

Variáveis Simples

Teoria:

Variáveis simples armazenam um único valor. Elas podem conter tipos de dados primitivos como números, strings, booleanos, etc.

Exemplo:

```
// Variável simples do tipo número
let idade = 25;

// Variável simples do tipo string
let nome = "Ana";

// Variável simples do tipo booleano
let casado = true;
```

Variáveis Compostas

Teoria:

Variáveis compostas podem armazenar múltiplos valores. Em JavaScript, os principais tipos de variáveis compostas são arrays e objetos.

Exemplos:

Array:

```
// Array que armazena múltiplos valores do mesmo tipo
let frutas = ["maçã", "banana", "laranja"];

// Acessando um valor do array
console.log(frutas[1]); // Saída: banana
```

Objeto:

```
// Objeto que armazena múltiplos valores nomeados
let pessoa = {
  nome: "Carlos",
  idade: 30,
  casado: false
};

// Acessando valores do objeto
console.log(pessoa.nome); // Saída: Carlos
console.log(pessoa["idade"]); // Saída: 30
```

Variáveis de Variáveis

```
let a = 10;
let b = 20;
let c = 30;

// Array de variáveis
let variaveis = [a, b, c];

// Acessando valores através do array
console.log(variaveis[0]); // Saída: 10
console.log(variaveis[1]); // Saída: 20
```

Teoria:

Variáveis de variáveis referem-se a estruturas onde o valor de uma variável é outra variável ou mesmo uma referência dinâmica. Em JavaScript, isso pode ser feito de maneiras criativas, como utilizando arrays ou objetos para armazenar referências de variáveis.

Exemplo:

Utilizando Arrays:

```
let a = 10;
let b = 20;
let c = 30;

// Array de variáveis
let variaveis = [a, b, c];

// Acessando valores através do array
console.log(variaveis[0]); // Saída: 10
console.log(variaveis[1]); // Saída: 20
```

Utilizando Objetos:

```
let x = "valorX";
let y = "valorY";

// Objeto que referencia variáveis
let referencias = {
  refX: x,
  refY: y
};

// Acessando valores através do objeto
console.log(referencias.refX); // Saída: valorX
console.log(referencias.refY); // Saída: valorY
```

Exemplos Práticos

Exemplo com Função e Escopo:

```
function exemploVariaveis() {
  let variavelSimples = "Olá, Mundo!";
  let arrayExemplo = ["um", "dois", "três"];
  let objetoExemplo = { a: 1, b: 2, c: 3 };

  console.log(variavelSimples); // Saída: Olá, Mundo!
  console.log(arrayExemplo[0]); // Saída: um
  console.log(objetoExemplo.a); // Saída: 1
}

exemploVariaveis();
```

Exemplo com Variáveis Dinâmicas (variáveis de variáveis):

```
let chave = "dinamica";
let objetoDinamico = {
  dinamica: "Este é um valor dinâmico"
};

console.log(objetoDinamico[chave]); // Saída: Este é um valor dinâmico
```

Criando a Estrutura HTML

Crie um arquivo `index.html` com o seguinte conteúdo:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Variáveis em JavaScript</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Variáveis em JavaScript</h1>
    <p id="variavel-simples"></p>
    <p id="variavel-composta"></p>
    <p id="variavel-de-variaveis"></p>
    <button id="mostrar-variaveis">Mostrar Variáveis</button>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

Adicionando Estilos com CSS

Crie um arquivo `styles.css` com o seguinte conteúdo:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

button {
  display: block;
  margin: 20px 0 0;
  padding: 10px 20px;
  background-color: #007bff;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}
```

Implementando a Lógica com JavaScript

Crie um arquivo `script.js` com o seguinte conteúdo:

```
document.getElementById('mostrar-variaveis').addEventListener('click', function() {  
  // Variáveis Simples  
  let idade = 25;  
  let nome = "Ana";  
  let casado = true;  
  
  document.getElementById('variavel-simples').innerText = `Simples: Nome: ${nome}, Idade: ${idade}, Casado: ${casado}`;  
  
  // Variáveis Compostas  
  let frutas = ["maçã", "banana", "laranja"];  
  let pessoa = {  
    nome: "Carlos",  
    idade: 30,  
    casado: false  
  };  
  
  document.getElementById('variavel-composta').innerText = `Compostas: Frutas: ${frutas.join(', ')},  
  Pessoa: Nome: ${pessoa.nome}, Idade: ${pessoa.idade}, Casado: ${pessoa.casado}`;  
  
  // Variáveis de Variáveis  
  let a = 10;  
  let b = 20;  
  let c = 30;  
  
  let variaveis = [a, b, c];  
  let chave = "dinamica";  
  let objetoDinamico = {  
    dinamica: "Este é um valor dinâmico"  
  };  
  
  document.getElementById('variavel-de-variaveis').innerText = `Variáveis de Variáveis:  
  Array: ${variaveis.join(', ')}, Objeto Dinâmico: ${objetoDinamico[chave]}`;  
});
```

Exercício Simples

Objetivo do Exercício: Crie uma página que permita ao usuário adicionar frutas a um array e exibir o array atualizado.

Atualizando o HTML

Atualize o arquivo `index.html` para incluir um formulário:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Variáveis em JavaScript</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Variáveis em JavaScript</h1>
    <p id="variavel-simples"></p>
    <p id="variavel-composta"></p>
    <p id="variavel-de-variaveis"></p>
    <button id="mostrar-variaveis">Mostrar Variáveis</button>

    <h2>Adicionar Frutas</h2>
    <form id="form-frutas">
      <input type="text" id="nova-fruta" placeholder="Digite uma fruta" required>
      <button type="submit">Adicionar Fruta</button>
    </form>
    <p id="lista-frutas"></p>
  </div>
  <script src="script.js"></script>
</body>
</html>
```


Atualizando o JavaScript

Atualize o arquivo `script.js` para incluir a lógica do exercício:

```
document.getElementById('mostrar-variaveis').addEventListener('click', function() {
  // Variáveis Simples
  let idade = 25;
  let nome = "Ana";
  let casado = true;

  document.getElementById('variavel-simples').innerText = `Simples: Nome: ${nome},
    Idade: ${idade}, Casado: ${casado}`;

  // Variáveis Compostas
  let frutas = ["maçã", "banana", "laranja"];
  let pessoa = {
    nome: "Carlos",
    idade: 30,
    casado: false
  };

  document.getElementById('variavel-composta').innerText = `Compostas: Frutas:
    ${frutas.join(', ')}, Pessoa: Nome: ${pessoa.nome}, Idade: ${pessoa.idade}, Casado:
    Casado: ${pessoa.casado}`;

  // Variáveis de Variáveis
  let a = 10;
  let b = 20;
  let c = 30;

  let variaveis = [a, b, c];
  let chave = "dinamica";
  let objetoDinamico = {
    dinamica: "Este é um valor dinâmico"
  };

  document.getElementById('variavel-de-variaveis').innerText = `Variáveis de Variáveis:
    Array: ${variaveis.join(', ')}, Objeto Dinâmico: ${objetoDinamico[chave]}`;
});

// Exercício: Adicionar Frutas ao Array
let frutasArray = ["maçã", "banana", "laranja"];

document.getElementById('form-frutas').addEventListener('submit', function(event) {
  event.preventDefault();
  let novaFruta = document.getElementById('nova-fruta').value;
  frutasArray.push(novaFruta);
  document.getElementById('lista-frutas').innerText = `Lista de Frutas: ${frutasArray.join(', ')}`;
  document.getElementById('nova-fruta').value = '';
});
```

Testando a Aplicação

1. Abra o arquivo `index.html` no seu navegador.
2. Clique no botão "Mostrar Variáveis" para ver os exemplos de variáveis simples, compostas e de variáveis.
3. Adicione novas frutas no formulário e veja a lista de frutas sendo atualizada dinamicamente.

Exercício 1: Calculadora Simples

Objetivo do Exercício: Criar uma calculadora simples que permita ao usuário fazer operações básicas de adição, subtração, multiplicação e divisão.

Passo 1: Atualizando o HTML

Atualize o arquivo `index.html` para incluir os elementos da calculadora:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculadora Simples</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Calculadora Simples</h1>
    <input type="number" id="numero1" placeholder="Número 1">
    <input type="number" id="numero2" placeholder="Número 2">
    <div class="operacoes">
      <button onclick="calcular('+')">+</button>
      <button onclick="calcular('-')">-</button>
      <button onclick="calcular('*')">*</button>
      <button onclick="calcular('/')">/</button>
    </div>
    <p id="resultado"></p>
  </div>
  <script src="script.js"></script>
</body>
</html>
```


Passo 2: Adicionando Estilos CSS

Atualize o arquivo `styles.css` para incluir estilos para a lista de tarefas:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

button {
  display: block;
  margin: 20px 0 0;
  padding: 10px 20px;
  background-color: #007bff;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}
```

Passo 3: Atualizando o JavaScript

Atualize o arquivo `script.js` para incluir a lógica da calculadora:

Exercício 2: Lista de Tarefas

Objetivo do Exercício: Criar uma lista de tarefas onde o usuário possa adicionar novas tarefas, marcar como concluídas e remover tarefas da lista.

Passo 1: Atualizando o HTML

Atualize o arquivo `index.html` para incluir a lista de tarefas:

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lista de Tarefas</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Lista de Tarefas</h1>
    <form id="form-tarefa">
      <input type="text" id="nova-tarefa" placeholder="Digite uma nova tarefa" required>
      <button type="submit">Adicionar Tarefa</button>
    </form>
    <ul id="lista-tarefas"></ul>
  </div>
  <script src="script.js"></script>
</body>
</html>

```

Passo 2: Adicionando Estilos CSS

Atualize o arquivo `styles.css` para incluir estilos para a lista de tarefas:

```

body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

```

```

button {
  display: block;
  margin: 20px 0 0;
  padding: 10px 20px;
  background-color: #007bff;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}

```

Passo 3: Atualizando o JavaScript

Atualize o arquivo `script.js` para incluir a lógica da lista de tarefas:

Exercício 3: Gerador de Números Aleatórios

Objetivo do Exercício: Criar um gerador de números aleatórios que gera um número entre 1 e 10 quando o usuário clica em um botão.

Passo 1: Atualizando o HTML

Atualize o arquivo `index.html` para incluir o botão do gerador de números:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Gerador de Números Aleatórios</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Gerador de Números Aleatórios</h1>
    <button id="gerar-numero">Gerar Número Aleatório</button>
    <p id="numero-gerado"></p>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

Passo 2: Atualizando o JavaScript

Atualize o arquivo `script.js` para incluir a lógica do gerador de números:

Exercício 4: Contador de Cliques

Objetivo do Exercício: Criar um contador de cliques que mostra quantas vezes um botão foi clicado.

Passo 1: Atualizando o HTML

Atualize o arquivo `index.html` para incluir o botão e o contador de cliques:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contador de Cliques</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Contador de Cliques</h1>
    <button id="botao-cliques">Clique Aqui</button>
    <p id="contador-cliques">Cliques: 0</p>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

Passo 2: Atualizando o JavaScript

Atualize o arquivo `script.js` para incluir a lógica do contador de cliques:

Testando os Exercícios

1. Abra os arquivos `index.html`, `styles.css` e `script.js` no seu editor de código.
 2. Execute os exercícios em seu navegador para testar as funcionalidades da calculadora e da lista de tarefas.
- Divirta-se criando!