

Call

```
const pessoa = {
  nome: 'Miguel',
};

const animal = {
  nome: 'Murphy',
};

function getSomething() {
  console.log(this.nome);
}

getSomething.call(pessoa);
```

```
Stephany@DESKTOP-KTK90HF MINGW64 /c/DIO (main)
$ node playground.js
Miguel
```

```
const pessoa = {
  nome: 'Miguel',
};

const animal = {
  nome: 'Murphy',
};

function getSomething() {
  console.log(this.nome);
}

getSomething.call(animal);
```

```
Stephany@DESKTOP-KTK90HF MINGW64 /c/DIO (main)
$ node playground.js
Murphy
```

```
const myObj = {
  num1: 2,
  num2: 4,
};

function soma(a, b) {
  console.log(this.num1 + this.num2 + a + b);
}

soma.call(myObj, 1, 5);
// 12
```

É possível passar
parâmetros para essa
função separando-os
por vírgulas.

Apply

```
const pessoa = {
  nome: 'Miguel',
};

const animal = {
  nome: 'Godi',
};

function getSomething() {
  console.log(this.nome);
}

getSomething.apply(animal);
```

```
Stephany@DESKTOP-KTK90HF MINGW64 /c/DIO (main)
$ node playground.js
Godi
```

```
const myObj = {
  num1: 2,
  num2: 4,
};

function soma(a, b) {
  console.log(this.num1 + this.num2 + a + b);
}

soma.apply(myObj, [1, 5]);
// 12
```

É possível passar
parâmetros para essa
função dentro de um
array.

Bind

Clona a estrutura da função onde é chamada e aplica o valor do objeto passado como parâmetro.

```
const retornaNomes = function () {  
  return this.nome;  
};  
  
let bruno = retornaNomes.bind({ nome: 'Bruno' });  
  
bruno();  
// Bruno
```