

**Bora continuar ver mais conceitos  
básicos de desenvolvimento**

# **Estruturas Condicionais**

# O que são Estruturas Condicionais?

- Permitem que o programa tome decisões baseadas em condições.
- Execução de diferentes blocos de código dependendo se a condição é verdadeira ou falsa.

# if Statement

```
let idade = 18;

if (idade >= 18) {
  console.log('Você é maior de idade.');
```

  

```
} else {
  console.log('Você é menor de idade.');
```

  

```
}
```

- **if** verifica a condição, e se verdadeira, executa o bloco de código dentro das chaves.

## 4: `else if` e `else`

```
let hora = 14;

if (hora < 12) {
  console.log('Bom dia!');
} else if (hora < 18) {
  console.log('Boa tarde!');
} else {
  console.log('Boa noite!');
}
```

- Encadeamento de condições com `else if`.
- `else` captura todas as outras situações.

# Operador Ternário

```
let clima = 'ensolarado';  
let acao = clima === 'ensolarado' ? 'Vou à praia!' : 'Ficarei em casa.';  
console.log(acao);
```

- Forma concisa de escrever uma estrutura condicional em uma única linha.

# switch Statement

```
let diaDaSemana = 3;
let mensagem = '';

switch (diaDaSemana) {
  case 1:
    mensagem = 'Segunda-feira';
    break;
  case 2:
    mensagem = 'Terça-feira';
    break;
  default:
    mensagem = 'Dia inválido';
}
console.log(mensagem);
```

Útil quando há muitas condições a serem testadas.



# Operadores Lógicos

```
let temCartao = true;
let saldo = 100;

if (temCartao || saldo > 50) {
  console.log('Compra aprovada!');
} else {
  console.log('Saldo insuficiente e sem cartão.');
```

- `&&` (AND), `||` (OR), `!` (NOT) para combinar condições.

- Estruturas condicionais são cruciais para a lógica de programação.
- Permitem que o código tome decisões dinâmicas com base em condições específicas.

# Arrays em JavaScript

- Arrays são coleções ordenadas de elementos.
- Podem armazenar diferentes tipos de dados.
- Índices começam em 0.

```
let frutas = ['maçã', 'banana', 'laranja'];  
console.log(frutas[1]); // Output: banana
```

# Métodos de Array - **push** e **pop**

```
let animais = ['cachorro', 'gato', 'passarinho'];  
  
animais.push('peixe'); // Adiciona 'peixe' ao final do array  
console.log(animais); // Output: ['cachorro', 'gato', 'passarinho', 'peixe']  
  
animais.pop(); // Remove o último elemento do array  
console.log(animais); // Output: ['cachorro', 'gato', 'passarinho']
```

# Métodos de Array - **shift** e **unshift**

```
let cores = ['verde', 'azul', 'amarelo'];  
  
cores.shift(); // Remove o primeiro elemento do array  
console.log(cores); // Output: ['azul', 'amarelo']  
  
cores.unshift('vermelho'); // Adiciona 'vermelho' no início do array  
console.log(cores); // Output: ['vermelho', 'azul', 'amarelo']
```

# Métodos de Array - splice

```
let numeros = [1, 2, 3, 4, 5];  
  
numeros.splice(2, 1); // Remove 1 elemento a partir do índice 2  
console.log(numeros); // Output: [1, 2, 4, 5]  
  
numeros.splice(1, 0, 6, 7); // Adiciona 6 e 7 no índice 1  
console.log(numeros); // Output: [1, 6, 7, 2, 4, 5]
```

# Objetos em JavaScript

- Objetos são coleções de pares chave-valor.
- As chaves são strings que identificam os valores associados.

```
let pessoa = {  
  nome: 'João',  
  idade: 30,  
  profissao: 'engenheiro',  
};
```

```
console.log(pessoa.nome); // Output: João
```

# Acessando e Modificando Propriedades de Objetos

```

pessoa.idade = 31; // Modificando o valor da propriedade 'idade'
console.log(pessoa.idade); // Output: 31

pessoa['profissao'] = 'desenvolvedor'; // Outra forma de modificar propriedades
console.log(pessoa.profissao); // Output: desenvolvedor

```



# Adicionando e Removendo Propriedades de Objetos

```
pessoa.cidade = 'São Paulo'; // Adicionando nova propriedade
console.log(pessoa); // Output: { nome: 'João', idade: 31, profissao: 'desenvolvedor', cidade: 'São Paulo' }

delete pessoa.idade; // Removendo a propriedade 'idade'
console.log(pessoa); // Output: { nome: 'João', profissao: 'desenvolvedor', cidade: 'São Paulo' }
```

# Iterando em Arrays e Objetos

```
let numeros = [1, 2, 3, 4, 5];

numeros.forEach(function (numero) {
  console.log(numero);
});
// Output:
// 1
// 2 ...

for (let chave in pessoa) {
  console.log(chave + ': ' + pessoa[chave]);
}
// Output:
// nome: João
// profissao: desenvolvedor
// cidade: São Paulo
```

- Arrays e objetos são fundamentais para organizar e manipular dados em JavaScript.
- A compreensão dessas estruturas é crucial para o desenvolvimento de aplicações web eficientes e escaláveis.

Atividade:

<https://gist.github.com/oAllanWeb/1f11016bf3af171a401ed7ba2eedf636>