

# ESTRUTURA DE DADOS AULA

6

MSC SANDIR RODRIGUES CAMPOS

# ARROW FUNCTIONS

- AS ARROW FUNCTIONS SÃO UMA FORMA CONCISA DE DEFINIR FUNÇÕES EM JAVASCRIPT. ELAS FORAM INTRODUZIDAS NO ECMA SCRIPT 2015 (ES6) E SÃO AMPLAMENTE UTILIZADAS DESDE ENTÃO.
- SINTAXE:
  - A SINTAXE BÁSICA DE UMA ARROW FUNCTION É:
    - (PARAMETROS) => { CORPO DA FUNÇÃO }

# EXEMPLOS

## EXEMPLO 1: FUNÇÃO SIMPLES

```
CONST SAUDAR = (NOME) => {  
  console.log(`Olá, ${NOME}!`)  
};  
SAUDAR("João"); // Olá, João!
```

## EXEMPLO 2: FUNÇÃO COM MÚLTIPLOS PARÂMETROS

```
CONST SOMAR = (A, B) => { return A + B };  
console.log(SOMAR(2, 3)); // 5
```

## EXEMPLO 3: FUNÇÃO COM CORPO IMPLÍCITO

```
CONST DOBRAR = (X) => X * 2;  
console.log(DOBRAR(5)); // 10
```

# OBJETOS

EM JAVASCRIPT, UM OBJETO É UMA COLEÇÃO DE PROPRIEDADES, ONDE CADA PROPRIEDADE É UM PAR DE CHAVE-VALOR. AS CHAVES SÃO STRINGS, E OS VALORES PODEM SER DE QUALQUER TIPO, INCLUINDO NÚMEROS, STRINGS, BOOLEANS, ARRAYS, FUNÇÕES E OUTROS OBJETOS.

## TIPOS DE OBJETOS EM JAVASCRIPT

EXISTEM DOIS TIPOS PRINCIPAIS DE OBJETOS EM JAVASCRIPT:

1. **OBJETOS LITERAIS:** SÃO OBJETOS CRIADOS USANDO A SINTAXE DE OBJETO LITERAL, QUE É UMA FORMA CONCISA DE CRIAR OBJETOS.
2. **OBJETOS CONSTRUÍDOS:** SÃO OBJETOS CRIADOS USANDO UM CONSTRUTOR DE OBJETO, QUE É UMA FUNÇÃO QUE CRIA UM NOVO OBJETO.

# OBJETO LITERAL COM RETURN

- `CONST PESSOA = {`
- `NOME: "João",`
- `IDADE: 30,`
- `SAUDACAO: FUNCTION() {`
- `RETURN `Olá, meu nome é ${this.nome} e tenho ${this.idade} anos.`;`
- `}`
- `};`

## OBJETO CONSTRUÍDO

- A SINTAXE DE OBJETO CONSTRUÍDO É A SEGUINTE:

```
FUNCTION CONSTRUTOROBJETO(CHAVE1, VALOR1, CHAVE2, VALOR2, ...) {  
  THIS.CHAVE1 = VALOR1;  
  THIS.CHAVE2 = VALOR2;  
  ...}  
CONST OBJETO = NEW CONSTRUTOROBJETO(VALOR1, VALOR2, ...)
```

# OBJETO CONSTRUÍDO

EXEMPLO DE OBJETO CONSTRUÍDO

```
FUNCTION PESSOA(NOME, IDADE, CIDADE) {
```

```
  THIS.NOME = NOME;
```

```
  THIS.IDADE = IDADE;
```

```
  THIS.CIDADE = CIDADE;
```

```
}
```

```
CONST PESSOA = NEW PESSOA("JOÃO", 30, "SÃO PAULO");
```

```
CONSOLE.LOG(PESSOA)
```

# CRIANDO OBJETO

VOCÊ PODE CRIAR UM OBJETO EM JAVASCRIPT USANDO A SINTAXE DE OBJETO LITERAL:

```
CONST PESSOA = {  
  NOME: "NICOLE",  
  IDADE: 20,  
  CIDADE: "SÃO JOÃO DOS ANÉIS"  
};
```

# ACESSANDO UM OBJETO

VOCÊ PODE ACESSAR AS PROPRIEDADES DE UM OBJETO USANDO A SINTAXE DE PONTO (.) OU COLCHETES ([]):

```
console.log(pessoa.nome); // NICOLE
```

```
console.log(pessoa["idade"]); // 20
```

# ADICIONAR PROPRIEDADES A UM OBJETO

VOCÊ PODE ADICIONAR PROPRIEDADES A UM OBJETO USANDO A SINTAXE DE PONTO (.) OU COLCHETES ([]):

```
PESSOA.PAIS = "BRASIL";  
PESSOA["ESTADO"] = "GOIAS";  
CONSOLE.LOG(PESSOA);  
// {  
//   NOME: "NICOLE",  
//   IDADE: 20,  
//   CIDADE: "FORMOSA",  
//   PAIS: "BRASIL",  
//   ESTADO: "GOIAS"  
// }
```

# REMOVER PROPRIEDADES DE UM OBJETO

VOCÊ PODE REMOVER PROPRIEDADES DE UM OBJETO USANDO A PALAVRA-CHAVE DELETE:

```
DELETE PESSOA.CIDADE;
```

```
CONSOLE.LOG(PESSOA);
```

```
// { // NOME: "NICOLE", // IDADE: 20, // PAIS: "BRASIL", // ESTADO: "GOIAS" // }
```

# VERIFICAR SE UMA PROPRIEDADE EXISTE EM UM OBJETO

VOCÊ PODE VERIFICAR SE UMA PROPRIEDADE EXISTE EM UM OBJETO USANDO A PALAVRA-CHAVE IN:

```
console.log("NOME" in pessoa); // true
```

```
console.log("ENDERECO" in pessoa); // false
```

# ITERAR SOBRE AS PROPRIEDADES DE UM OBJETO

- VOCÊ PODE ITERAR SOBRE AS PROPRIEDADES DE UM OBJETO USANDO A SINTAXE FOR...IN:

```
FOR (CONST PROP IN PESSOA) {  
  CONSOLE.LOG(`${PROP}: ${PESSOA[PROP]}`);  
} // NOME: NICOLE // IDADE: 20 // PAIS: BRASIL // ESTADO: GOIAS
```

# CONVERTER UM OBJETO EM UMA STRING

- VOCÊ PODE CONVERTER UM OBJETO EM UMA STRING USANDO A FUNÇÃO `JSON.stringify()`:

```
CONST PESSOAString = JSON.stringify(PESSOA);
```

```
console.log(PESSOAString);
```

```
// {"NOME":"JOÃO","IDADE":30,"PAIS":"BRASIL","ESTADO":"SÃO PAULO"}
```

# CONVERTER UMA STRING EM UM OBJETO

- VOCÊ PODE CONVERTER UMA STRING EM UM OBJETO USANDO A FUNÇÃO `JSON.PARSE()`:

```
CONST PESSOAOBJETO = JSON.PARSE(PESSOASTRING);
```

```
CONSOLE.LOG(PESSOAOBJETO);
```

```
// { // NOME: "JOÃO", // IDADE: 30, // PAIS: "BRASIL", // ESTADO: "SÃO PAULO" // }
```

# USANDO OBJECT.KEYS() (ITERA SOBRE AS CHAVES)

ESSE MÉTODO RETORNA UM ARRAY CONTENDO TODAS AS CHAVES DO OBJETO.

```
CONST CARRO = {  
  MARCA: "TOYOTA",  
  MODELO: "COROLLA",  
  ANO: 2020  
};
```

```
CONST CHAVES = OBJECT.KEYS(CARRO);  
CONSOLE.LOG(CHAVES); // ["MARCA", "MODELO", "ANO"]
```

```
CHAVES.FOREACH(CHAVE => {  
  CONSOLE.LOG(`${CHAVE}: ${CARRO[CHAVE]}`);  
});
```

# USANDO OBJECT.VALUES() (ITERA SOBRE OS VALORES)

ESSE MÉTODO RETORNA UM ARRAY APENAS COM OS VALORES DAS PROPRIEDADES.

```
CONST PRODUTO = {  
  NOME: "CELULAR",  
  PRECO: 1500,  
  ESTOQUE: 20  
};
```

```
CONST VALORES = OBJECT.VALUES(PRODUTO);  
CONSOLE.LOG(VALORES); // ["CELULAR", 1500, 20]
```

# USANDO OBJECT.ENTRIES() (ITERA SOBRE CHAVE-VALOR)

Esse método retorna um array de arrays, onde cada subarray contém [chave, valor].

```
CONST USUARIO = {  
  NOME: "LUCAS",  
  EMAIL: "LUCAS@EMAIL.COM",  
  IDADE: 28  
};  
CONST ENTRADAS = OBJECT.ENTRIES(USUARIO);  
CONSOLE.LOG(ENTRADAS);  
  
ENTRADAS.FOREACH(([CHAVE, VALOR]) => {  
  CONSOLE.LOG(`${CHAVE}: ${VALOR}`);  
});
```

# EXERCÍCIOS OBJETOS

## 1. CRIANDO E ACESSANDO UM OBJETO:

CRIE UM OBJETO CHAMADO CARRO COM AS SEGUINTE PROPRIEDADES:

- MARCA (STRING)
- MODELO (STRING)
- ANO (NÚMERO)
- LIGADO (BOOLEAN, INICIALIZADO COMO FALSE)

EM SEGUIDA, EXIBA CADA PROPRIEDADE NO CONSOLE

DICA: USE `console.log()` E A NOTAÇÃO DE PONTO OU COLCHETES.

# EXERCÍCIOS OBJETOS

## 2. ADICIONANDO MÉTODOS A UM OBJETO

NO MESMO OBJETO CARRO DO EXERCÍCIO ANTERIOR, ADICIONE UM MÉTODO CHAMADO LIGAR() QUE ALTERA A PROPRIEDADE LIGADO PARA TRUE E IMPRIME "O CARRO ESTÁ LIGADO".

DEPOIS, CHAME ESSE MÉTODO E VERIFIQUE O ESTADO DA PROPRIEDADE LIGADO.

# EXERCÍCIOS OBJETOS

## 3. ITERANDO SOBRE UM OBJETO

CRIE UM OBJETO CHAMADO PESSOA COM AS PROPRIEDADES:

- NOME (STRING)
- IDADE (NÚMERO)
- PROFISSAO (STRING)

PERCORRA ESSE OBJETO E EXIBA TODAS AS SUAS CHAVES E VALORES NO CONSOLE USANDO UM LOOP FOR...IN.

# EXERCÍCIOS OBJETOS

4. TRABALHANDO COM `OBJECT.KEYS()`, `OBJECT.VALUES()` E `OBJECT.ENTRIES()` DADO O OBJETO:

```
CONST PRODUTO = {  
  NOME: "NOTEBOOK",  
  PRECO: 4500,  
  ESTOQUE: 10  
};
```

1. LISTE TODAS AS CHAVES DO OBJETO USANDO `OBJECT.KEYS()`.
2. LISTE TODOS OS VALORES DO OBJETO USANDO `OBJECT.VALUES()`.
3. LISTE TODOS OS PARES CHAVE-VALOR USANDO `OBJECT.ENTRIES()`.

# EXERCÍCIOS ARROW FUNCTION

EXERCÍCIO 1: DOBRAR NÚMEROS EM UM ARRAY CRIE UMA ARROW FUNCTION QUE DOBRE CADA NÚMERO EM UM ARRAY.  
CONST NUMEROS = [1, 2, 3, 4, 5];

EXERCÍCIO 2: FILTRAR OBJETOS EM UM ARRAY CRIE UMA ARROW FUNCTION QUE FILTRE OBJETOS EM UM ARRAY COM BASE EM UMA CONDIÇÃO.  
CONST PESSOAS = [ { NOME: "JOÃO", IDADE: 25 }, { NOME: "MARIA", IDADE: 30 }, { NOME: "PEDRO", IDADE: 20 }];

EXERCÍCIO 3: SOMAR PROPRIEDADES EM UM ARRAY DE OBJETOS CRIE UMA ARROW FUNCTION QUE SOME AS PROPRIEDADES DE UM ARRAY DE OBJETOS.  
CONST PRODUTOS = [ { NOME: "PRODUTO 1", PRECO: 10.99 }, { NOME: "PRODUTO 2", PRECO: 5.99 }, { NOME: "PRODUTO 3", PRECO: 7.99 }];