**FUNÇÕES**

Uma função JavaScript é um bloco de código projetado para executar uma tarefa específica.

Uma função JavaScript é executada quando "algo" a invoca (a chama).

**SINTAXE DE FUNÇÃO JAVASCRIPT**

Uma função JavaScript é definida com a**function** palavra-chave, seguida por um **nome** , seguido por parênteses **()** .

Os nomes das funções podem conter letras, dígitos, sublinhados e cifrões (mesmas regras das variáveis).

Os parênteses podem incluir nomes de parâmetros separados por vírgulas:  
**( parâmetro1, parâmetro2, ... )**

O código a ser executado, pela função, é colocado entre chaves: **{}**

function name(parameter1, parameter2, parameter3) {  
  // código que será executado}

**Os parâmetros** da função são listados entre parênteses () na definição da função.

**Os argumentos** da função são os **valores** recebidos pela função quando ela é invocada.

Dentro da função, os argumentos (os parâmetros) se comportam como variáveis ​​locais.

**INVOCAÇÃO DE FUNÇÃO**

O código dentro da função será executado quando "algo" **invocar** (chamar) a função:

* Quando ocorre um evento (quando um usuário clica em um botão)
* Quando é invocado (chamado) a partir do código JavaScript
* Automaticamente (auto-invocado)

Você aprenderá muito mais sobre invocação de função posteriormente neste tutorial.

**FUNÇÃO RETORNO**

Quando o JavaScript atinge uma returninstrução, a função para de ser executada.

Se a função foi invocada a partir de uma instrução, o JavaScript "retornará" para executar o código após a instrução invocada.

As funções geralmente calculam um **valor de retorno** . O valor de retorno é "retornado" de volta ao "chamador":

function myFunction(a, b) {  
  return a \* b;

// Função vai retornar o resultado do calculo  
}

**O OPERADOR ()** INVOCA A FUNÇÃO

Usando o exemplo acima, toCelsiusrefere-se ao objeto de função e toCelsius()refere-se ao resultado da função.

Acessar uma função sem () retornará o objeto da função em vez do resultado da função.

function toCelsius(fahrenheit) {  
  return (5/9) \* (fahrenheit-32);  
}  
document.getElementById("demo").innerHTML = toCelsius;

**FUNÇÕES USADAS COMO VALORES DE VARIÁVEIS**

As funções podem ser usadas da mesma forma que você usa variáveis, em todos os tipos de fórmulas, atribuições e cálculos.

Em vez de usar uma variável para armazenar o valor de retorno de uma função:

let x = toCelsius(77);  
let text = "The temperature is " + x + " Celsius";

Você pode usar a função diretamente, como um valor variável.

let text = "The temperature is " + toCelsius(77) + " Celsius";

**VARIÁVEIS LOCAIS**

As variáveis ​​declaradas dentro de uma função JavaScript tornam-se **LOCAL** para a função.

As variáveis ​​locais só podem ser acessadas de dentro da função.

// code here can NOT use carName  
  
function myFunction() {  
  let carName = "Volvo";  
  // code here CAN use carName  
}  
  
// code here can NOT use carName

Como variáveis ​​locais só são reconhecidas dentro de suas funções, variáveis ​​com o mesmo nome podem ser utilizadas em funções diferentes.

As variáveis ​​locais são criadas quando uma função é iniciada e excluídas quando a função é concluída.

**EXEMPLO DE UMA FUNÇÃO:**

O usuário vai digitar no prompt um número qualquer, que vai ser enviado para o parâmetro ( numero ) e vai trocar com o parâmetro ( n ) da função, que vai retornar impar ou par.

function calculo(n){

    if(n % 2 == 0){

        return 'Par'

    } else {

        return 'Ímpar'

    }}

let numero = Number(prompt('Digite um numero'))

let res = calculo(numero)

alert(res)