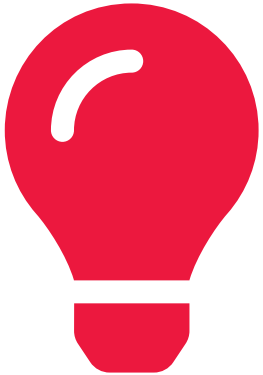
Callbacks

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Uma **callback é uma função** que é passada como parâmetro para outra função.
A função que a recebe é responsável por **executá-la** quando necessário.



Callbacks anônimas

Nesse caso, a função que passamos como **callback** não tem nome, ou seja, é uma função anônima.

Como **funções anônimas** não podem ser chamadas pelo nome, devemos escrevê-las dentro da função que se encarregará de chamar o callback.

```
setTimeout( function(){
    console.log('Olá Mundo!')
} , 1000)
```

Callbacks definidas

A função que passamos como **callback** pode ser uma função **previamente definida**. Ao passá-la como parâmetro de outra função, vamos nos referir a ela pelo seu nome.

```
{}
let minhaCallback = () => console.log('Olá mundo!');
setTimeout(minhaCallback, 2000);
```

Ao escrever uma função como parâmetro, fazemos isso sem os parênteses para evitar que seja executada. Será a função que o recebe quem se encarrega de a executar.

{código}

```
function nomeCompleto(nome, sobrenome) {
    return nome + ' ' + sobrenome;
};
```

Definimos a função **nomeCompleto()**.

Ele se encarrega de juntar um nome e um sobrenome com um espaço.

Ela nos retorna uma string.

```
function bomDia(nome, sobrenome, callback) {
    return  'Olá, ' + callback(nome, sobrenome);
};


bomDia('Amanda', 'Ferreira', nomeCompleto)
// Olá, Amanda Ferreira
```

{código}

```
function nomeCompleto(nome, sobrenome) {
    return nome + ' ' + sobrenome;
};
```

```
function bomDia(nome, sobrenome, callback) {
    return  'Olá, ' + callback(nome, sobrenome);
};
```

```
bomDia('Amanda', 'Ferreira', nomeCompleto)
// Olá, Amanda Ferreira
```

Definimos a função **bomDia()**.

Ela recebe um **nome**, **sobrenome** e uma **callback**.

A **callback será a função** que iremos executar internamente.

{código}

```
function nomeCompleto(nome, sobrenome) {
    return nome + ' ' + sobrenome;
};


function bomDia(nome, sobrenome, callback) {
    return  'Olá, ' + callback(nome, sobrenome);
};


bomDia('Amanda', 'Ferreira', nomeCompleto)
// Olá, Amanda Ferreira
```

O que queremos retornar é uma **string completa**.

Temos a primeira parte no retorno: **'Olá, (...) '**.

O **(...)** virá do que a **callback** retorna no momento em que é executada.

{código}

```
function nomeCompleto(nome, sobrenome) {
    return nome + ' ' + sobrenome;
};


function bomDia(nome, sobrenome, callback) {
    return  'Olá, ' + callback(nome, sobrenome);
};
```

```
bomDia('Amanda', 'Ferreira', nomeCompleto)
// Olá, Amanda Ferreira
```

Executamos a função **bomDia**, passando os parâmetros (nome, sobrenome e **callback**)

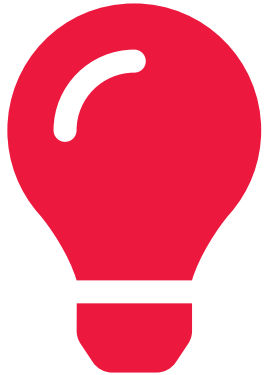
Dentro da função **bomDia** é executada a callback (**função nomeCompleto**)

Após isso será retornada a mensagem completa.

A função **bomDia()** só funciona se passarmos a função **nomeCompleto()** como uma callback?

Não!

Podemos passar **qualquer outra função** que retorne uma string, já que na estrutura interna de bomDia() a definimos para operar com aquele tipo de dado.



{código}

```
function iniciais(nome, sobrenome) {
    return nome[0] + sobrenome[0];
};
```

```
function bomDia(nome, sobrenome, callback) {
    return  'Olá, ' + callback(nome, sobrenome);
};


bomDia('Amanda', 'Ferreira', iniciais)
// Olá, AF
```

Desta vez, quando executamos a função bomDia(), passamos a **função iniciais()** para ela como **uma callback**.
A callback será executada novamente, onde irá retornar as iniciais e **após isso** a função **bomDia retorna o valor**.


DigitalHouse>
Coding School