar, depois de terminar ou ambos)
animation-iteration-count	Especifica o número de vezes que uma animação deve ser reproduzida
animation-name	Especifica o nome da animação @keyframes
animation-play-state	Especifica se a animação está em execução ou pausada
animation-timing-function	Especifica a curva de velocidade da animação



CSS

transition, animation