Funções de JavaScript

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_datatypes.asp&usg=ALkJrhiM3YXAfepUkv97ajRrYJ0BQL9K2w)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_objects.asp&usg=ALkJrhi_OlO88zTloWAw1zj3NmAN_Zs4Wg)

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

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

Exemplo

function myFunction(p1, p2) {  
    return p1 \* p2;              // The function returns the product of p1 and p2  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_functions&usg=ALkJrhhQ97zWGdSc2RSpInf_Td3gI8jCuw)

Sintaxe da função JavaScript

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

Os nomes de funções podem conter letras, dígitos, sublinhados e sinais de dólar (mesmas regras que variáveis).

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

O código a ser executado, pela função, é colocado dentro de suportes curly: **{}**

function *name* ( *parameter1, parameter2, parameter3* ) {  
    *code to be executed*   
}

Os **parâmetros de** função estão listados dentro dos suportes () da definição da função.

Os **argumentos de** função são os **valores** recebidos pela função quando é invocado.

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

Uma função é muito parecida com um procedimento ou uma sub-rotina, em outras linguagens de programação.

Invocação de função

O código dentro da função será executado quando "algo" **invoca** (chamadas) a função:

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

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

Retorno de Função

Quando o JavaScript alcança uma **declaração de retorno** , a função deixará de executar.

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

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

Exemplo

Calcule o produto de dois números e retorne o resultado:

var x = myFunction(4, 3);        // Function is called, return value will end up in x  
  
function myFunction(a, b) {  
    return a \* b;                // Function returns the product of a and b  
}

O resultado em x será:

12

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_function_return&usg=ALkJrhgd8oQWBAfPcQVTFQhtjPIjnEVdtA)

Por que Funções?

Você pode reutilizar o código: defina o código uma vez e use-o muitas vezes.

Você pode usar o mesmo código muitas vezes com diferentes argumentos, para produzir resultados diferentes.

Exemplo

Converta Fahrenheit para Celsius:

function toCelsius(fahrenheit) {  
    return (5/9) \* (fahrenheit-32);  
}  
document.getElementById("demo").innerHTML = toCelsius(77);

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_farenheit_to_celsius&usg=ALkJrhh2w2XWG0oLkBZJ7iQKNINaOnEvyQ)

O operador () invoca a função

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

O acesso a uma função sem () retornará a definição da função em vez do resultado da função:

Exemplo

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_farenheit_to_celsius_2&usg=ALkJrhgaIl9SaYqI90vh0QZf5_72HuiYQw)

Funções usadas como valores variáveis

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

Exemplo

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

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

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

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_function_variable&usg=ALkJrhhQJJ998Y-w753YsMIfZV_KABR7PA)