



rildo@uespi.br

Tel: 86 98837-2010

Programação Web Responsiva

Variáveis



Variáveis

No Javascript sempre que iniciamos uma variável e inserimos algo nela, a variável se torna do tipo do dados inserido. Vejamos:

```
> var nova = 25  
undefined  
> nova  
25  
> typeof nova  
'number'  
> nova = "rildo"  
'rildo'  
> typeof nova  
'string'  
> 
```

Usando valores no código

O nome digitado poderá ser recebido, pois basta criar uma variável para isso

```
<h1>Página de testes</h1>
<script>
window.alert('Alerta!')
window.confirm('Você é aluno da AESPI?')
window.prompt('Qual é o seu nome ?')
</script>
</body>
```

Usando valores no código

O nome digitado poderá ser recebido, pois basta criar uma variável para isso

```
12  
13   window.confirm('Você é aluno da AESPI?')  
14   var nome = window.prompt('Qual é o seu nome ?')  
15   window.alert('O nome digitado foi ' + nome)  
16   </script>  
17 </body>
```

Usando valores no código

Agora vamos fazer uma soma entre valores em duas variáveis e mostrar em tela

```
<h1>Página de testes</h1>
<script>
var n1 = window.prompt('Digite o primeiro número: ')
var n2 = window.prompt('Digite o segundo número: ')
var s = n1 + n2
window.alert('A soma dos valores é ' + s)
</script>
</body>
</html>
```

Usando valores no código

Os valores dão errados pois o comando `window.prompt` sempre entrega valores em string, então neste caso precisamos converter os dados usando:

Para transformar em numero inteiro

```
Number.parseInt
```

Para transformar em valores reais

```
Number.parseFloat
```

Usando valores no código

Convertendo os dados

```
</head>
<body>
  <h1>Página de testes</h1>
  <script>
var n1 = window.prompt('Digite o primeiro número: ')
var n2 = window.prompt('Digite o segundo número: ')
var s = Number.parseInt(n1) + Number.parseInt(n2)
window.alert('A soma dos valores é ' + s)
  </script>
</body>
```

Como mostrar resultados em tela

Para mostrar em tela devemos chamar :

Document.write

```
9  <body>
10  |   <h1>Página de testes</h1>
11  |   <script>
12  |   var n1 = window.prompt('Digite o primeiro número: ')
13  |   var n2 = window.prompt('Digite o segundo número: ')
14  |   var s = Number.parseInt(n1) + Number.parseInt(n2)
15  |   document.write('A soma dos valores '+ n1 + ' e '+ n2 + 'resultaram em : ' + s)
16  |   </script>
17  </body>
```


Como mostrar resultados em tela

Simplificando o comando

```
document.write('A soma dos valores '+ n1 +' e '+ n2 + 'resultaram em : ' + s)
```

O símbolo deixa de ser aspa e passa a ser crase

```
<h1>Página de testes</h1>
<script>
var n1 = window.prompt('Digite o primeiro número: ')
var n2 = window.prompt('Digite o segundo número: ')
var s = Number.parseInt(n1) + Number.parseInt(n2)
document.write(`A soma dos valores ${n1} e ${n2} resultaram em : ${s}`)
</script>
</body>
```

Como mostrar resultados em tela

Alterando strings

```
11 <script>
12 var nome = window.prompt('Digite o seu nome ')
13 var n1 = window.prompt('Digite o primeiro número: ')
14 var n2 = window.prompt('Digite o segundo número: ')
15 var s = Number.parseInt(n1) + Number.parseInt(n2)
16 document.write(`A soma dos valores ${n1} e ${n2} resultaram em : ${s}<br>`)
17 document.write(`Seu nome em letras maiusculas é ${nome.toUpperCase()}<br>`)
18 document.write(`Seu nome em letras minusculas é ${nome.toLowerCase()}`)
19 </script>
20 </body>
```

Como mostrar resultados em tela

Podemos também chamar como elemento em tela. Isso nós dá uma liberdade maior no que aparece na tela.

```
<p id="teste"></p>
```

```
<script>
```

```
document.getElementById("teste").innerHTML = "Escrevi isso em tela!";
```

```
</script>
```

Como mostrar resultados em tela

Podemos usar as seguintes saídas:

Escrevendo em um elemento HTML, usando `innerHTML`.

Escrevendo na saída HTML usando `document.write()`.

Escrevendo em uma caixa de alerta, usando `window.alert()`.

Escrevendo no console do navegador, usando `console.log()`.

Operadores Aritméticos

Para fazer uma atribuição de valor para uma variável criada devemos usar o símbolo de =

O símbolo = significa atribuição

O símbolo == significa igualdade ampla

O símbolo === significa igualdade estrita

O símbolo != significa diferença

Operadores Aritméticos

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
**	Exponencial
/	Divisão
%	Modulo
++	Incremento
--	Decremento

Operadores Aritméticos

Nome	Operador encurtado	Significado
Atribuição	$x = y$	$x = y$
Atribuição de adição	$x += y$	$x = x + y$
Atribuição de subtração	$x -= y$	$x = x - y$
Atribuição de multiplicação	$x *= y$	$x = x * y$
Atribuição de divisão	$x /= y$	$x = x / y$
Atribuição de resto	$x \% = y$	$x = x \% y$
Atribuição exponencial	$x ** = y$	$x = x ** y$

Operadores Aritméticos

Vamos fazer as seguintes operações no nodejs:

- Variável recebe valor 5
- Adicione 2 a esta variável
- Reduza 3 a esta variável
- Multiplique o valor existente da variável por 2
- Divida o valor existente da variável por 2

Operadores Aritméticos

O que acontece nesta situação?

Var teste= 15 + “rildo”;

Var teste= “rildo” + 15;

Operadores Aritméticos

Quando vamos colocar valores diferentes dentro de um espaço usamos a matriz.

Vetor (**array** uni-dimensional) é uma variável que armazena várias variáveis do mesmo tipo.

Matriz (**array** multi-dimensional) é um **vetor** de **vetores**.

0	1	2	3	4	5

	Matriz B				
	0	1	2	3	4
0	A	E	I	O	U
1	X	A	Y	I	O
2	G	A	S	J	A
3	A	F	D		A
4	T	X	C	Q	E

Tipo array

Os vetores de JavaScript são escritas com colchetes.

Os itens do vetor devem ter descrição:elemento

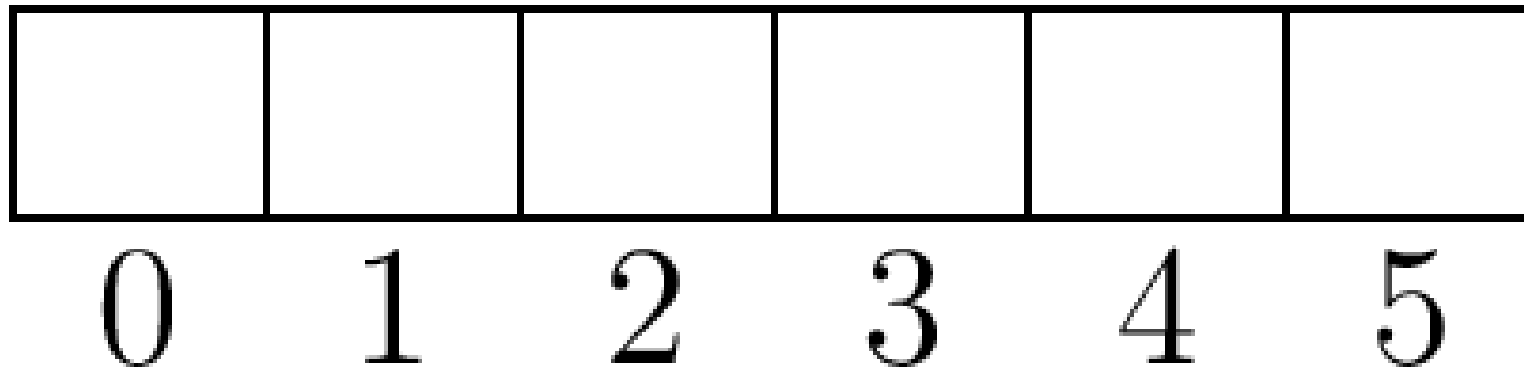
exemplo

```
var a = ["john", "peter", "cassius"]
```

Como adicionar mais pessoas?

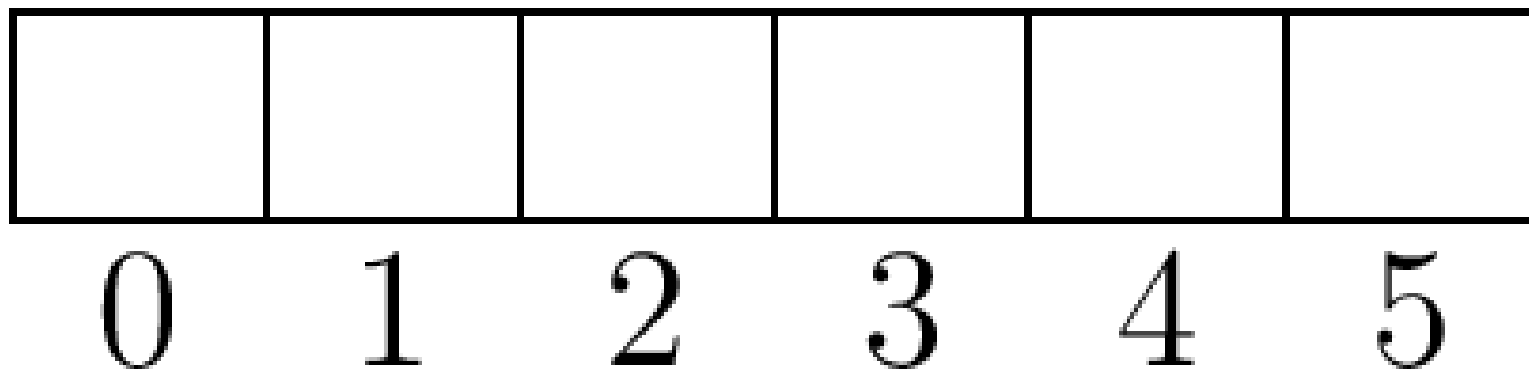
Tipo array

Temos que entender que vetores tem suas posições



$a[0]$, $a[1]$, $a[2]$

Tipo array



Devemos dar a posição adicionar da seguinte forma:

`a[3]="rildo"`

Tipo array

0	1	2	3	4	5

Para remover itens:

Usamos uma nova variável para remover o item
exemplo

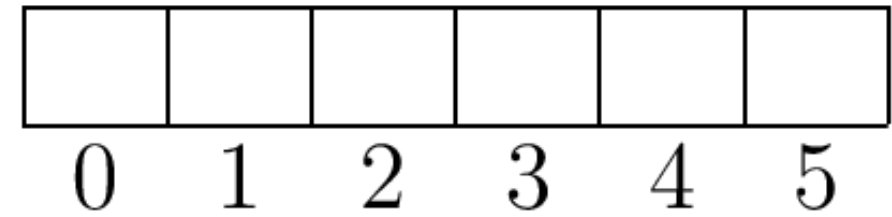
```
var b = a.splice(pos, n);
```

// Isso é como se faz para remover itens, n define o número de
itens a se remover,

// a partir da posição (pos) em direção ao fim da array.

```
> var b = a.splice(3,1);  
undefined
```

Tipo array



Para adicionar um item no final do array usamos o push()

Para adicionar um item no início do array usamos o unshift()

Para remover um item no final do array usamos o pop()

Para remover um item no início do array usamos o shift()

```
> var b = a.splice(3,1);  
undefined  
> b=a.push("kiwi")  
5  
> a  
[ 'jonh', 'peter', 'cassius', 'tony', 'kiwi' ]
```

Matrizes

As matrizes de JavaScript são escritas com colchetes.

Os itens da matriz são separados por vírgulas.

```
var amarelas = ["laranja", "caja"]
```

```
var vermelhas = ["morango", "acerola"]
```

```
var pretas = ["ameixa", "jabuticaba"]
```

```
var frutas=[amarelas,vermelhas,pretas]
```

```
Uncaught SyntaxError: Invalid or unexpected token
> var amarelas = ["laranja","caja"]
undefined
> var vermelhas = ["morango", "acerola"]
undefined
> var pretas = ["ameixa", "jabuticaba"]
undefined
> var frutas=[amarelas,vermelhas,pretas]
undefined
> frutas
[
  [ 'laranja', 'caja' ],
  [ 'morango', 'acerola' ],
  [ 'ameixa', 'jabuticaba' ]
]
>
```


Tipo matriz

Agora adicione umas frutas no vetor amarelo:

Abacaxi

Manga

O que será que acontece quando chamamos a matriz frutas?

Tipo matriz

Agora adicione umas frutas no vetor amarelo:

Abacaxi

Manga

Vamos adicionar usando o push()

```
> var excluir = amarelas.push("abacaxi")
undefined
> var excluir = amarelas.push("manga")
undefined
> amarelas
[ 'laranja', 'caja', 'abacaxi', 'manga' ]
>
```

Tipo matriz

O que será que acontece quando chamamos a matriz frutas?

```
[ 'laranja', 'caja', 'abacaxi', 'manga' ]  
> frutas  
[  
  [ 'laranja', 'caja', 'abacaxi', 'manga' ],  
  [ 'morango', 'acerola' ],  
  [ 'ameixa', 'jabuticaba' ]  
]  
>
```

Booleanos JavaScript

Os booleanos podem ter apenas dois valores: true ou false.

```
]
> var x = 5;
undefined
> var y = 5;
undefined
> var z = 6;
undefined
> (x == y)
true
> (x == z)
false
>
```

Tipo objetc

Os objetos JavaScript são escritos com chaves {}.

As propriedades do objeto são escritas como pares nome: valor, separados por vírgulas.

```
var x = {firstName:"John", lastName:"Doe"};
```

```
var person = {firstName:"John", lastName:"Doe", age:50,  
eyeColor:"blue"};
```

Próxima aula

Condicionais e repetições