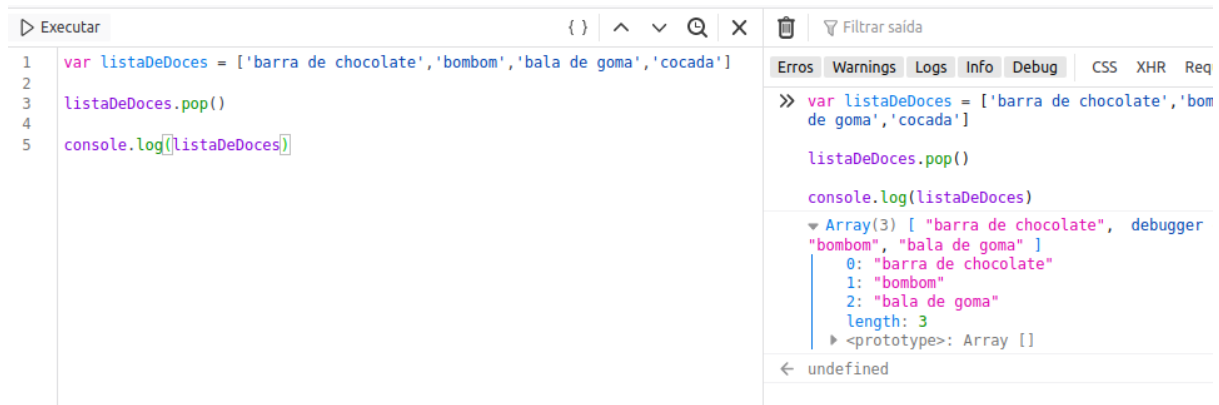


# Exercícios - Aula 7 - Funções Nativas

- 1) Remova o último item da variável:

```
listaDeDoce = ["barra de chocolate", "bombom", "bala de goma", "cocada"]
```

Imprima a nova variável para verificar se o elemento foi removido



```
1 var listaDeDoce = ['barra de chocolate', 'bombom', 'bala de goma', 'cocada']
2
3 listaDeDoce.pop()
4
5 console.log(listaDeDoce)
```

Erros Warnings Logs Info Debug CSS XHR Req

```
>> var listaDeDoce = ['barra de chocolate', 'bom
de goma', 'cocada']

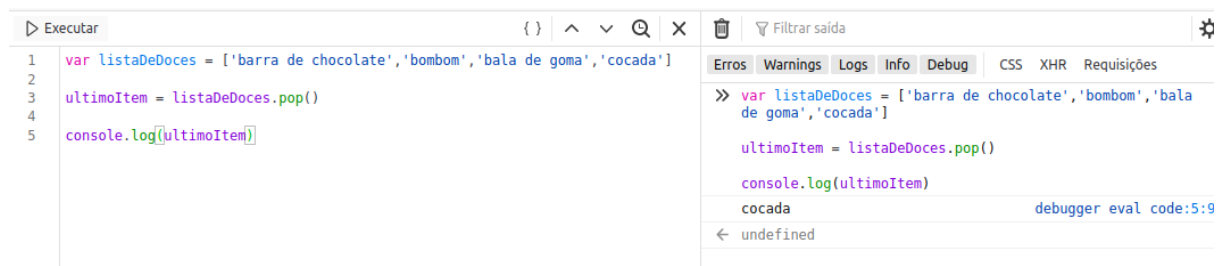
listaDeDoce.pop()

console.log(listaDeDoce)

▼ Array(3) [ "barra de chocolate", debugger
  "bombom", "bala de goma" ]
  0: "barra de chocolate"
  1: "bombom"
  2: "bala de goma"
  length: 3
  ▶ <prototype>: Array []

← undefined
```

- 2) Agora não vamos só remover, você deve guardar o último elemento da variável listaDeDoce em uma nova variável



```
1 var listaDeDoce = ['barra de chocolate', 'bombom', 'bala de goma', 'cocada']
2
3 ultimoItem = listaDeDoce.pop()
4
5 console.log(ultimoItem)
```

Erros Warnings Logs Info Debug CSS XHR Requisições

```
>> var listaDeDoce = ['barra de chocolate', 'bombom', 'bala
de goma', 'cocada']

ultimoItem = listaDeDoce.pop()

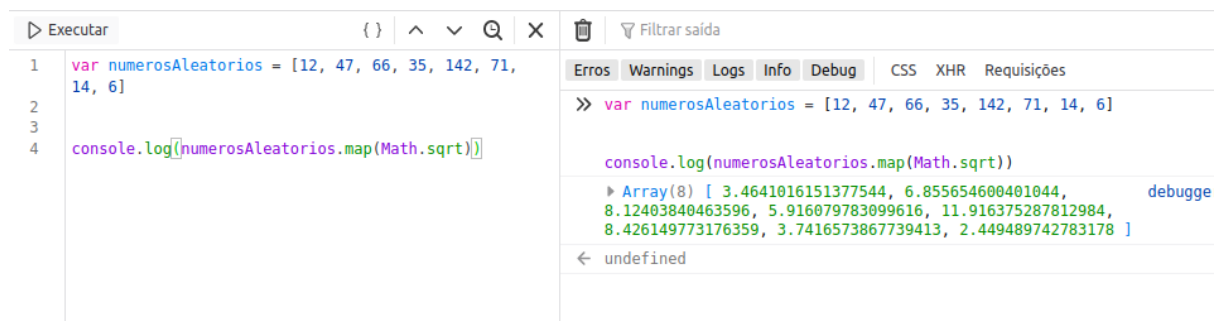
console.log(ultimoItem)

cocada debugger eval code:5:9

← undefined
```

- 3) Utilize a função .map() no array a seguir e retorne a raiz quadrada de cada elemento usando também a função Math.sqrt (exemplo no slide), não esqueça de imprimir na tela para ver o resultado.

```
numerosAleatorios = [12, 47, 66, 35, 142, 71, 14, 6]
```



```
1 var numerosAleatorios = [12, 47, 66, 35, 142, 71, 14, 6]
2
3
4 console.log(numerosAleatorios.map(Math.sqrt))
```

Erros Warnings Logs Info Debug CSS XHR Requisições

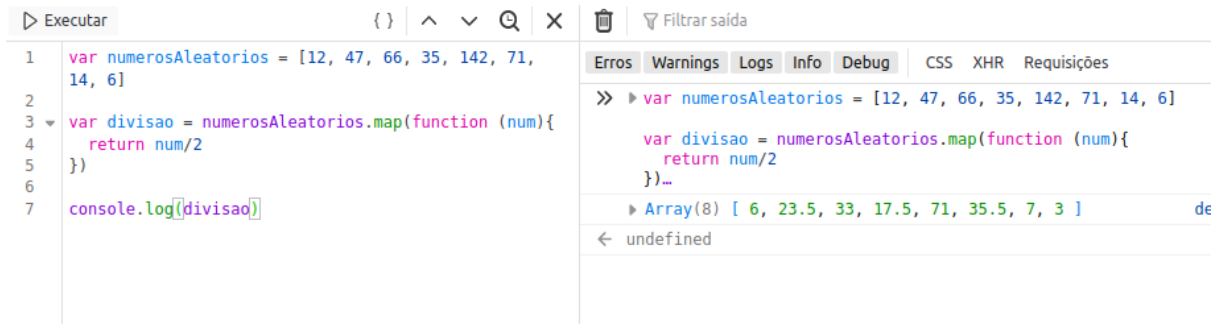
```
>> var numerosAleatorios = [12, 47, 66, 35, 142, 71, 14, 6]

console.log(numerosAleatorios.map(Math.sqrt))

▶ Array(8) [ 3.4641016151377544, 6.855654600401044, debugger
  8.12403840463596, 5.916079783099616, 11.916375287812984,
  8.426149773176359, 3.7416573867739413, 2.449489742783178 ]

← undefined
```

- 4) Utilize a função `.map()` na variável `numerosAleatorios` com o seguinte callback:
- ```
function(num){  
  return num / 2;  
}
```
- Ficou com dúvida? Olha no slide como fizemos

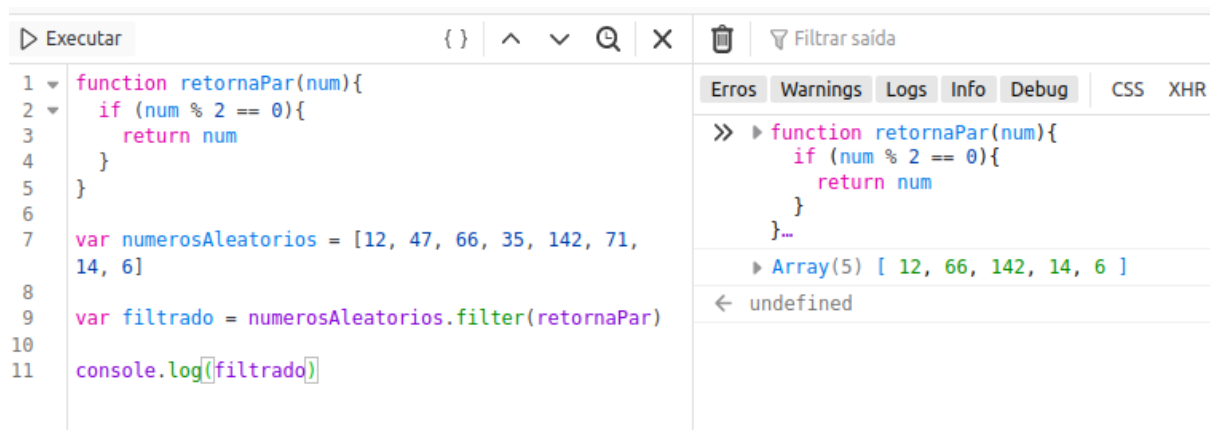


The screenshot shows a code editor with a JavaScript file. The code defines an array `numerosAleatorios` with values `[12, 47, 66, 35, 142, 71, 14, 6]`. It then uses the `.map()` method with a callback function that divides each number by 2. The result is stored in `divisao` and logged to the console.

```
1 var numerosAleatorios = [12, 47, 66, 35, 142, 71, 14, 6]  
2  
3 var divisao = numerosAleatorios.map(function (num){  
4   return num/2  
5 })  
6  
7 console.log(divisao)
```

The right-hand pane shows the console output. The first line is the code being executed. The second line shows the result of the `.map()` operation: `Array(8) [ 6, 23.5, 33, 17.5, 71, 35.5, 7, 3 ]`.

- 5) Crie uma função que retorna os números pares, a seguir, utilize a função `.filter()` na variável `numerosAleatorios` passando a função que você criou como callback. Veja no slide sobre `.filter()` como fizemos



The screenshot shows a code editor with a JavaScript file. It defines a function `retornaPar` that checks if a number is even. Then, it uses the `.filter()` method on the `numerosAleatorios` array with `retornaPar` as the callback. The result is stored in `filtrado` and logged to the console.

```
1 function retornaPar(num){  
2   if (num % 2 == 0){  
3     return num  
4   }  
5 }  
6  
7 var numerosAleatorios = [12, 47, 66, 35, 142, 71, 14, 6]  
8  
9 var filtrado = numerosAleatorios.filter(retornaPar)  
10  
11 console.log(filtrado)
```

The right-hand pane shows the console output. The first line is the code being executed. The second line shows the result of the `.filter()` operation: `Array(5) [ 12, 66, 142, 14, 6 ]`.