

W3schools

Tutorials

References

Exercises

Paid Courses

Log in

HTML

CSS

JAVASCRIPT

SQL

PYTHON

PHP

BOOTSTRAP

Português

Objetos JS

Definições de Objeto

Propriedades do objeto

Métodos de Objeto

Exibição de objeto

Acessores de objeto

Construtores de objetos

Protótipos de objeto

Referência de Objeto

Mapa de Objetos ()

Conjunto de objetos ()

Funções JS

Definições de função

Parâmetros de Função

Invocação de Função

Chamada de Função

Função Aplicar

Fechamentos de funções

Parâmetros da função JavaScript

< Anterior

Próximo >

Um JavaScript **function** não executa nenhuma verificação nos valores dos parâmetros (argumentos).

Parâmetros e argumentos da função

No início deste tutorial, você aprendeu que as funções podem ter **parâmetros** :

```
function functionName(parameter1, parameter2, parameter3) {  
  // code to be executed  
}
```

Os **parâmetros da** função são os **nomes** listados na definição da função.

Os **argumentos da** função são os **valores** reais passados para (e recebidos pela) função.

Regras de Parâmetro

As definições de função JavaScript não especificam tipos de dados para parâmetros.

As funções JavaScript não executam verificação de tipo nos argumentos passados.

As funções JavaScript não verificam o número de argumentos recebidos.

Parâmetros padrão

Se uma função for chamada com **argumentos ausentes** (menos do que declarado), os valores ausentes serão definidos como **undefined** .

Às vezes, isso é aceitável, mas às vezes é melhor atribuir um valor padrão ao parâmetro:

Exemplo

```
function myFunction(x, y) {  
  if (y === undefined) {  
    y = 2;  
  }  
}
```

Tente você mesmo "

ECMAScript 2015

 permite valores de parâmetro padrão na declaração da função:  

```
function (x, y = 2) {  
  // function code  
}
```

Tente você mesmo "

## O Objeto Argumentos

As funções JavaScript têm um objeto interno denominado objeto de argumentos.

The argument object contains an array of the arguments used when the function was called (invoked).

This way you can simply use a function to find (for instance) the highest value in a list of numbers:

Example

```
x = findMax(1, 123, 500, 115, 44, 88);  
  
function findMax() {  
  let max = -Infinity;  
  for (let i = 0; i < arguments.length; i++) {  
    if (arguments[i] > max) {  
      max = arguments[i];  
    }  
  }  
  return max;  
}
```

Try it Yourself »

Or create a function to sum all input values:

Example

```
x = sumAll(1, 123, 500, 115, 44, 88);  
  
function sumAll() {  
  let sum = 0;  
  for (let i = 0; i < arguments.length; i++) {  
    sum += arguments[i];  
  }  
  return sum;  
}
```

Try it Yourself »

If a function is called with **too many arguments** (more than declared), these arguments can be reached using **the arguments object**.

## Arguments are Passed by Value

The parameters, in a function call, are the function's arguments.

JavaScript arguments are passed by **value**: The function only gets to know the values, not the argument's locations.

If a function changes an argument's value, it does not change the parameter's original value.

**As alterações nos argumentos não são visíveis (refletidas) fora da função.**

## Objetos são passados por referência

Em JavaScript, as referências de objeto são valores.

Por causa disso, os objetos se comportarão como se fossem passados por **referência**:

Se uma função altera a propriedade de um objeto, ela altera o valor original.

**As alterações nas propriedades do objeto são visíveis (refletidas) fora da função.**

< Anterior

Próximo >

Reportar erro

Fórum

Cerca de

Comprar

Principais tutoriais

HTML Tutorial

CSS Tutorial

JavaScript Tutorial

How To Tutorial

SQL Tutorial

Python Tutorial

W3.CSS Tutorial

Bootstrap Tutorial

PHP Tutorial

Java Tutorial

C++ Tutorial

jQuery Tutorial

Top References

HTML Reference

CSS Reference

JavaScript Reference

SQL Reference

Python Reference

W3.CSS Reference

Bootstrap Reference

PHP Reference

HTML Colors

Java Reference

Angular Reference

jQuery Reference

Top Examples

HTML Examples

CSS Examples

JavaScript Examples

How To Examples

SQL Examples

Python Examples

W3.CSS Examples

Bootstrap Examples

PHP Examples

Java Examples

XML Examples

jQuery Examples

Web Courses

HTML Course

CSS Course

JavaScript Course

Front End Course

SQL Course

Python Course

PHP Course

jQuery Course

Java Course

C++ Course

C# Course

XML Course

Get Certified »

W3Schools é otimizado para aprendizagem e treinamento. Os exemplos podem ser simplificados para melhorar a leitura e o aprendizado. Tutoriais, referências e exemplos são constantemente revisados para evitar erros, mas não podemos garantir a correção total de todo o conteúdo. Ao usar o W3Schools, você concorda em ter lido e aceito nossos termos de uso , cookies e política de privacidade .

Copyright 1999-2021 de Refsnes Data. Todos os direitos reservados.  
W3Schools é desenvolvido por W3.CSS .

W3schools

COLOR PICKER

COMO NÓS

Obtenha a certificação completando um curso hoje!

W3schools

1999-2021

2021-11-15

iniciar

JOGO DE CÓDIGOS

Jogar um jogo