

Criando um Botão Interativo com Efeito de Perseguição de Cor

Este tutorial demonstra como implementar um botão interativo com um efeito visual envolvente de perseguição de cor utilizando HTML, CSS e JavaScript. Este efeito reage aos movimentos do cursor do usuário.

Pré-requisitos:

Para compreender e aplicar este tutorial, alguns conhecimentos prévios são úteis:

- HTML e Estruturação de Página;
- CSS e Estilização;
- JavaScript básico;
- Manipulação de Elementos do DOM (Compreensão de como selecionar elementos HTML usando JavaScript, como `querySelector`, e de como manipular propriedades e estilos desses elementos).

Extra: Conceitos de CSS Avançados (Se quiser compreender melhor o uso de variáveis CSS (`--x`, `--y`) e pseudo-elementos (`::before`) para criar efeitos dinâmicos, um conhecimento mais avançado de CSS pode ser benéfico).

Embora não seja essencial ter um domínio avançado desses conceitos, ter uma compreensão básica de HTML, CSS e JavaScript ajudará na implementação e compreensão do tutorial. Este tutorial é ideal para quem deseja explorar efeitos visuais interativos em elementos da web e está buscando expandir suas habilidades de desenvolvedor web front-end.

Código CSS para o Efeito de Perseguição de Cor:

```
1  .color-pursuit-button::before {
2      --size: 0;
3      content: "";
4      position: absolute;
5      left: var(--x);
6      top: var(--y);
7      width: var(--size);
8      height: var(--size);
9      background: radial-gradient(circle closest-side, #5eef3a, transparent);
10     transform: translate(-50%, -50%);
11     transition: width 0.2s ease, height 0.2s ease;
12 }
13
14 .color-pursuit-button:hover::before {
15     --size: 400px;
```

Explicação Passo a Passo:

1. Pseudo-elemento **::before**:

::before é um pseudo-elemento que cria um elemento filho do elemento com a classe **.color-pursuit-button**. Ele é usado para gerar e estilizar o efeito visual.

2. Variáveis CSS (**--size**, **--x**, **--y**):

--size: Define uma variável que controla o tamanho do elemento. Inicialmente, o tamanho é 0.

--x e **--y**: Variáveis que armazenam as coordenadas X e Y do cursor.

3. Posicionamento Absoluto:

position: absolute: Define o elemento gerado (**::before**) em relação ao elemento pai (**.color-pursuit-button**).

left: var(--x) e **top: var(--y)** Posicionam o elemento nas coordenadas definidas pelas variáveis **--x** e **--y**.

4. Estilo e Transição do Elemento:

width: var(--size); e **height: var(--size);** Define a largura e altura do elemento como a variável **--size**.

background: Cria um gradiente radial que transita de uma cor (verde, #5eef3a) para transparente.

transform: translate(-50%, -50%); Centraliza o elemento na posição determinada pelas variáveis **--x** e **--y**.

transition: width 0.2s ease, height 0.2s ease: Define a transição suave do tamanho do elemento ao longo do tempo.

5. Pseudo-classe **:hover**:

.color-pursuit-button:hover::before: Quando o cursor paira sobre o botão, a **pseudo-classe :hover** ativa o efeito.

--size: 400px: Define o tamanho do elemento como 400px, expandindo-o ao passar o cursor sobre o botão.

Conclusão

Este código CSS utiliza variáveis, pseudo-elementos e pseudo-classes para criar um efeito de transição suave que segue o cursor ao interagir com um botão. Experimente ajustar as propriedades para personalizar ainda mais esse efeito visual em seus projetos web.

Interatividade com JavaScript para Efeitos Visuais:

Este tutorial explora como usar JavaScript para criar interatividade em elementos HTML, permitindo a criação de efeitos visuais dinâmicos, como o efeito de perseguição de cor em um botão.

➤ Estrutura HTML e JavaScript:

```
1 <body>
2   <button class="color-pursuit-button">
3     <span>@code55.io</span>
4   </button>
5   <script>
6     document.querySelector('.color-pursuit-button').onmousemove = (e) => {
7
8       const x = e.pageX - e.target.offsetLeft
9       const y = e.pageY - e.target.offsetTop
10
11       e.target.style.setProperty('--x', `${x}px`)
12       e.target.style.setProperty('--y', `${y}px`)
13
14     }
15   </script>
16 </body>
```

Passo a Passo da Funcionalidade JavaScript:

1. Selecionando o Elemento:

document.querySelector('.color-pursuit-button'); Utiliza o seletor CSS *.color-pursuit-button* para selecionar o botão específico.

2. Evento onmousemove:

.onmousemove = (e) => { ... }; Atribui uma função ao evento mousemove do botão. Essa função é acionada quando o cursor se move sobre o botão.

3. Captura de Coordenadas do Cursor:

const x = e.pageX - e.target.offsetLeft: Calcula a posição horizontal do cursor em relação ao canto esquerdo do botão.

const y = e.pageY - e.target.offsetTop: Calcula a posição vertical do cursor em relação ao topo do botão.

4. Atualização das Variáveis CSS (--x, --y):

e.target.style.setProperty('--x', `\${x}px)` e **e.target.style.setProperty('--y', `\${y}px`): Define as variáveis CSS **--x** e **--y** com as coordenadas X e Y do cursor, respectivamente, no elemento **e.target** (o botão).**

Conclusão

Este código JavaScript permite rastrear os movimentos do cursor sobre o botão e atualizar as variáveis CSS **--x** e **--y** de acordo com as coordenadas do cursor. Essas variáveis CSS são usadas para criar efeitos visuais, como o efeito de perseguição de cor, que dependem da posição do cursor para serem ativados.

Experimente incorporar essa lógica de rastreamento do cursor em seus projetos para adicionar interatividade e efeitos visuais dinâmicos aos elementos da sua página!