

# Função JavaScript Aplicar

[< Anterior](#)[Próximo >](#)

## Método de Reutilização

Com o `apply()` método, você pode escrever um método que pode ser usado em diferentes objetos.

## O método JavaScript apply ()

O `apply()` método é semelhante ao `call()` método (capítulo anterior).

Neste exemplo, a `fullName` método da `pessoa` é **aplicada** sobre `pessoa1` :

### Exemplo

```
const person = {
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}

const person1 = {
  firstName: "Mary",
  lastName: "Doe"
}

// This will return "Mary Doe":
person.fullName.apply(person1);
```

[Tente você mesmo "](#)

## The Difference Between call() and apply()

The difference is:

The `call()` method takes arguments **separately**.

The `apply()` method takes arguments as an **array**.

The `apply()` method is very handy if you want to use an array instead of an argument list.

## The apply() Method with Arguments

The `apply()` method accepts arguments in an array:

### Example

```
const person = {
  fullName: function(city, country) {
    return this.firstName + " " + this.lastName + "," + city + "," + country;
  }
}

const person1 = {
  firstName: "John",
  lastName: "Doe"
}

person.fullName.apply(person1, ["Oslo", "Norway"]);
```

[Try it Yourself »](#)

Compared with the `call()` method:

### Example

```
const person = {
  fullName: function(city, country) {
    return this.firstName + " " + this.lastName + "," + city + "," + country;
  }
}

const person1 = {
  firstName: "John",
  lastName: "Doe"
}

person.fullName.call(person1, "Oslo", "Norway");
```

[Try it Yourself »](#)

## Simulate a Max Method on Arrays

You can find the largest number (in a list of numbers) using the `Math.max()` method:

### Example

```
Math.max(1,2,3); // Will return 3
```

[Try it Yourself »](#)

Since JavaScript **arrays** do not have a `max()` method, you can apply the `Math.max()` method instead.

### Example

```
Math.max.apply(null, [1,2,3]); // Will also return 3
```

[Try it Yourself »](#)

The first argument (null) does not matter. It is not used in this example.

These examples will give the same result:

### Example

```
Math.max.apply(Math, [1,2,3]); // Will also return 3
```

[Try it Yourself »](#)

### Example

```
Math.max.apply(" ", [1,2,3]); // Will also return 3
```

[Try it Yourself »](#)

### Example

```
Math.max.apply(0, [1,2,3]); // Will also return 3
```

[Try it Yourself »](#)

## Modo estrito de JavaScript

No modo estrito do JavaScript, se o primeiro argumento do `apply()` método não for um objeto, ele se tornará o proprietário (objeto) da função invocada. No modo "não estrito", ele se torna o objeto global.

[< Anterior](#)[Próximo >](#)

COLOR PICKER



COMO NÓS



Obtenha a  
certificação  
completando  
um curso hoje!



iniciar

JOGO DE  
CÓDIGOS



Jogar um jogo

Reportar erro

Fórum

Cerca de

Comprar

### Principais tutoriais

Tutorial HTML  
Tutorial CSS  
JavaScript  
Como fazer Tutorial  
SQL Tutorial  
Python Tutorial  
W3.CSS Tutorial  
Bootstrap Tutorial  
PHP Tutorial  
Java Tutorial  
C ++ Tutorial  
jQuery Tutorial

### Referências principais

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

Curso de HTML  
Curso de CSS  
Curso de JavaScript  
Curso de front end  
Curso de SQL  
Curso de Python  
Curso de PHP  
Curso de jQuery  
Curso de Java  
Curso de C ++  
Curso de C #  
Curso de XML

Obter certificação "