

05

Para saber mais: bind(this)

Durante quase todo o processo de construção da lógica do carrossel usamos `this.{alguma coisa}` para criar atributos e fazer chamada de funções. No momento em que vamos colocar uma função para um evento, nós acabamos usando a função `.bind(this)`. Mas por que?

```
this.proximo.addEventListener('click', this.proximoSlide.bind(this))
```

Primeiro precisamos entender quem é `this` no momento em que colocamos a função no `eventListener`. Montando uma função rápida para ver o que vem de retorno:

```
// Arquivo carousel.js
...
this.proximo.addEventListener('click', this.getThis)

getThis() {
  console.log(this) // O botão vai aparecer no console
}
...
```



Então `this` sem a função `bind()` terá o botão como referência, e o botão não terá acesso a todas as funções e atributos da classe que criamos. Vamos ver o que acontece com `this` depois da função `bind()`:

```
// Arquivo carousel.js
...
this.proximo.addEventListener('click', this.getThis.bind(this))

getThis() {
  console.log(this) // A classe Carousel vai aparecer no console
}
...
```



Elements Console Sources Network Performance Memory Application Security Lighthouse

Default levels ▾

1 hidden

Navigated to <http://localhost:8080/>

GET <http://localhost:8080/favicon.ico> 404 (Not Found)

favicon.ico:1
carousel.js:22

```
Carousel {anterior: button.carousel__botao.carousel__botao--anterior, proximo: button.carousel__botao.carousel__botao--proximo, ListaProdutos: ul.carousel__lista-produtos, navegacao: div.carousel__indicadores, slides: Array(5), ...}
```

Agora `this` virou o objeto de `Carousel`, e esse objeto tem acesso a todos os atributos e funções necessários.