

≡ 04

Iterando com o forEach

Em um arquivo JavaScript, há um array de instrutores da Alura:

```
var nomes = ["Douglas", "Flávio", "Nico", "Rômulo", "Leonardo"];
```

Para percorrer o array e imprimir cada instrutor, deixando claro que eles são instrutores da Alura, podemos utilizar o *loop for*:

```
var nomes = ["Douglas", "Flávio", "Nico", "Rômulo", "Leonardo"];
```

```
for (var i = 0; i < nomes.length; i++) {  
    console.log(nomes[i] + " é instrutor da Alura");  
}
```

Mas vimos nesse capítulo uma outra forma de iterar sobre os elementos de um array, isto é, com o método `forEach()`. Como ficaria a implementação anterior, utilizando o `forEach()`?

A  `forEach (var i = 0; i < nomes.length; i++) {
 console.log(nomes[i] + " é instrutor da Alura");
}`

B  `nomes->forEach(function(nome) {
 console.log(nome + " é instrutor da Alura");
})`

C   `nomes.forEach(function(nome) {
 console.log(nome + " é instrutor da Alura");
})`

Alternativa correta. Em JavaScript, todo array possui a função `forEach`. Passamos para ela uma função por parâmetro, e nessa função fazemos o que quisermos **para cada item do array**. O item do array é recebido por parâmetro na função interna.

D  `forEach (nome : nomes) {
 console.log(nome + " é instrutor da Alura");
}`

Todo array possui a função `forEach()`, em JavaScript. Passamos para ela uma função por parâmetro, e nessa função fazemos o que quisermos **para cada item do array**. O item do array é recebido por parâmetro na função interna:

```
nomes.forEach(function(nome) {  
    console.log(nome + " é instrutor da Alura");  
})
```

PRÓXIMA ATIVIDADE