

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Clara</title>
7     <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10     <div id="estrelas-container"></div> <!--Contêiner para as estrelas -->
11     <section>
12         <h1>Clara </h1>
13         
14         <h2>Apresentação</h2>
15         <p>Olá</p>
16     </section>
17     <footer>&copy;SoftexPernambuco</footer>
18     <script src="script.js"></script>
19 </body>
20 </html>
```

```

59 #estrelas-container {
60     position: fixed;
61     top: 0;
62     left: 0;
63     width: 100%;
64     height: 100%;
65     pointer-events: none;
66     overflow: hidden;
67     z-index: 0;
68 }
69
70 .estrela {
71     position: absolute;
72     color: yellow;
73     font-size: 30px; /* tamanho da estrela */
74     opacity: 0.8;
75     animation: cair linear infinite;
76     pointer-events: none;
77 } /* cair → é o nome da animação (deve estar definida com @keyframes cair)
78
79 linear → é a "velocidade constante" da animação (sem aceleração ou desaceleração)
80
81 infinite → significa que a animação nunca para, ela repete para sempre
82
83 pointer-events: none;
84 Essa é uma propriedade de controle de clique/interação.
85
86 none significa que o elemento será ignorado pelo mouse.
87
88 Você não consegue clicar, arrastar ou interagir com esse elemento.
89
90 No caso das estrelas:
91 Como elas estão apenas decorando a tela, você não quer que o usuário clique nelas por acidente */
92
93 @keyframes cair { /* animação */
94     0% {
95         transform: translateY(-10px);
96         opacity: 1;
97     }
98     100% {
99         transform: translateY(100vh);
100        opacity: 0;
101    }
102 }

```

```
10 function criarEstrela() {
11     const estrela = document.createElement('div');
12     estrela.classList.add('estrela');
13     estrela.innerText = '★'; // Unicode da estrela
14
15
16     // Posição horizontal aleatória
17     estrela.style.left = Math.random() * window.innerWidth + 'px';
18
19     // Duração aleatória da animação
20     const duracao = Math.random() * 3 + 2; // entre 2 e 5 segundos
21     estrela.style.animationDuration = duracao + 's';
22
23     document.getElementById('estrelas-container').appendChild(estrela);
24
25     // Remover estrela após a animação terminar
26     setTimeout(() => {
27         estrela.remove();
28     }, duracao * 1000);
29 }
30
31 // Criar uma nova estrela a cada 150ms
32 setInterval(criarEstrela, 150);
33
34 // Geralmente, usamos Math.random() com multiplicações para gerar valores em faixas específicas.
35 // Exemplo 1: Um número aleatório entre 0 e 100
36 // setTimeout serve para executar uma função uma única vez, depois de um tempo determinado (em milissegundos).
37 // setInterval executa uma função repetidamente em intervalos de tempo fixos (em milissegundos).
38 // appendChild é um método que adiciona um novo elemento como filho de outro no DOM (Document Object Model).
```