

Usando o "this" e o "bind"

A palavra-chave `this` (podemos traduzir livremente como "isto") está presente em várias partes do código que rodamos em JavaScript, embora muitas vezes não apareça. Por padrão, ela se refere a um objeto global, porém pode ter diferentes valores, dependendo de onde está sendo utilizada, como os casos:

- Em um método, refere-se ao objeto ao qual o método está ligado (por exemplo, `obj.metodo()`).
- Sozinha ou em uma função, refere-se ao objeto global (no caso do navegador, o objeto `window`).
- Em uma função em `strict mode`, é `undefined`.
- Em um evento, refere-se ao elemento que recebeu o evento.

Quando falamos de objetos (criados com `class` ou não), `this` refere-se ao objeto. Veja esse exemplo com o objeto `pessoa`:

```
function Pessoa(nome, sobrenome) {
  this.nome = nome;
  this.sobrenome = sobrenome;

  this.nomeCompleto = function() {
    console.log(`Meu nome é ${this.nome} ${this.sobrenome}`);
  }
}

const pessoa = new Pessoa("Leia", "Organa");
pessoa.nomeCompleto() // 'Meu nome é Leia Organa'

const outraPessoa = new Pessoa("Luke", "Skywalker");
outraPessoa.nomeCompleto() // 'Meu nome é Luke Skywalker'

const nomeOutraPessoa = pessoa.nomeCompleto.bind(outraPessoa);
// Cria uma nova função onde o valor de `this` se liga ("bind") ao objeto "outraPessoa"
nomeOutraPessoa(); // 'Meu nome é Luke Skywalker'
```

Como feito durante a aula, utilizamos `bind` (em português: unir, ligar, prender) para fazer isso mesmo, "ligar" `this` a um contexto específico para que o `this` não fique "perdido" ao ser invocado fora de seu objeto.

O `this` tem muitos usos e existem outros métodos além de `bind` que podem ser usados com ele, conforme a necessidade. O MDN tem um [artigo bem completo \(https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/this\)](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/this) sobre essa palavra-chave e seus diversos métodos.