

Arrow Functions: deixando o código ainda menos verboso

Transcrição

Podemos ainda deixar o nosso código menos "verboso", usando uma forma diferente de declarar função - utilizada a partir da versão ES2015: falamos das *arrow functions*. O termo *arrow* traduzido para o português significa **flecha**. Com estas funções, podemos eliminar a palavra `function` do código. Mas se simplesmente apagarmos a palavra e tentarmos executar o código, teremos um erro de sintaxe que será apontado no navegador. Para omitirmos o termo, teremos que adicionar `=>` (que representa uma flecha). Ela sinalizará que temos uma *arrow function* que receberá dois parâmetros.

```
adiciona(event) {  
  
    event.preventDefault();  
  
    let data = new Date(...  
        this._inputData.value  
        .split('-')  
        .map((item, indice) => {  
            return item - indice % 2;  
        })  
    );  
    console.log(data);  
}
```

Quando usamos a sintaxe da flecha, já sabemos que se trata de uma *arrow function*. Se testarmos atualizar a página no navegador, veremos que tudo funciona.

A data preenchida no formulário corresponde com a data no Console. Agora, uma pergunta: no bloco da *arrow functions*, quantas instruções nós temos? Apenas **uma**: `return item - indice % 2`. A linha está dentro da função `map()`:

```
.map((item, indice) => {  
    return item - indice % 2;  
});
```

Quando trabalhamos só com uma instrução, podemos omitir o bloco:

```
let data = new Date(...  
    this._inputData.value  
    .split('-')  
    .map((item, indice) => item - indice % 2)  
);
```

Temos que colocar a função com a instrução *inline* e não precisamos colocar o `;`. Como só temos essa instrução, a *arrow function* já irá fazer o retorno e podemos apagar o `return`. O código fica muito mais simples!

Se rodarmos o código no navegador, o formulário está funcionando normalmente e a data aparecerá como desejamos no Console. Nós conseguimos reduzir a "verbosidade" do código, usando uma *arrow function*. Em seguida, criaremos a

variável `negociacao` :

```
let data = new Date(...
  this._inputData.value
  .split('-')
  .map((item, indice) => item - indice % 2)
);

let negociacao = new Negociacao(
  data,
  this._inputQuantidade.value,
  this._inputValor.value
);

console.log(negociacao);
```

Se testarmos no navegador veremos no console a `Negociacao` impressa.