

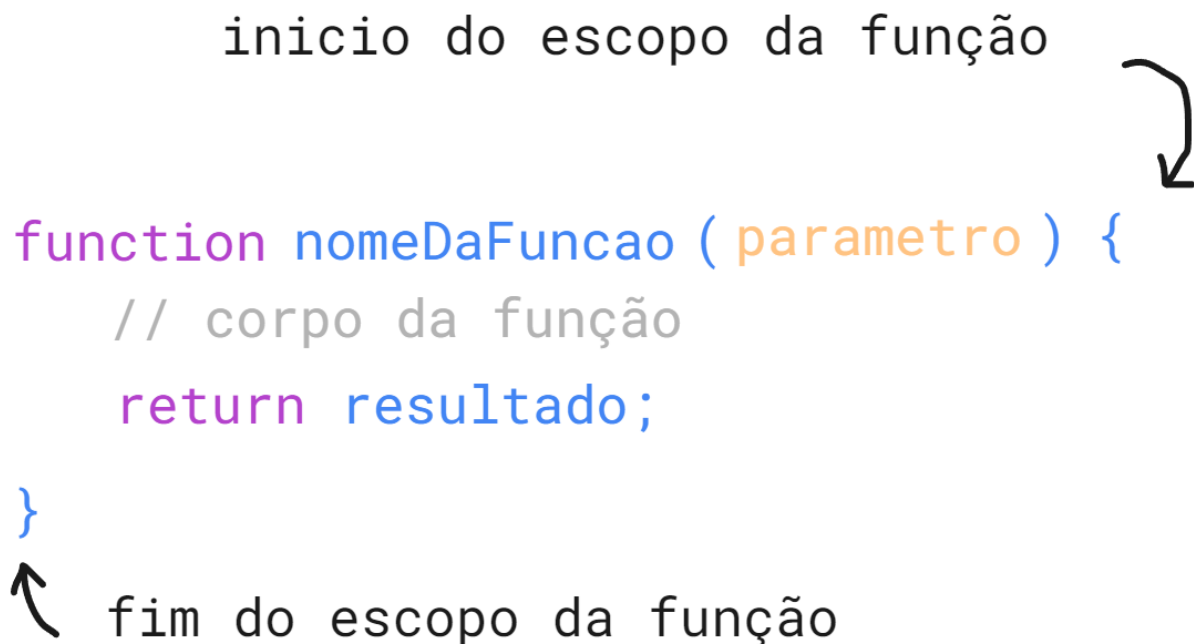
Funções servem para momentos em que vamos precisar repetir um código mais de uma vez, dessa forma, podemos isolar uma pequena parte da lógica do nosso código e chamá-la sempre que necessário. Além de tornar o código mais legível, também facilita na hora da manutenção, pois será necessário modificar um único trecho do nosso código.

A sintaxe de declaração de funções é muito semelhante às declarações de variáveis, utilizamos uma palavra reservada para indicar a declaração, nesse caso "function", seguida do nome da função¹. As funções podem ou não retornar valores:

```
function somaValores(num1, num2) {  
    return num1 + num2;  
}  
  
function logInfo(param) {  
    console.log(param);  
}
```

De modo geral, essa é a estrutura de uma função:

início do escopo da função



```
function nomeDaFuncao ( parametro ) {  
    // corpo da função  
    return resultado;  
}
```

fim do escopo da função

As funções podem possuir mais de um parâmetro, que devem ser declarados de acordo com o que será realizado.

Dentro do corpo de uma função podem ser declaradas variáveis que estarão disponíveis somente dentro do escopo da função, podemos realizar operações matemáticas ou também chamar outras funções.

Após a declaração da função, podemos chamá-la em qualquer parte do nosso código utilizando a seguinte sintaxe:

```
somaValores(1, 4);
```

Ao chamar uma função, cada parâmetro é separado por uma vírgula, os valores dos parâmetros podem ser passados diretamente como valores (exemplo acima), ou como variáveis:

```
const a = 1;
const b = 4;

somaValores(a, b);
```

Desafios

Fáceis

1. Escreva uma função que recebe um parâmetro numérico, e retorna a soma de todos os valores de 0 até o número informado;
2. Escreva uma função que recebe um parâmetro numérico, e retorna a "par" ou "ímpar", de acordo com o resto da divisão do parâmetro por 2.

Médios

1. Escreva uma função que recebe um parâmetro de texto, e retorna se o valor informado é um palíndromo;
2. Escreva uma função que recebe um parâmetro numérico, e retorna se o valor informado é um número primo.

Difícil

Escreva uma função que recebe um parâmetro de texto, e verifica se o valor informado é um CPF válido.

Explicação do cálculo do CPF

Um CPF é composto por 11 dígitos, sendo os 8 primeiros dígitos únicos, o 9º como indicador da região, e os 2 últimos os dígitos verificadores.

Para verificar se um CPF é válido, é necessário primeiro multiplicar os 9 primeiros dígitos considerando sua posição, em uma sequência decrescente de 10 a 2, ou seja, o primeiro dígito é multiplicado por 10, o segundo por 9, e assim por diante finalizando no 9º dígito sendo multiplicado por 2.

Em seguida somamos esses valores e verificamos o resto da divisão por 11.

Por fim, subtraímos o resto novamente pelo número 11, se o resultado for maior ou igual a 10, o primeiro dígito verificador é 0, caso contrário, o primeiro dígito é o resultado dessa subtração.

Para encontrar o segundo dígito verificador, realizamos o mesmo procedimento anterior, porém agora nossa sequência iniciará em 11, e incluiremos o 10º dígito encontrado nos cálculos anteriores.

A partir daqui a lógica é exatamente igual, verificamos o resto da divisão da soma dos cálculos por 11, subtraímos o resultado de 11, se for maior que 10 o dígito é 0, caso contrário, é o dígito é o encontrado na soma.

¹ O Nome da função pode ser dado a critério do programador, porém é importante nos atentarmos às boas práticas, como utilizar um verbo para descrever o que a função faz, não utilizar acentos ou caracteres como "ç".