

# Carrossel de Imagens com HTML, CSS e JavaScript

## 1. Estrutura do Arquivo index.html

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Carrossel</title>
    <link rel="stylesheet" href="styles.css">
    <script src="script.js" defer></script>
  </head>
  <body>
    <div id="carousel">
      <div id="left"></div>
      <div id="display">
        <img id="carousel-image">
      </div>
      <div id="right"></div>
    </div>
  </body>
</html>
```

## 2. Estilos no Arquivo styles.css

```
/*
```

Folha de estilos CSS para um simples carrossel de imagens concebido da seguinte forma:

- 3 colunas flex ocupando, respectivamente, 2.5% 95% e 2.5% da largura do contêiner do carrossel, onde as colunas laterais são usadas para navegação (setas esquerda e direita) e a coluna do meio exibe a imagem atual do carrossel.
- A largura do contêiner será de 50% da largura da janela de visualização (viewport) e o contêiner estará centralizado horizontalmente na página.
- A altura do contêiner do carrossel será fixa em 400px, e a imagem exibida na coluna do meio deve se ajustar proporcionalmente para caber dentro dessa altura, mantendo sua proporção original.
- As colunas laterais serão como abas de navegação com fundos claros apenas, para indicar que são clicáveis, e quando clicadas, atualizarão os valores do cursor do carrossel para mostrar a imagem anterior ou seguinte, devendo ter uma pequena animação de deslocamento ao serem clicadas, bem como, um efeito de transição suave no container de imagem do carrossel ao mudar de imagem, tanto para a imagem anterior quanto para a próxima.
- No html estarão dispostos por meio de elementos com ids específicos para que o CSS e o JavaScript possam manipulá-los adequadamente.

```
*/
```

```

/* Estilos gerais do carrossel */
#carousel {
  display: flex;
  height: 400px;
  max-width: 50vw;
  margin: 0 auto;
  overflow: hidden;
  position: relative;
}
#left, #right {
  flex: 0 0 2.5%;
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  background-color: #ccccccc;
  transition: transform 0.2s ease;
}

#left:hover,
#right:hover {
  transform: scale(1.1);
}

#display {
  flex: 0 0 95%;
  overflow: hidden;
  transition: transform 0.5s ease;
}

#carousel-image {
  height: 100%;
  width: 100%;
  object-fit: contain;
  transition: opacity 0.5s ease;
}

```

### 3. Lógica no Arquivo script.js

```

/*
  Esse script contém a lógica necessária para o funcionamento do carrossel
  de imagens que estamos implementando. É baseado em um cursor que controla
  o índice da imagem atualmente visível e duas abas onde cliques são efetuados
  que incrementam ou decrementam o índice do cursor para navegar pelas imagens.
*/

// Array que contém as URLs das imagens do carrossel
let img_urls = [
  "img/1.jpg",
  "img/2.webp",
  "img/3.jpg",
  "img/4.jpg"
];

// Índice da imagem atualmente visível
let cursor = 0;

```

```
// função para atualizar a imagem exibida com base no cursor
function atualizaImagem() {
    let img = document.querySelector("#carousel-image");
    img.src = img_urls[cursor];
}

// Função para avançar para a próxima imagem
function proximaImagem() {
    cursor++;
    if (cursor >= img_urls.length) {
        cursor = 0; // Volta para a primeira imagem se estiver na última
    }
    atualizaImagem();
}

// Função para voltar para a imagem anterior
function imagemAnterior() {
    cursor--;
    if (cursor < 0) {
        cursor = img_urls.length - 1; // Vai para a última imagem se estiver na primeira
    }
    atualizaImagem();
}

// Adiciona event listeners (on clicks) às abas de navegação
let abaEsquerda = document.querySelector("#left");
abaEsquerda.addEventListener("click", imagemAnterior);
let abaDireita = document.querySelector("#right");
abaDireita.addEventListener("click", proximaImagem);

// Atualiza a imagem inicial ao carregar a página
window.onload = atualizaImagem;
```