



HELLO
WORLD



ANIMAÇÕES

COM CSS

INTRODUÇÃO ÀS ANIMAÇÕES COM CSS



- **Transições vs. Animações:** Explicar a diferença.
 - Transição: Anima uma mudança entre dois estados (por exemplo, ao passar o mouse em um botão).
 - Animação: Permite criar efeitos mais complexos, com várias etapas e loops.
- **Quando e por que usar animações:** melhorar a experiência do usuário, guiar o olhar para elementos importantes, e adicionar interatividade.



TRANSIÇÕES

Propriedades principais: transition-property, transition-duration, transition-timing-function, e transition-delay.

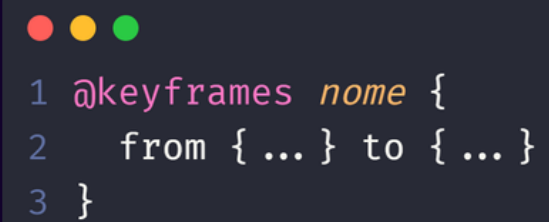
Exemplo básico: Criar um botão que muda de cor ao ser clicado.

```
● ● ●
1  .button {
2    background-color: #3498db;
3    color: white;
4    padding: 10px 20px;
5    transition: background-color 0.3s ease;
6  }
7
8  .button:hover {
9    background-color: #2ecc71;
10 }
```



ANIMAÇÕES COM @KEYFRAMES

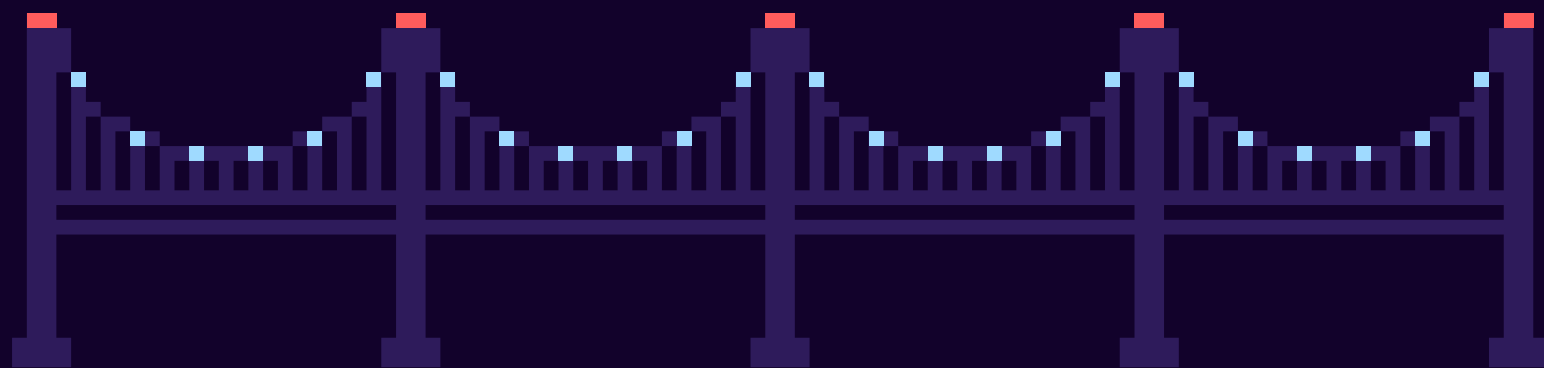
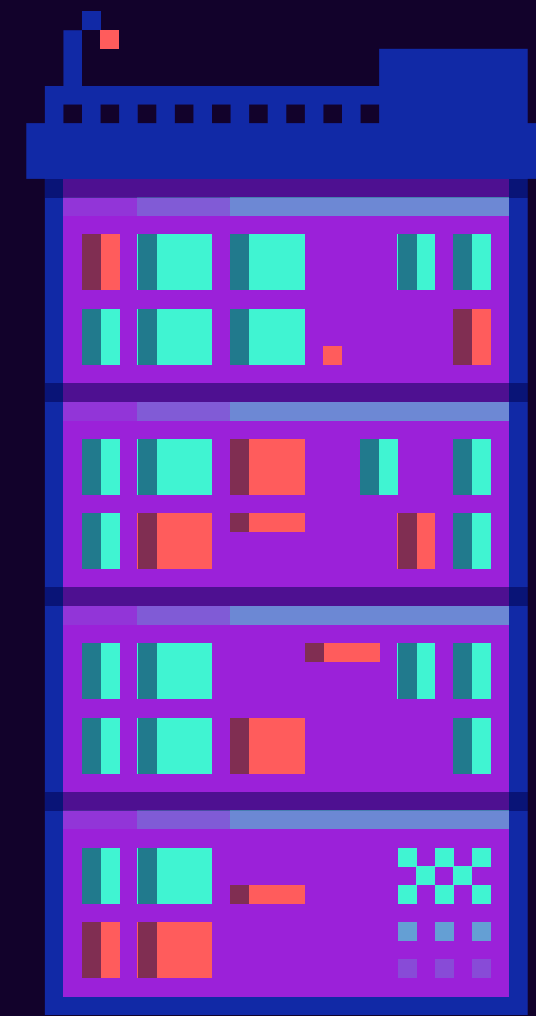
- **Estrutura Básica do @keyframes:**
 - @keyframes nome { from { ... } to { ... } } ou com porcentagens para definir múltiplas etapas.



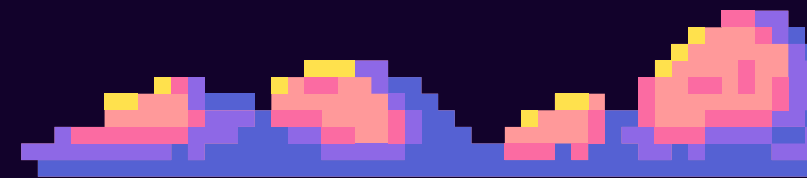
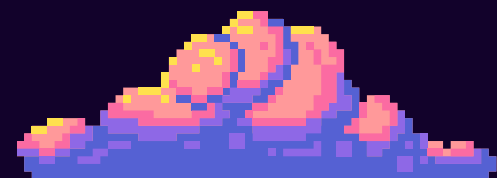
```
1 @keyframes nome {  
2   from { ... } to { ... }  
3 }
```

- **Exemplo básico:** Criar uma animação que faça uma caixa se mover para a direita.

[Saiba mais](#)



EXEMPLO BÁSICO



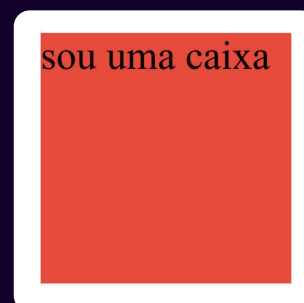
01

Exemplo básico: Criar uma animação que faça uma caixa se mover para a direita.

```
1 @keyframes moveRight {  
2   from { transform: translateX(0); }  
3   to { transform: translateX(100px); }  
4 }  
5  
6 .box {  
7   width: 100px;  
8   height: 100px;  
9   background-color: #3498db;  
10  animation: moveRight 2s ease-in-out;  
11 }
```

sou uma caixa

PROPRIEDADES DE ANIMAÇÃO



01

animation-name: Nome do @keyframes utilizado.

02

animation-duration: Duração da animação.

03

animation-timing-function: Curva de velocidade (ease, ease-in, linear, etc.).

04

animation-delay: Espera antes da animação começar.

05

animation-iteration-count: Quantas vezes repetir (número específico ou infinite).

06

animation-direction: Direção (normal, reverse, alternate).



```
1  @keyframes pulse {
2    0% { transform: scale(1); }
3    50% { transform: scale(1.1); }
4    100% { transform: scale(1); }
5  }
6
7  .pulse-box {
8    width: 100px;
9    height: 100px;
10   background-color: #e74c3c;
11   animation: pulse 1s infinite;
12 }
```

ANIMATION-TIMING-FUNCTION

- Define como a velocidade da animação muda ao longo do tempo, ou seja, a curva de aceleração e desaceleração.
- Controla se a animação será suave, acelerada, desacelerada ou constante.

01

linear: Velocidade constante.

02

ease: Começo lento, aceleração no meio e final lento.

03

ease-in: Começo lento, acelera ao longo da animação.

04

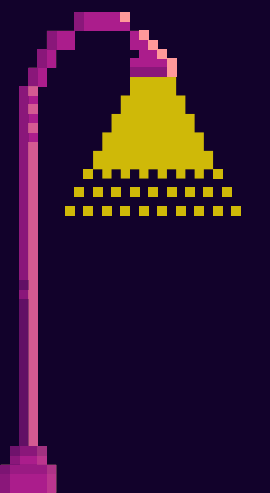
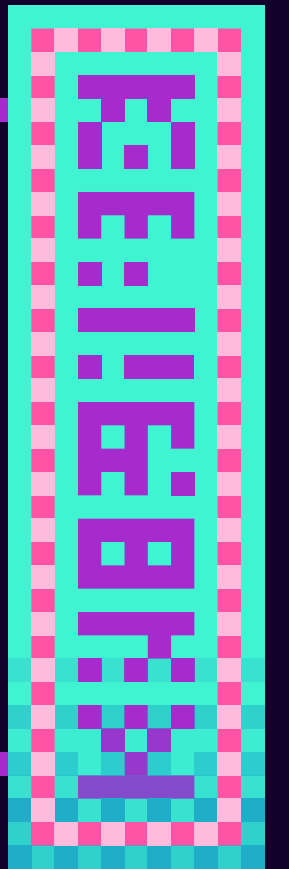
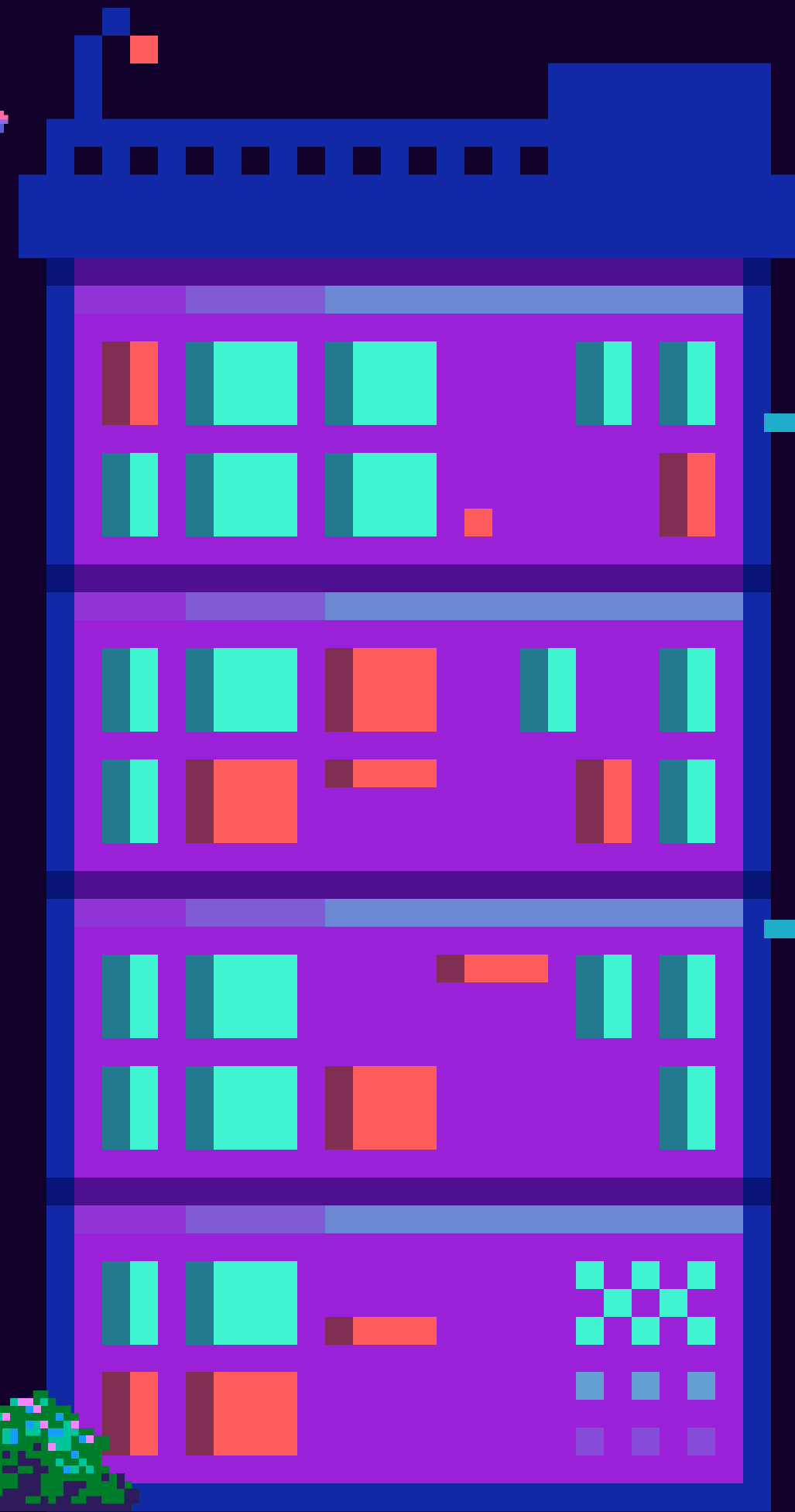
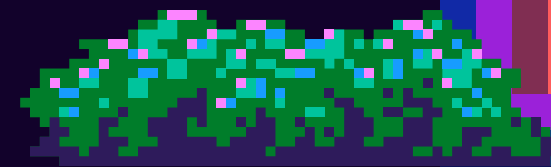
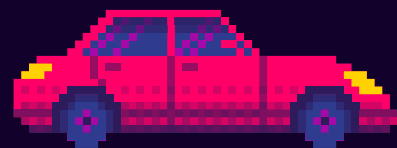
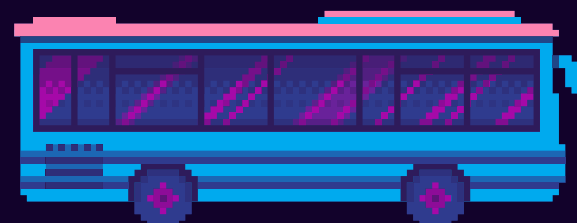
ease-out: Começa rápido e desacelera no final.

05

ease-in-out: Começo e final lentos, com aceleração no meio.

06

cubic-bezier(x1, y1, x2, y2): Define uma curva personalizada.



EXEMPLOS PRÁTICOS DE ANIMAÇÕES

Boas Práticas e Considerações

- Moderação: Evite exagero em animações para não prejudicar a performance e usabilidade.
- Perfis e Acessibilidade: Dê preferência a animações leves e rápidas, e considere usuários com restrições de movimento ou que preferem animações reduzidas.
- Compatibilidade de Navegadores: Teste as animações nos principais navegadores para garantir consistência.

```
1 @keyframes fadeIn {  
2   from { opacity: 0; }  
3   to { opacity: 1; }  
4 }  
5 .fade-in {  
6   animation: fadeIn 1s ease;  
7 }
```

```
1 /* Slide Up: Caixa deslizando para cima. */  
2  
3 @keyframes slideUp {  
4   from { transform: translateY(100px); opacity: 0; }  
5   to { transform: translateY(0); opacity: 1; }  
6 }  
7 .slide-up {  
8   animation: slideUp 0.5s ease;  
9 }  
10
```

```
1 /* Rotação: Animação de rotação com efeito infinito.*/  
2  
3 @keyframes rotate {  
4   from { transform: rotate(0deg); }  
5   to { transform: rotate(360deg); }  
6 }  
7 .rotate {  
8   animation: rotate 2s linear infinite;  
9 }
```

DESAFIOS PRÁTICOS

01

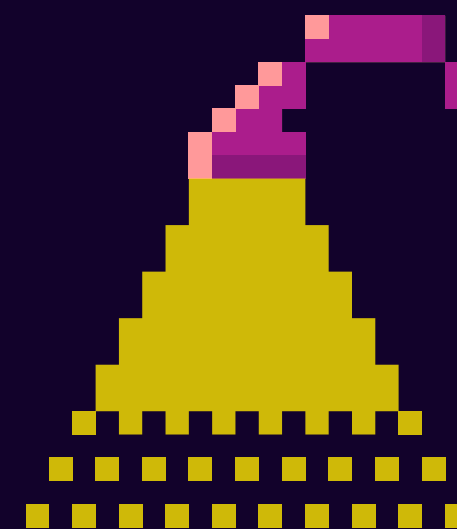
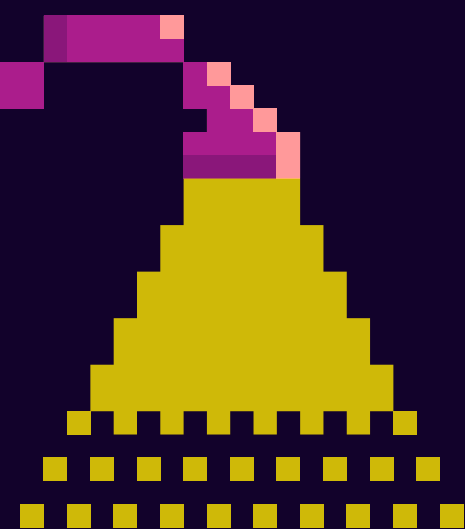
Desafio 1: Criar um botão com efeito de hover onde o botão cresce suavemente e muda de cor.

02

Desafio 2: Criar uma animação de "coração pulsante", onde o elemento cresce e encolhe em loop.

03

Desafio 3: Criar uma sequência de loading animada usando @keyframes.





ECOA

ANIMAR

NOSSOS CONTATOS:



27 99500-7495



<https://beacons.ai/prismatech>



producaoprismatech@gmail.com



Avenida Jerônimo Monteiro 145, Vitória





THANK
YOU