

JS



INTRODUÇÃO SIMPLES AO JAVASCRIPT

POLAR DO JS

Introdução ao Javascript

- Arrays
- Funções
- Condições



Capítulo 01

Arrays



ARRAYS

Arrays em Javascript: Arrays são como caixas que guardam várias coisas. Cada coisa dentro da caixa tem um número especial chamado índice, que começa em zero E aumenta para cada nova coisa na caixa.



ARRAYS

Operações com Arrays: Podemos realizar diversas operações com arrays, como adicionar elementos, remover elementos, acessar elementos específicos e percorrer todos os elementos do array. Isso torna os arrays uma ferramenta poderosa para armazenar e manipular dados em JavaScript.



ARRAYS

Exemplo prático: Se pensarmos em uma caixa de brinquedos, podemos adicionar novos brinquedos, tirar aqueles que não queremos mais, pegar um brinquedo específico quando queremos brincar com ele ou até mesmo contar quantos brinquedos temos na caixa.



ARRAYS

Aplicação real: Quando fazemos uma lista de compras para ir ao mercado, essa lista é como uma caixa onde colocamos todos os itens que precisamos comprar.

Quando pegamos um item da lista, podemos riscá-lo, como fazemos quando retiramos um brinquedo da caixa.



ARRAYS

O que é percorrer um array?

Percorrer um array é como contar os itens de uma lista, um por um.

Como funciona?

Podemos imaginar que estamos olhando cada item da lista, um de cada vez, e fazendo algo com ele.



ARRAYS

Por que é útil?

Percorrer um array nos permite fazer algo com cada item da lista sem ter que repetir o mesmo código várias vezes.

Exemplo prático:

Imagine que temos uma lista de compras e queremos ver cada item da lista. Podemos olhar para cada item e dizer em voz alta qual é.



ARRAYS

Aplicação real:

Percorrer um array é como ler um livro, página por página. Podemos fazer algo com cada palavra ou imagem do livro enquanto o lemos. Da mesma forma, podemos fazer algo com cada item de uma lista enquanto a percorremos.



ARRAYS



Polar do JS

```
let frutas = ['maçã', 'banana', 'laranja', 'uva'];  
  
frutas.forEach(function(fruta) {  
    console.log(fruta);  
});
```



ARRAYS

Neste código, usamos o método **forEach** para percorrer cada elemento do array **frutas** e executar uma função para cada elemento. Essa função simplesmente imprime cada elemento no console.



Capítulo 02

Funções



FUNÇÕES

Funções em JavaScript: Uma função é como uma receita que ensina o computador a fazer alguma coisa. É um bloco de código que podemos reutilizar várias vezes para realizar uma tarefa específica.



FUNÇÕES

É como uma receita de bolo. Ela ensina o computador a fazer algo específico sempre que precisarmos.

Por exemplo, podemos ter uma função que ensina o computador a contar até 10.



FUNÇÕES

Operações com Funções: Com as funções, podemos pedir ao computador para fazer várias coisas diferentes, como somar números, dizer "oi" ou até mesmo desenhar uma carinha feliz.



FUNÇÕES

Exemplo prático: Podemos criar uma função chamada "contarMaças" que ensina o computador a contar quantas maçãs temos em uma cesta. Sempre que quisermos saber, podemos pedir para a função contar para nós.



FUNÇÕES



Polar do JS

```
function contarMacas(cesta) {  
  var contador = 0;  
  for (var i = 0; i < cesta.length; i++) {  
    if (cesta[i] === 'maçã') {  
      contador++;  
    }  
  }  
  return contador;  
}
```



FUNÇÕES

Nesta função simples, percorremos cada item da cesta (um array de frutas) e contamos quantos deles são "maçã". Se encontrarmos uma maçã, aumentamos o contador. No final, retornamos o total de maçãs encontradas.



FUNÇÕES

Benefícios das Funções: As funções nos ajudam a organizar nossas instruções para o computador, assim como organizar brinquedos em uma caixa. Elas facilitam a reutilização de instruções sem precisar escrever tudo de novo.



FUNÇÕES

Aplicação real: Quando jogamos um jogo de computador ou usamos um aplicativo no celular, muitas funções estão trabalhando juntas para nos ajudar a jogar ou fazer algo divertido. Por exemplo, uma função pode ser responsável por mostrar o nosso personagem se movendo na tela.



Capítulo 03

Condições



CONDIÇÕES

- O que são condições?

Condições são como perguntas que fazemos ao computador para saber se algo é verdadeiro ou falso



CONDIÇÕES

Como funcionam?

Por exemplo, podemos perguntar ao computador se um número é maior que outro, se um nome é igual a outro, ou se algo está acontecendo.



CONDIÇÕES

Por que são importantes?

Condições nos ajudam a tomar decisões no nosso código.

Podemos fazer o computador fazer uma coisa se a resposta for verdadeira e outra coisa se a resposta for falsa.



CONDIÇÕES

Exemplo prático:

Imagine que queremos saber se está chovendo para decidir se devemos levar um guarda-chuva.

Podemos perguntar ao computador "Está chovendo?" Se a resposta for sim, levamos o guarda-chuva, se não, não levamos.



CONDIÇÕES

Aplicação real:

As condições são como bifurcações em uma estrada. Podemos escolher um caminho se algo for verdadeiro e outro se for falso. Isso nos ajuda a escrever programas que se adaptam às diferentes situações.



CONDIÇÕES



Polar do JS

```
var idade = 20;  
  
if (idade ≥ 18) {  
    console.log("Você pode dirigir!");  
} else {  
    console.log("Você ainda não pode dirigir.");  
}
```



CONDIÇÕES

Neste exemplo, nós definimos a variável **idade** como 20. Então, usamos um **if** para verificar se a idade é maior ou igual a 18. Se for **verdadeiro**, mostramos a mensagem "Você pode dirigir!". Se for **falso**, mostramos a mensagem "Você ainda não pode dirigir."



OBRIGADO POR LER ATÉ AQUI

- Este conteúdo foi gerado por IA e diagramado por humanos. Embora tenhamos feito o possível para garantir a precisão, este material é destinado apenas para fins didáticos e pode conter erros.
- Agradecemos o seu interesse e feedback para aprimorar este eBook. Esperamos que seja útil em sua jornada de aprendizado de JavaScript. Boa leitura!

