

FUNÇÕES

São blocos de código que encapsulam instruções que executam uma tarefa específica. Visam modularizar os programas, isto é, dividir um programa em partes menores, em que, essas partes menores tenham uma única responsabilidade

Estrutura da função:

```
function nomeDaFunção(param1, param2 ou parenteses vazios caso a função não tenha parâmetro) {  
    código que faz alguma coisa  
}
```

Em funções, podemos usar o comando **return** para que ela retorne algum valor:

```
function nomeDaFunção(param1, param2) {  
    return param1 + param2;  
}
```

Ou podemos declarar variáveis dentro de uma função:

```
function nomeDaFunção(param1, param2) {  
    let resultado = param1 + param2;  
  
    return resultado;  
}
```

USANDO VARIÁVEIS EXTERNAS:

```
function soma(num1, num2) {
```

```
return num1 + num2;  
}
```

```
console.log(soma(3, 4));  
// 7;
```

exemplos:

FUNÇÃO PARA DESCOBRIR MAIOR PALAVRA DE UM ARRAY:

```
function maiorPalavra(palavras) {  
  let maiorPalavra = palavras[0];  
  for(let indice in palavras) {  
    if(maiorPalavra.length < palavras[indice].length) {  
      maiorPalavra = palavras[indice];  
    }  
  }  
  return maiorPalavra;  
}  
  
console.log(indiceMenorValor(['José', 'Lucas', 'Nádia', 'Fernanda', 'Cairo', 'Joana']));
```

FUNÇÃO PARA DESCOBRIR O MENOR ÍNDICE:

```
function indiceMaiorValor(numeros) {
  let indiceValor = 0;
  for(let indice in numeros) {
    if(numeros[indiceValor] < numeros[indice]) {
      indiceValor = indice;
    }
  }
  return indiceValor;
}
```

Função Controle do carro:

```
let statusCarro = "desligado";
```

```
let aceleracao = 0;
```

```
let rotacaoDoVolante = 0;
```

```
function ligarDesligar() {
  if (statusCarro === "desligado") {
    statusCarro = "ligado";
  }
  else {
    statusCarro = "desligado";
  }
  return statusCarro;
}
```

```
function acelerar(incremento) {
  aceleracao = aceleracao + incremento;
  return "Acelerando a " + aceleração + "m/s";
}
```

```
}
```

```
function frear(decremento) {  
    aceleracao = aceleracao - decremento;  
    return "Desacelerando para " + aceleracao + "m/s";  
}
```

```
function girarVolante(anguloRotacao) {  
    rotacaoDoVolante = anguloRotacao;  
    return rotacaoDoVolante + "°";  
}
```

```
function velocidadeAtual() {  
    let kmPorHora = aceleracao + 3.6;  
    return "velocidade atual: " + kmPorHora + "km/h";  
}
```

FUNÇÃO CONTROLE DA TV:

```
tv  Settings  tv -- n/work

1  * Liga/Desliga~
2  * Mudar de canal~
3  ~
4  1. Ao pressionarmos o botão de liga/desliga, a tv deve ligar ou desligar.~
5  2. Condições:~
6  ··a. Se a tv estiver desligada, então a tv entende que é para se ligar.~
7  ··b. Se a tv estiver ligada, então a tv entende que é para se desligar.~
8  ~
9  ··

1  var statusTv = 'desligada'~
2  ~
3  function ligaDesliga() {~
4  ··if (statusTv === 'desligada') {~
5  ···statusTv = 'ligada'~
6  ··} else {~
7  ···statusTv = 'desligada'~
8  ··}~
9  ··~
10 ··return statusTv~
11 }
```

FUNÇÃO RESETA COR:

Function resetaCor(){

};

Aplicando um exemplo:

Let corSite = “azul”;

Function resetaCor(){

corSite = “”;

};

Outro exemplo:

Let corSite = “azul”;

Function resetaCor(cor){

corSite = “cor”;

```
};
```

resetaCor("vermelho"); → nesse momento, após aplicar a função, a cor do site seria resetada para vermelho.

Exemplo com mais parâmetros:

```
Let corSite = "azul";
```

```
Function resetaCor(cor,tonalidade){
```

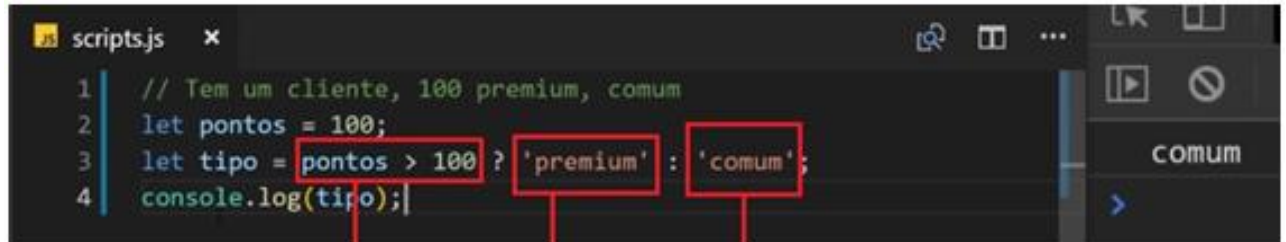
```
corSite = cor + ' ' + tonalidade;***
```

```
};
```

resetaCor("verde","claro"); → nesse momento, a cor seria redefinida para verde claro, redefinindo os dois parâmetros.

```
scripts.js x
1 // Realizar um tarefa , não devolve nada
2 function dizerNome(){
3   console.log('jhonatan');
4 }
5
6 dizerNome();
7 // Faz um cálculo ou operação e retorna algo
8 function MultiplicarPorDois(valor){
9   return valor* 2;
10 }
11
12 //console.log(MultiplicarPorDois(5));
13
14 let resultado = MultiplicarPorDois(5);
15
16 console.log(resultado);
```

OPERADOR TERNÁRIO:



```
1 // Tem um cliente, 100 premium, comum
2 let pontos = 100;
3 let tipo = pontos > 100 ? 'premium' : 'comum';
4 console.log(tipo);
```

The screenshot shows a code editor with a dark theme. The code is as follows:

```
1 // Tem um cliente, 100 premium, comum
2 let pontos = 100;
3 let tipo = pontos > 100 ? 'premium' : 'comum';
4 console.log(tipo);
```

Three red boxes highlight parts of the code on line 3:

- A box around `pontos > 100` with a line pointing to the text: "Aqui se coloca a expressão a ser comparada."
- A box around `'premium'` with a line pointing to the text: "Na primeira parte se coloca o que deve acontecer caso a expressão seja verdadeira."
- A box around `: 'comum';` with a line pointing to the text: "Na segunda parte se coloca o que deve acontecer caso a expressão seja falsa."

On the right side of the editor, a console output is visible showing the value `comum`.

Na segunda parte se coloca o que deve acontecer caso a expressão seja falsa.

Na primeira parte se coloca o que deve acontecer caso a expressão seja verdadeira.

Aqui se coloca a expressão a ser comparada.