

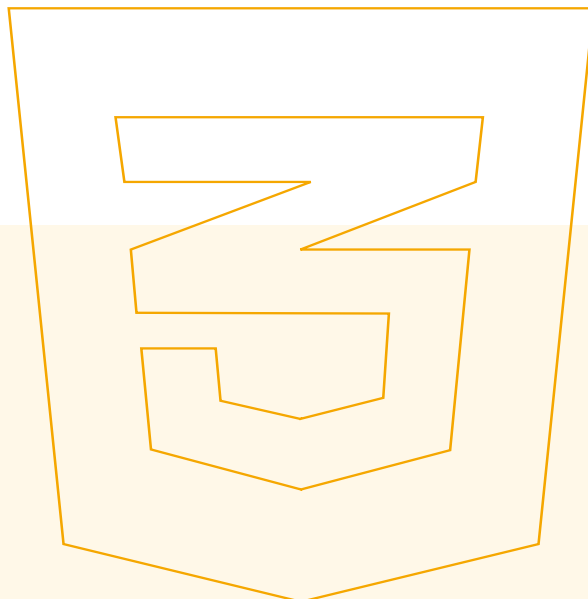


Segura esse estilo!

Animações em CSS

Animações em CSS são formas de **fazer elementos da página se moverem ou mudarem de aparência ao longo do tempo**. Elas ajudam a deixar a interface mais viva, interativa e agradável para o usuário. Ele define o que vai mudar, quanto tempo vai durar, como será o ritmo da mudança e se vai repetir ou parar no final. Neste ebook exploraremos mais essas animações e teremos exemplos de códigos em contextos reais, por isso sente-se, pegue uma pipoca que a animação já vai começar!

Só lembrando que os alguns códigos que serão apresentados, são códigos mais simples com o qual podemos trabalhar em cima, é apenas um exemplo de algo que pode ser ainda mais implementado.



Keyframes e o Controle Total da Animação

CAPÍTULO 1





Animando com keyframes

O que são @keyframes?

Enquanto as transições em CSS funcionam bem para mudanças simples e diretas, o @keyframes entra em cena quando você precisa de **mais controle sobre o processo da animação**. Com ele, você define etapas específicas (keyframes) de uma animação e pode alterar várias propriedades ao longo do tempo.

Pense como uma linha do tempo: você escolhe **o que acontece no começo, no meio e no fim da animação**.

Esse código, por exemplo, nos especifica que ao carregar a página, o título sobe com suavidade e aparece gradualmente.

```
Keyframe.CSS

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Animação de Título</title>
  <style>
    body {
      margin: 0;
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
      background-color: #f5f5f5;
      font-family: Arial, sans-serif;
    }

    @keyframes deslizante {
      from {
        opacity: 0;
        transform: translateY(30px);
      }
      to {
        opacity: 1;
        transform: translateY(0);
      }
    }

    .titulo {
      font-size: 3rem;
      color: #333;
      animation: deslizante 0.8s ease-out forwards;
    }
  </style>
</head>
<body>
  <h1 class="titulo">Bem-vindo!</h1>
</body>
</html>
```

Agora vamos analisar as partes mais importantes do nosso código, nós começamos com `class="titulo"` que é a nossa estrutura HTML, nela se aplica o estilo e a animação definidos no CSS ao elemento.

Temos também o estilo do "corpo":

Display: flex: ativa o Flexbox.

Justify-content: center: centraliza horizontalmente.

Align-items: center: centraliza verticalmente.

Height: 100vh: ocupa toda a altura da tela.

Resultado: o título fica perfeitamente centralizado.

Temos a nossa animação com o @Keyframes:

@Keyframes deslizante: define a animação chamada **deslizante**.

Opacity: 0 → 1: o título aparece suavemente (efeito de fade-in).

Transform: translateY(30px → 0): o título sobe 30px, criando o efeito de deslizamento.

E finalmente nós temos a nossa aplicação da animação

Animation: aplica a animação ao elemento.

Deslizante: nome da animação.

0.8s: duração da animação.

Ease-out: curva de aceleração (começa rápido e desacelera no final).

Forwards: mantém o estado final da animação (não volta ao início).

Esse tipo de efeito é ideal para títulos ou elementos que aparecem suavemente ao carregar a página, dando mais fluidez e impacto visual.

Transições Suaves Tornando Mudanças Naturais

CAPÍTULO 2



Transições com TRANSITION

As transições permitem criar mudanças suaves quando uma propriedade CSS é alterada. Muito usadas para interações como passar o mouse sobre botões, imagens ou links. Isso melhora a experiência do usuário, tornando as interações mais naturais e menos abruptas.

```
Transição suave.CSS

.botao {
  background-color: #3498db;
  transition: background-color 0.3s ease;
}

.botao:hover {
  background-color: #2ecc71;
}
```


Vamos analisar o passo a passo, explicando as partes mais importantes desse código, ele nos especifica que quando o usuário passa o mouse sobre o botão, a cor muda suavemente em 0.3 segundos.

Como funciona:

Transition: ativa a transição.

Background-color: é a propriedade que será animada.

0.3s: duração da transição.

Ease: define a velocidade (começa devagar, acelera e desacelera no final).

Você pode aplicar transições em várias propriedades como color, transform, opacity, height, entre outras.

Loop Infinito

Animações Contínuas

CAPÍTULO 3



Usando INFINITE para ciclos sem fim

O que é loop infinito em CSS?

É quando uma animação se repete **sem parar**, criando um efeito contínuo. Isso é feito com a propriedade **animation-iteration-count: infinite;**, que instrui o navegador a repetir a animação para sempre.

É Ideal para ícones de carregamento, elementos decorativos ou indicadores de processo (como ondas, objetos/pessoas piscando ou engrenagens girando).

Use-o sempre com moderação pois animações infinitas podem cansar ou distrair o usuário.

Esse código, por exemplo, cria um **ícone de carregamento (loader)** em forma de círculo giratório, muito comum em interfaces web para indicar que algo está sendo processado.

Loop infinito.CSS

CSS

```
<div class="loader"></div>

<style>
.loader {
  width: 40px;
  height: 40px;
  border: 5px solid #ccc;
  border-top-color: #3498db;
  border-radius: 50%;
  animation: girar 1s linear infinite;
}

@keyframes girar {
  to {
    transform: rotate(360deg);
  }
}
</style>
```

Agora verificamos as partes mais importantes desse código:

Na estrutura HTML o **Div.loader**: é o elemento visual que será animado como um círculo girando.

Estilo CSS

Loop infinito.CSS

```
.loader {  
  width: 40px;           /* Largura do círculo */  
  height: 40px;          /* Altura do círculo */  
  border: 5px solid #ccc; /* Borda cinza clara */  
  border-top-color: #3498db; /* Parte superior azul (dá o efeito de rotação) */  
  border-radius: 50%;     /* Torna a div um círculo */  
  animation: girar 1s linear infinite; /* Aplica a animação contínua */  
}
```


Animações com Hover Interações em Tempo Real

CAPÍTULO 4



Combinando várias transições

E por último mas não menos importante, você pode animar múltiplas propriedades ao mesmo tempo, como tamanho, cor, sombra e aplicando várias animações a um único elemento separando-as por vírgulas.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Card com Hover</title>
  <style>
    body {
      margin: 0;
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
      background-color: #f0f0f0;
      font-family: Arial, sans-serif;
    }

    .card {
      background: white;
      padding: 20px;
      border-radius: 12px;
      box-shadow: 0 4px 6px rgba(0,0,0,0.1);
      transition: transform 0.3s, box-shadow 0.3s;
      font-size: 1.2rem;
      cursor: pointer;
    }

    .card:hover {
      transform: scale(1.05);
      box-shadow: 0 8px 16px rgba(0,0,0,0.2);
    }
  </style>
</head>
<body>
  <div class="card">Produto</div>
</body>
</html>
```

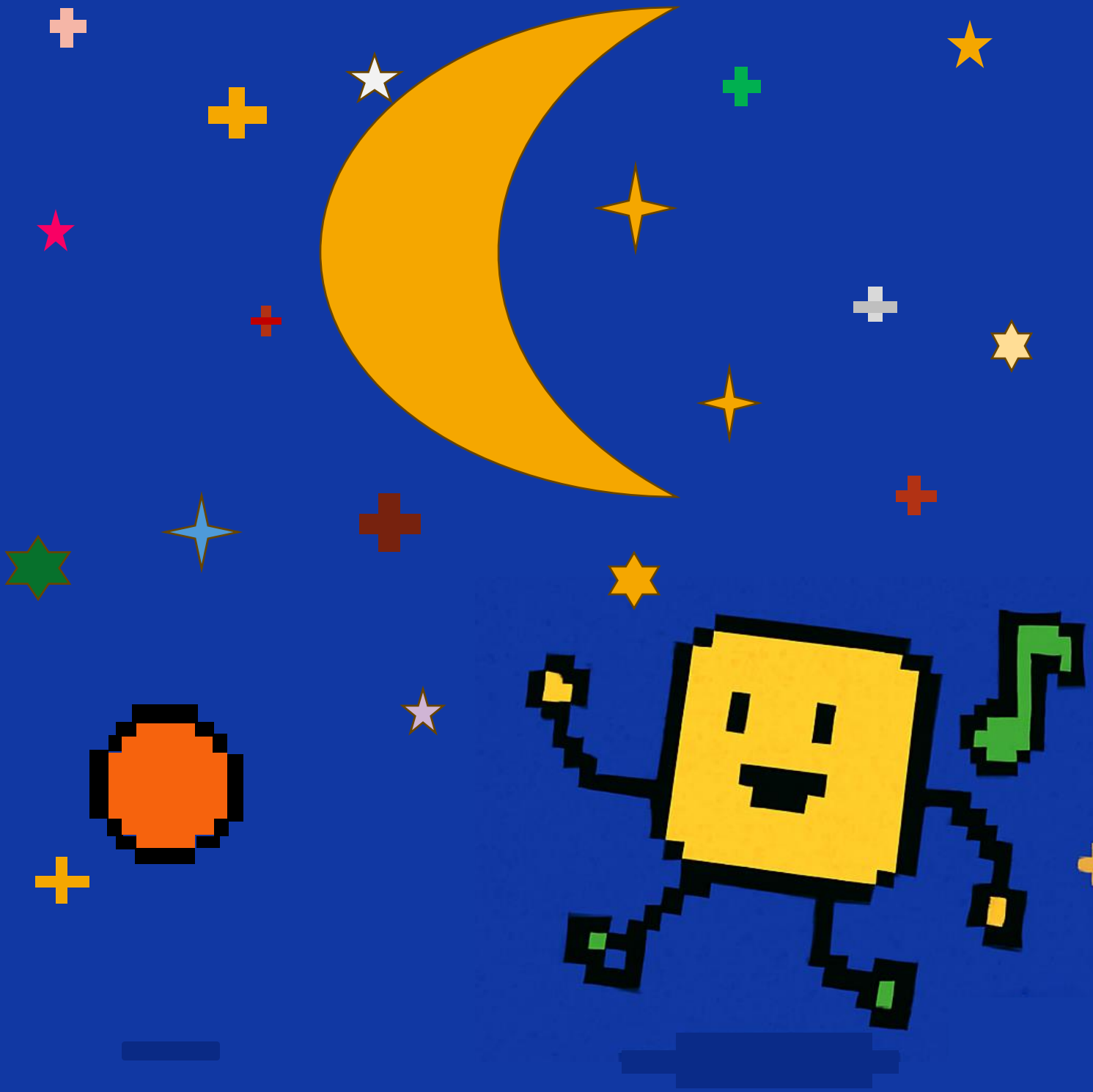
Esse código anterior, por exemplo, cria um **efeito de destaque ao passar o mouse sobre um cartão de produto**, muito usado em interfaces modernas para dar sensação de profundidade e interatividade. Vamos destacar as partes mais importantes:

Na estrutura de HTML o **Div.card**: representa um bloco visual (card) com conteúdo, como um produto, serviço ou informação.

Nós temos o estilo em CSS:

```
.card {  
  background: white;           /* Cor de fundo branca */  
  padding: 20px;              /* Espaçamento interno */  
  border-radius: 12px;        /* Cantos arredondados */  
  box-shadow: 0 4px 6px rgba(0,0,0,0.1); /* Sombra leve para dar profundidade */  
  transition: transform 0.3s, box-shadow 0.3s; /* Transições suaves */  
}
```

```
.card:hover {  
  transform: scale(1.05);      /* Aumenta levemente o tamanho */  
  box-shadow: 0 8px 16px rgba(0,0,0,0.2); /* Sombra mais intensa */  
}
```



Chegamos ao famoso
THE END!

Meus agradecimentos a quem chegou até aqui.

Este ebook teve ajuda de IA, mas foi diagramado por humano, isso inclui também alguns desenhos que foram feitos por mim e edits que foram necessários durante o processo de criação.

Acredito que não aja alucinações neste ebook, pois foi revisado e reescrito algumas vezes por mim, mas para desencargo, pode conter erros gerados por IA, esse ebook foi criado apenas para meios de aprendizagem.

Novamente agradeço pelo tempo de vocês :)

[Haxsandra/prompts-to-create-a-ebook](https://haxsandra.com/prompts-to-create-a-ebook)