06

Para saber mais: Objeto literal e referência

Vimos anteriormente como é a estrutura de um objeto, com seus pares de chave e valor:

```
const objPersonagem = {
 nome: "Gandalf",
 classe: "mago",
 nivel: "20"
}
```

COPIAR CÓDIGO

O exemplo acima, assim como o que estamos criando durante esta aula, é o de um **objeto literal**.

Um objeto literal é um objeto criado com a **notação literal**, ou seja: uma lista de chave e valores dentro de chaves `{ }`, que atribuímos a uma variável para que o valor possa ser acessado depois. Exatamente como no exemplo acima.

Objetos literais funcionam bem quando queremos ter um objeto único, com seus próprios dados. Isso porque um objeto literal sempre aponta para um mesmo local na memória, mesmo se você criar cópias dele. Vejamos o código a seguir:

```
const objPersonagem = {
 nome: "Gandalf",
 classe: "mago",
 nivel: "20"
}
```

```
const objPersonagem2 = objPersonagem
```

COPIAR CÓDIGO

Se alterarmos apenas o `objPersonagem2`, o resultado é:

```
const objPersonagem2 = objPersonagem
objPersonagem2.nome = "Gandalf, o Cinzento"

console.log(objPersonagem.nome) //Gandalf, o Cinzento
console.log(objPersonagem2.nome) //Gandalf, o Cinzento
```

COPIAR CÓDIGO

A variável `objPersonagem2` não fez uma cópia do objeto original, apenas serviu como **referência** para o objeto original `objPersonagem`. Assim, qualquer alteração em qualquer um dos objetos altera ambos. Isso porque o JavaScript, quando trabalha com objetos, **acessa os valores deles fazendo referência ao original. mas não cria uma cópia**. Já o acesso por cópia funciona com tipos primitivos (string, number, booleano, `null`, `symbol`):

```
let num = 50
let num2 = num

num2 = 100
console.log(num) //50
console.log(num2) //100
```

COPIAR CÓDIGO

Como podemos contornar esse comportamento quando criamos objetos? Além de utilizar a notação literal, objetos também podem ser criados através do método `Object.create()`:

```
const objPersonagem = {
 nome: "Gandalf",
 classe: "mago",
 nivel: "20"
}

const objPersonagem2 = Object.create(objPersonagem)
objPersonagem2.nome = "Gandalf, o Cinzento"

console.log(objPersonagem.nome) //Gandalf
console.log(objPersonagem2.nome) //Gandalf, o Cinzento
```

COPIAR CÓDIGO

O método `Object.create()` cria um novo objeto **utilizando como protótipo o objeto passado via parâmetro**. Dessa forma, `objPersonagem2` é uma instância diferente de `objPersonagem` e pode ser trabalhada de forma independente.

Você pode ver mais exemplos desse método na documentação do MDN (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Object/create).

Nas próximas aulas veremos também outra forma de criar objetos, utilizando **funções construtoras**.