

# Funções em JavaScript

## 1. Sintaxe Básica

```
function saudacao() {  
    console.log("Olá, mundo!");  
}  
  
saudacao(); // Chamada da função
```

## 2. Função com Parâmetros

```
function somar(a, b) {  
    return a + b;  
}  
  
console.log(somar(5, 3)); // 8
```

## 3. Função Anônima

```
const saudacao = function () {  
    console.log("Olá de novo!");  
};  
  
saudacao();
```

## 4. Função Arrow (=>)

```
const multiplicar = (a, b) => a * b;  
console.log(multiplicar(4, 2)); // 8
```

## 5. Função com Valor Padrão

```
function saudacao(nome = "Visitante") {  
    console.log(`Olá, ${nome}`);  
}  
  
saudacao(); // Olá, Visitante  
saudacao("Murilo"); // Olá, Murilo
```

## 6. Funções como Argumentos (Callback)

```
function processar(valor, funcao) {  
    return funcao(valor);  
}  
  
const dobrar = (n) => n * 2;  
console.log(processar(5, dobrar)); // 10
```

## Funções em JavaScript

### 7. Função Recursiva

```
function fatorial(n) {  
  if (n === 0) return 1;  
  return n * fatorial(n - 1);  
}  
console.log(fatorial(5)); // 120
```

### 8. Função Immediately Invoked (IIFE)

```
(function () {  
  console.log("Executa automaticamente!");  
})();
```

### 9. Função dentro de Objeto

```
const pessoa = {  
  nome: "João",  
  falar: function () {  
    console.log(`Olá, meu nome é ${this.nome}`);  
  }  
};  
pessoa.falar();
```

### 10. Diferença entre `function` e `=>` em objetos

```
const obj = {  
  nome: "Lucas",  
  normal() {  
    console.log(this.nome); // Lucas  
  },  
  arrow: () => {  
    console.log(this.nome); // undefined  
  }  
};  
obj.normal();  
obj.arrow();
```