

INSTITUTO FEDERAL

São Paulo

Câmpus São Carlos

Aula 03

Arrays e Funções

Prof. Tiago Henrique Trojahn

`tiagotrojahn@ifsp.edu.br`

Objetivo desta aula

Apresentar a variável do tipo Array, um importante recurso presente na linguagem JavaScript, e suas principais funcionalidades

Apresentar as **functions** do JavaScript, um conjunto de instruções que funcionam como subprogramas

Vetores (Arrays) - Criação

```
var vetor = Array();  
vetor[0] = "IFSP";  
vetor[1] = "JavaScript";  
vetor[2] = 2019;  
vetor[3] = true;  
vetor[4] = Array();  
vetor[5] = 1.4567;
```

```
var vetor = Array("IFSP", "JavaScript",  
2019, true, Array(), 1.4567);
```

```
var vetor = ["IFSP", "JavaScript",  
2019, true, Array(), 1.4567];
```

0	1	2	3	4	5
"IFSP"	"JavaScript"	2019	true	Array()	1.4567

Vetores (Arrays) – Manipulação

```
var vetor = ["Banana", "Abacaxi", "Uva"];
```

0

1

2

"Banana"	"Abacaxi"	"Uva"
----------	-----------	-------

Adiciona ao final do
vetor

```
vetor.push("Melão");
```

0

1

2

3

"Banana"	"Abacaxi"	"Uva"	"Melão"
----------	-----------	-------	---------

e retorna...

4

Vetores (Arrays) – Manipulação

```
var vetor = ["Banana", "Abacaxi", "Uva"];
```

0

1

2

"Banana"	"Abacaxi"	"Uva"
----------	-----------	-------

Adiciona ao início do
vetor

```
vetor.unshift("Melão");
```

0

1

2

3

"Melão"	"Banana"	"Abacaxi"	"Uva"
---------	----------	-----------	-------

e retorna...

4

Vetores (Arrays) – Manipulação

```
var vetor = ["Banana", "Abacaxi", "Uva"];
```

0 1 2

"Banana"	"Abacaxi"	"Uva"
----------	-----------	-------

Remove do final do
vetor

`vetor.pop();`

0 1

"Banana"	"Abacaxi"
----------	-----------

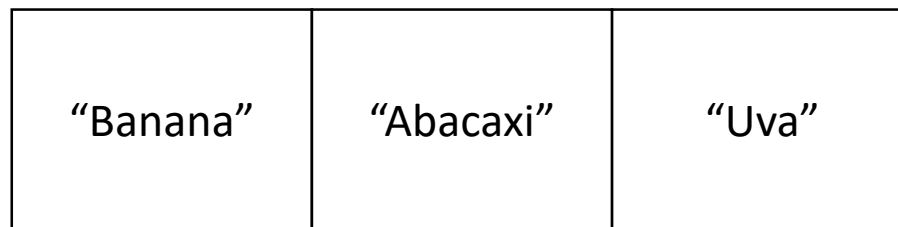
e retorna...

"Uva"

Vetores (Arrays) – Manipulação

```
var vetor = ["Banana", "Abacaxi", "Uva"];
```

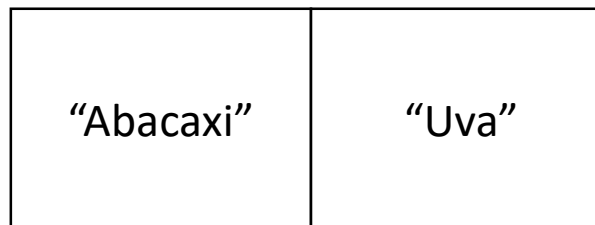
0 1 2



Remove do início do
vetor

`vetor.shift();`

0 1



e retorna...

"Banana"

Exercícios

- Utilizando as funcionalidades vistas até agora, crie um algoritmo que, dado um array de entrada, imprime no console o array revertido.
 - 1 – Alterando o array original
 - 2 – Sem alterar o array original

Atribuição

```
var v1 = [1, 2, 3, 4];  
var v2 = v1;  
console.log(v2); //[1,2,3,4]  
v1.pop();  
console.log(v2); //[1,2,3]
```

Atribuição

- Para evitar a atribuição por referência, deve-se copiar os dados do vetor.
- Para isso, usa-se o **slice**

```
var novoVetor = vetorAntigo.slice();
```

Outros métodos úteis

`vetor.reverse();` //Reverte o vetor

`vetor.sort();` //Ordena o vetor

`vetor.indexOf(valor);` //Retorna o índice do primeiro elemento 'valor', ou -1 se não existir

`vetor.lastIndexOf(valor);` //Retorna o índice do último elemento 'valor', ou -1 se não existir

`vetor.includes(valor);` //Retorna true se 'valor' existir no vetor, ou false caso contrário.

Iteração

- Além de usar o índice, é possível iterar sobre um vetor usando o método **values()**

```
var vetor = ["a", "b", "c", 10];  
  
for (valor of vetor.values()) {  
    console.log(valor); // Saída: "a" "b" "c" "10"  
}
```

Iteração

... ou usando o “in”

```
var vetor = ["a", "b", "c", 10];  
  
for (valor in vetor) {  
    console.log(valor); // Saída: "a" "b" "c" "10"  
}
```

Métodos extras

- Além dos métodos mencionados anteriormente, um *Array* possui ainda diversos outros métodos.
- Alguns dos métodos mais úteis (*forEach*, *every*, *some*) necessitam especificar um *call-back*, representado por uma função que é executada para cada elemento contido no vetor.
- Para mais detalhes, consulte a documentação do Array:

https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array

Funções

- O JavaScript, assim como a maioria das linguagens modernas, suporta a criação de funções.
- Uma função é, resumidamente, um conjunto de instruções que são executadas sistematicamente dado um conjunto de **parâmetros**, podendo ou não **retornar algum valor**. Uma função pode ser vista como um **subprograma**.
 - Funções que não retornam nenhum valor são chamados também de *procedimentos (procedure)*

Funções

- Uma função pode conter uma ampla gama de instruções, tais como:
 - Declaração de variáveis
 - Atribuição, cálculo e operações matemáticas e lógicas sobre as variáveis
 - Laços de repetição e condicionais diversos
 - Especificação e construção de objetos
 - Especificação de funções internas
 - Etc...

- Existem diversas formas de criar uma função. As duas principais formas são:

Funções - Declaração

```
function areaDoCirculo(raio) {  
    return Math.PI * raio**2;  
}
```

```
areaDoCirculo(10.134);
```

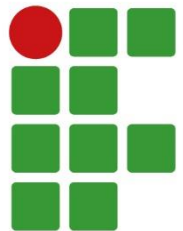
Funções - Expressão

```
var areaDoCirculo = function(raio) {  
    return Math.PI * raio**2;  
};  
areaDoCirculo(10.134);
```

Obrigado!

Prof. Tiago Henrique Trojahn

`tiagotrojahn@ifsp.edu.br`



INSTITUTO FEDERAL

São Paulo

Câmpus São Carlos