

Essa página diz  
Formulário está OK?

Campo obrigatório 1:

Campo obrigatório 2:

# JavaScript

Criação de scripts de automação de páginas WEB

# Unidade Curricular: JavaScript

## CONTEÚDO FORMATIVO

### 1. Criação de scripts de automação de páginas WEB

1. Características da linguagem
2. Introdução a lógica de programação
3. Tag Script
4. Sintaxe Básica
5. Operadores
6. Tipos de dados
7. Comparadores lógicos
8. Blocos condicionais
9. Array ←
10. Blocos de repetição
11. Funções
12. Console do navegador
13. Validação de formulário



#### Capacidades Técnicas

1. Automatizar documentos por meio da linguagem JavaScript
2. Integrar recursos dinâmicos da linguagem JavaScript à páginas HTML

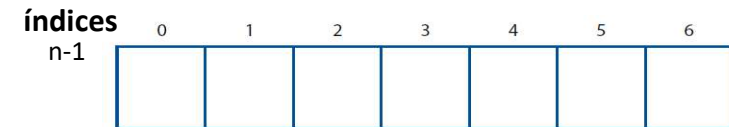
#### Capacidades sociais, organizativas e metodológicas

1. Planejar tarefas no ambiente de trabalho
2. Demonstrar raciocínio lógico
3. Demonstrar iniciativa na busca por novas informações em diferentes meios

# Vetores

→ Estruturas **unidimensionais** que recebem um conjunto de valores.

- ▶ Correspondem a posições da memória:
  - identificadas por um **único nome**
  - individualizadas por **índices**



## Sintaxe no JavaScript

```
[element0, element1, ..., elementN]  
new Array(element0, element1, ..., elementN)  
new Array(arrayLength)
```

Notas:

6,1	2,3	9,4	5,1	8,9	9,8	10	7,0	6,3	4,4
-----	-----	-----	-----	-----	-----	----	-----	-----	-----

Posição: 0 1 2 3 4 5 6 7 8 9

Vetores são estruturas unidimensionais de uma matriz -> Também são definidos como arrays unidimensionais.

# Vetores

Inserir, buscar, excluir e inverter os dados de um vetor

Como trabalhar com os vetores em javascript?

O javascript é orientado à objetos e desta forma todos seus objetos pré-definidos possuem métodos já programados para executar alguma rotina. O array é um exemplo disso.

Para consultar seus métodos pré-existente, podemos digitar no console do navegador o seguinte:

```
var array = [];  
array.__proto__
```

obtendo o seguinte resultado →

```
> var array = [];  
  
array.__proto__  
◀ ▼ [constructor: f, concat: f, copyWithin: f, fill: f, find: f, ...] ⓘ  
  ▶ at: f at()  
  ▶ concat: f concat()  
  ▶ constructor: f Array()  
  ▶ copyWithin: f copyWithin()  
  ▶ entries: f entries()  
  ▶ every: f every()  
  ▶ fill: f fill()  
  ▶ filter: f filter()  
  ▶ find: f find()  
  ▶ findIndex: f findIndex()  
  ▶ findLast: f findLast()  
  ▶ findLastIndex: f findLastIndex()  
  ▶ flat: f flat()  
  ▶ flatMap: f flatMap()  
  ▶ forEach: f forEach()  
  ▶ includes: f includes()  
  ▶ indexOf: f indexOf()  
  ▶ join: f join()  
  ▶ keys: f keys()  
  ▶ lastIndexOf: f lastIndexOf()  
    length: 0  
  ▶ map: f map()  
  ▶ pop: f pop()  
  ▶ push: f push()  
  ▶ reduce: f reduce()  
  ▶ reduceRight: f reduceRight()  
  ▶ reverse: f reverse()  
  ▶ shift: f shift()  
  ▶ slice: f slice()  
  ▶ some: f some()  
  ▶ sort: f sort()  
  ▶ splice: f splice()  
  ▶ toLocaleString: f toLocaleString()  
  ▶ toString: f toString()  
  ▶ unshift: f unshift()  
  ▶ values: f values()  
  ▶ Symbol(Symbol.iterator): f values()  
  ▶ Symbol(Symbol.unscopables): {copyWithin: true, entries: true, fill: true,  
  ▶ [[Prototype]]: Object
```

# Métodos do Array (JavaScript)

**Array.pop()** → O método **pop()** remove o **último** elemento de um array e retorna aquele elemento.

**Array.push()** → O método **push()** adiciona um ou mais elementos ao final de um array e retorna o novo comprimento desse array.

**Array.slice()** → O método **slice()** retorna uma cópia de parte de um array a partir de um subarray criado entre as posições **início** e **fim** (fim não é necessário) de um array original. O Array original não é modificado.

**Array[index]** → Manipula / mostra o valor do conteúdo na posição *index*.

**Arrey.splice()** → O método **splice()** altera o conteúdo de uma lista, adicionando novos elementos enquanto remove elementos antigos.

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array)

[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)

# VARIÁVEIS COMPOSTAS HOMOGÊNEAS

var

n1: inteiro

n2, n3, n4: inteiro

n: vetor[1..4] de inteiro

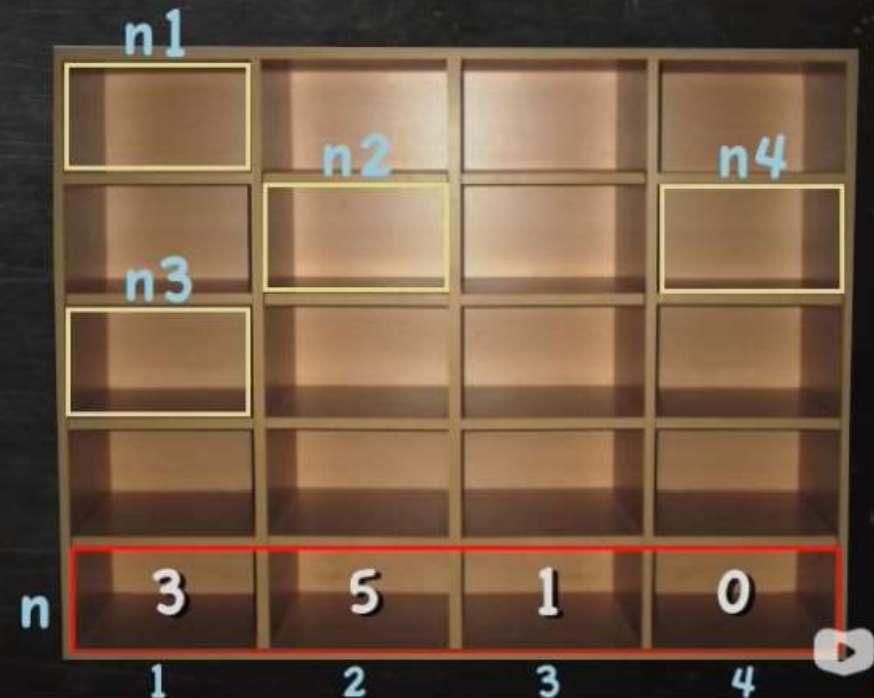
inicio

n[1] <- 3

n[2] <- 5

n[3] <- 1

n[4] <- 0





## Desenhista Web

Faça um algoritmo que receba 11 números armazenando-os em um vetor e depois mostre esses números e suas posições

```
1 Algoritmo "Exemplo_Leitura_Escrita_Vetor"
2
3 Var
4  numeros: vetor[0..10] de inteiro
5  i: inteiro
6
7
8 Inicio
9  Escreval ("Entre com os 11 valores do vetor:")
10 PARA i de 0 ate 10 passo 1 faca
11     LEIA (numeros[i])
12 Fimpara
13
14 Escreval ("Os 11 valores do vetor escolhidos foram:")
15 PARA i de 0 ate 10 passo 1 faca
16     Escreva (" -- ", numeros[i])
17 Fimpara
18
19
20 Fimalgoritmo
```

## Desenhista Web

Faça um algoritmo que receba 11 números armazenando-os em um vetor e depois mostre esses números e suas posições

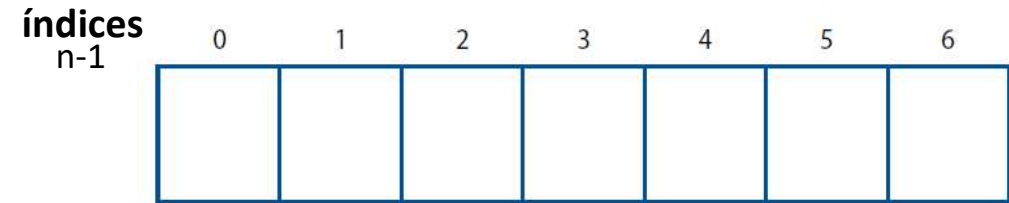
```
1  var numeros = [];  
2  
3  console.log("Entre com os 11 valores do vetor");  
4  
5  for (let i = 0; i <= 10; i++) {  
6      numeros.push(prompt(`Digite o numero na posição ${i}`));  
7  }  
8  
9  for (let i = 0; i <= 10; i++){  
10     console.log(`Número na posição ${i} = ${numeros[i]}`);  
11 }
```



## VETORES *Vamos praticar?*

1- Usando vetores: Calcular a média aritmética de 4 números fornecidos pelo usuário.

2- Imprimir o produto escalar de um número por um vetor de 10 elementos.



3- Ler um vetor Q de 20 posições (aceitar somente números positivos). Escrever a seguir o valor do maior elemento de Q e a respectiva posição que ele ocupa no vetor.

4- Faça um algoritmo para ler um vetor de 15 números. Após isto, ler mais um número qualquer, calcular e escrever quantas vezes esse número aparece no vetor.

## Desenhista Web

# VETORES *Vamos praticar?*

### Respostas

1- Usando vetores: Calcular a média aritmética de 4 números fornecidos pelo usuário.

```
1 algoritmo "Media_Aritmetica"
2
3 var
4 numeros: vetor[1..4] de real
5 media: real
6 i: inteiro
7
8 inicio
9
10 PARA i de 1 ate 4 FACA
11     ESCREVAL("Digite o número ", i, " :")
12     LEIA(numeros[i])
13 FIMPARA
14
15 ESCREVAL(" ")
16 media <- (numeros[1] + numeros[2] + numeros[3] + numeros[4])/4
17 ESCREVAL("Média Aritmética = ", media)
18
19 fimalgoritmo
```

2- Imprimir o produto escalar de um número por um vetor de 10 elementos.

```
1 algoritmo "Produto_escalar"
2
3 var
4 numeros: vetor[1..10] de real
5 multiplica: real
6 i: inteiro
7
8 inicio
9
10 PARA i de 1 ate 10 FACA
11     ESCREVAL("Digite o número ", i, " :")
12     LEIA(numeros[i])
13 FIMPARA
14
15 ESCREVAL(" ")
16 ESCREVAL("Digite o multiplicador :")
17 LEIA(multiplica)
18
19 PARA i de 1 ate 10 FACA
20     numeros[i] <- numeros[i] * multiplica
21 FIMPARA
22
23 ESCREVAL(" ")
24
25 ESCREVAL("O produto escalar do vetor numemeros :")
26 PARA i de 1 ate 10 FACA
27     ESCREVA(numeros[i], " ")
28 FIMPARA
29
30 fimalgoritmo
```



# VETORES *Vamos praticar?*

## *Respostas*

1- Usando vetores: Calcular a média aritmética de 4 números fornecidos pelo usuário.

```
1 var numeros = [];  
2 var media = [];  
3  
4 for (let i = 1; i <= 4; i++){  
5   numeros.push(parseInt(prompt(`Digite o número ${i}:`)));  
6 }  
7  
8 media = (numeros[0] + numeros[1] + numeros[2] + numeros[3]) / 4  
9 console.log("Média aritmética =", media);
```

2- Imprimir o produto escalar de um número por um vetor de 10 elementos.

```
1 var numeros = [];  
2 var multiplica;  
3  
4 for (let i = 1; i <= 10; i++){  
5   numeros[i] = prompt(`Digite o número ${i}:`);  
6 }  
7  
8 multiplica = prompt("Digite o multiplicador");  
9  
10 for (let i = 1; i <= 10; i++){  
11   numeros[i] = numeros[i] * multiplica;  
12 }  
13  
14 console.log("O produto escalar do vetor numeros:");  
15  
16 for (let i = 1; i <= 10; i++){  
17   console.log(numeros[i]);  
18 }
```



## VETORES *Vamos praticar?*

### *Respostas*

3- Ler um vetor Q de 20 posições (aceitar somente números positivos). Escrever a seguir o valor do maior elemento de Q e a respectiva posição que ele ocupa no vetor.

```
1 algoritmo "Maior_Elemento_Positivo"
2
3 var
4 auxiliar: inteiro
5 q: vetor[1..20] de inteiro
6 i, iMaior, maior: inteiro
7
8 inicio
9   maior := 0
10
11   PARA i de 1 ate 20 FACA
12     REPITA
13       ESCREVAL("Digite o número ", i, " :")
14       LEIA(auxiliar)
15
16       SE (auxiliar < 0) ENTAO
17         ESCREVAL("Números Negativos não são aceitos.")
18       FIMSE
19     ATE (auxiliar > 0)
20     q[i] <- auxiliar
21   FIMPARA
22
23   ESCREVAL(" ")
24
25   PARA i de 1 ate 20 FACA
26     SE (maior < q[i]) ENTÃO
27       maior := q[i]
28       iMaior := i
29     FIMSE
30   FIMPARA
31   ESCREVAL("O maior vetor é:: q[,iMaior, "] = ",maior)
32
33 fimalgoritmo
```

## VETORES *Vamos praticar?*

### *Respostas*

3- Ler um vetor Q de 20 posições (aceitar somente números positivos). Escrever a seguir o valor do maior elemento de Q e a respectiva posição que ele ocupa no vetor.

```
1  var auxiliar = 0;
2  var q = [];
3  var iMaior, maior = 0;
4
5  for (let i = 1; i <= 20 ; i++){
6      do {
7          auxiliar = parseInt(prompt(`Digite o número ${i}:`));
8
9          if (auxiliar < 0) {
10             console.log("Números negativos não são aceitos.");
11         }
12     } while (auxiliar < 0);
13     q.push(auxiliar);
14 };
15
16 for (let i = 1; i <= 20 ; i++){
17     if (maior < q[i]){
18         maior = q[i];
19         iMaior = i;
20     }
21 }
22
23 console.log(`O maior número do vetor é o q[${iMaior + 1}] = ${maior}`);
```



## VETORES *Vamos praticar?*

### *Respostas*

4- Faça um algoritmo para ler um vetor de 15 números. Após isto, ler mais um número qualquer, calcular e escrever quantas vezes esse número aparece no vetor.

```
1 algoritmo "Quanto_Repete"
2
3 var
4   guarda: vetor[1..15] de inteiro
5   i, qualquer, repete: inteiro
6
7 inicio
8   repete := 0
9
10  PARA i de 1 ate 15 FACA
11      ESCRIVAL("Digite o número ", i, " :")
12      LEIA(guarda[i])
13  FIMPARA
14
15  ESCRIVAL(" ")
16
17  ESCRIVAL("Digite um outro número qualquer:")
18  LEIA(qualquer)
19
20  ESCRIVAL(" ")
21
22  PARA i de 1 ate 15 FACA
23      SE (qualquer = guarda[i]) ENTÃO
24          repete := repete + 1
25      FIMSE
26  FIMPARA
27
28  ESCRIVA("O número qualquer = ", qualquer)
29  ESCRIVA(" se repete ", repete, " vezes no vetor")
30
31 fimalgoritmo
```

# VETORES

## Inserindo, Buscando, Excluindo e Reorganizando um Registro

- Faça um algoritmo com um vetor de 5 posições que pergunte através de um loop se o usuário deseja inserir um novo valor.
- Informe ao usuário quando o vetor estiver cheio e não for possível inserir um valor novo.
- Dê a opção de visualizar todos os valores inseridos

```
7 Inicio
8   tamanho:=1
9
10  Repita
11    //Opções para o usuário
12    EscrevaL("Digite 1- para inserir um novo")
13    EscrevaL("Digite 2- para ver todos animais")
14    EscrevaL("Digite 3- para sair ")
15    Escreva(">>Resposta = ")
16    leia (resposta)
17    EscrevaL(" ")
18
19    //Inserir valor no vetor
20    se resposta=1 entao
21      se (tamanho<=5) entao
22        Escreva("Digite um animal para inserir no zoologico :")
23        leia (zoo[tamanho])
24        tamanho:=tamanho+1
25      senao
26        EscrevaL("O vetor está cheio. Não é possível inserir um novo animal.")
27      fimSE
28      EscrevaL(" ")
29  fimSe
30
31  //Mostra valores do vetor
32  se resposta=2 entao
33    EscrevaL("Os animais inseridos foram:")
34    para i de 1 ate tamanho-1 passo 1 faca
35      Escreva(zoo[i], " ")
36    fimPara
37    EscrevaL(" ")
38    EscrevaL(" ")
39  fimSE
40  ate (resposta = 3)
41 Fimalgoritmo
```



# VETORES

## Inserindo, Buscando, Excluindo e Reorganizando um Registro

- Faça um algoritmo com um vetor de 5 posições que pergunte através de um loop se o usuário deseja inserir um novo valor.
- Informe ao usuário quando o vetor estiver cheio e não for possível inserir um valor novo.
- Dê a opção de visualizar todos os valores inseridos

- Incremente o programa anterior permitindo a exclusão do ultimo valor inserido.

```
10      Repita
11          //Opções para o usuário
12          EscrevaL("Digite 1- para inserir um novo")
13          EscrevaL("Digite 2- para ver todos animais")
14          EscrevaL("Digite 3- para Excluir o ultimo animail")
15          EscrevaL("Digite 4- para sair ")
16          Escreva(">>Resposta = ")
17          leia (resposta)
18          EscrevaL(" ")
```

O resto do programa se manteve o mesmo, apenas foram modificados o menu, a regra para finalizar o REPITA e a lógica de exclusão que foi adicionada

```
42          //Exclui ultimo valor
43          se resposta=3 entao
44              tamanho:=tamanho-1
45              EscrevaL("Os animal ",zoo[tamanho]," será excluido")
46              zoo[tamanho]:= " "
47              EscrevaL(" ")
48          fimSE
49
50      ate (resposta = 4)
51  Fimalgoritmo
```

## Matrizes

Estrutura de dados **bidimensional**

Conjunto de linhas e colunas → o índice de uma linha junto ao índice de uma coluna aponta para uma determinada célula

Var

<nome da matriz> : vetor[<índice inicial das linhas>..**<índice final das linhas>**,  
<índice inicial das colunas>..**<índice final das colunas>**] de <tipo de dado>

Var

matriz : vetor[0..4, 0..4] de inteiro

Linhas iniciando no índice 0 e finalizando no índice 4

Colunas iniciarão no índice 0 e finalizarão no índice 4

Inserir valores nas células de uma matriz é preciso chamar esta pelo seu identificador e informar o índice da linha e da coluna

matriz[2, 1] := 20

escreval(matriz[2, 1])

matriz[0, 0] := matriz[0, 1] + matriz[0, 2]

multiplicacao := matriz[1, 1] \* matriz[2, 2]

	0	1	2	3	4
0					
1					
2					
3					
4					

matriz 5x5

	Posição do livro				
	0	1	2	...	n-1
0	788	598	265	...	156
1	145	258	369	...	196
2	989	565	345	...	526
...	⋮	⋮	⋮	⋮	⋮
m-1	845	153	564	892	210

$$a_{m,n} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

matriz 3x3

Var

matrizA: vetor[1..4] de inteiro

matrizB: vetor[1..50, 1..4] de inteiro

- a matriz **matrizA** pode ser utilizada para armazenar 4 notas de um aluno;
- a matriz **matrizB** pode ser utilizada para armazenar 4 notas de 50 alunos;

# MAIS DIMENSÕES

var

m: vetor[1..3, 1..2] de inteiro

inicio

m[1,2] <- 4

m[2,2] <- 5

m[3,1] <- 8

	m			
1		4		
2		5		
3	8			
	1	2		

Matrizes - Curso de Algoritmos #15 - Gustavo Guanabara

<https://www.youtube.com/watch?v=hkE9WrjpAAk>

## Desenhista web

# Matrizes

## Exemplos

```
1 Algoritmo "Exemplo_Leia_Escreva_Matriz"
```

```
2  
3 Var  
4 i,j: inteiro  
5 minhaMatriz: vetor[1..2, 1..2] de caracter
```

```
6  
7  
8 Inicio  
9 Escreval ("Entre com os 4 valores da matriz:")
```

```
10 PARA i de 1 ate 2 faca  
11     PARA j de 1 ate 2 faca  
12         ESCREVAL("Posição A",i,j, ":")  
13         LEIA(minhaMatriz[i,j])  
14     Fimpara  
15 Fimpara
```

```
16  
17 ESCREVAL(" ")  
18  
19 Escreval ("Os 4 valores do vetor escolhidos foram:")  
20 PARA i de 1 ate 2 faca  
21     PARA j de 1 ate 2 faca  
22         ESCREVAL("Posição A",i,j, "= ", minhaMatriz[i,j])  
23     Fimpara  
24 Fimpara  
25  
26 Fimalgoritmo
```

```
Entre com os 4 valores da matriz:  
Posição A 1 1:  
Bola  
Posição A 1 2:  
Ball  
Posição A 2 1:  
Cão  
Posição A 2 2:  
Dog  
  
Os 4 valores do vetor escolhidos foram:  
Posição A 1 1= Bola  
Posição A 1 2= Ball  
Posição A 2 1= Cão  
Posição A 2 2= Dog  
  
>>> Fim da execução do programa !
```

Bola	Ball
Cão	Dog

matriz 2x2

```
1 Algoritmo "Exemplo_Monta_Matriz"  
2  
3 Var  
4 i,j: inteiro  
5 minhaMatriz: vetor[0..2, 0..2] de inteiro  
6  
7  
8 Inicio  
9 PARA i de 0 ate 2 faca  
10     PARA j de 0 ate 2 faca  
11         minhaMatriz[i,j]:= i+j  
12     Fimpara  
13 Fimpara  
14  
15 escreval(" -----")  
16 PARA i de 0 ate 2 faca  
17     PARA j de 0 ate 2 faca  
18         ESCREVA(" | ", minhaMatriz[i,j])  
19     Fimpara  
20  
21     ESCREVAL(" |")  
22     escreval(" -----")  
23 Fimpara  
24  
25 Fimalgoritmo
```

	0		1		2	
	1		2		3	
	2		3		4	

matriz 3x3



## Matrizes *Vamos praticar?*

1- Desenvolver um algoritmo que receba os valores de uma matriz de 4 linhas e 4 colunas e mostre quais são os elementos da diagonal principal.

2- Desenvolver um algoritmo que crie duas matrizes 10x10. Leia um valor digitado pelo usuário e com base nele atribua os valores para cada posição de cada matriz:

- $\text{MatrizA}[i,j] = \text{minhavariavel} + 2i + j$
- $\text{MatrizB}[i,j] = \text{minhavariavel} + i - j$

3- Implemente o programa anterior acrescentando uma matriz de soma das duas criadas.

4- Faça um algoritmo que leia 9 valores para uma matriz 3x3. Posteriormente, percorra esta matriz e imprima somente os números que são pares.

5- Faça um algoritmo que possua duas matrizes 4x4 de números inteiros. Posteriormente, imprima somente os números que estão nas duas matrizes.

	0	1	2	3	4
0					
1					
2					
3					
4					



# Matrizes *Vamos praticar?* *Respostas*

1- Desenvolver um algoritmo que receba os valores de uma matriz de 4 linhas e 4 colunas e mostre quais são os elementos da diagonal principal.

```
1 Algoritmo "DiagonalPrincipal"
2
3 Var
4   i,j: inteiro
5   matrizQuatro: vetor[1..4, 1..4] de inteiro
6
7 Inicio
8   Escreval ("Entre com os 16 valores da matriz:")
9   PARA i de 1 ate 4 faca
10     PARA j de 1 ate 4 faca
11       ESCREVAL("Posição A",i,j, ":")
12       LEIA(matrizQuatro[i,j])
13     Fimpara
14   Fimpara
15
16   ESCREVAL(" ")
17
18   Escreval ("Os valores da diagonal principal são:")
19   PARA i de 1 ate 4 faca
20     PARA j de 1 ate 4 faca
21       SE (i = j) entao
22         ESCREVAL("Posição A",i,j, " = ", matrizQuatro[i,j])
23       FIMSE
24     FIMPARA
25   FIMPARA
26
27 Fimalgoritmo
```

# Matrizes *Vamos praticar?*

## *Respostas*

**2-** Desenvolver um algoritmo que crie duas matrizes 10x10. Leia um valor digitado pelo usuário e com base nele atribua os valores para cada posição de cada matriz:

- $\text{MatrizA}[i,j] = \text{minhavariavel} + 2i + j$
- $\text{MatrizB}[i,j] = \text{minhavariavel} + i - j$


**3-** Implemente o programa anterior acrescentando uma matriz de soma das duas criadas.

```
1 Algoritmo "SomaMatriz"
2
3 Var
4   i,j: inteiro
5   minhavariavel: inteiro
6   matrizA: vetor[1..10,1..10] de inteiro
7   matrizB: vetor[1..10,1..10] de inteiro
8
9 Inicio
10  ESCRIVAL ("Digite a variável que criará as matrizes")
11  LEIA(minhavariavel)
12
13  ESCRIVAL (" ")
14
15  ESCRIVAL ("Matriz A:")
16    PARA i de 1 ate 10 faca
17      PARA j de 1 ate 10 faca
18        matrizA[i,j] := minhavariavel + 2*i + j
19        ESCRIVA("  |", matrizA[i,j])
20      FIMPARA
21    ESCRIVA("  |")
22    ESCRIVAL(" ")
23  FIMPARA
24
25  ESCRIVAL (" ")
26
27  ESCRIVAL ("Matriz B:")
28    PARA i de 1 ate 10 faca
29      PARA j de 1 ate 10 faca
30        matrizB[i,j] := minhavariavel + i - j
31        ESCRIVA("  |", matrizB[i,j])
32      FIMPARA
33    ESCRIVA("  |")
34    ESCRIVAL(" ")
35  FIMPARA
36
37 Fimalgoritmo
```




# Matrizes *Vamos praticar?* *Respostas*

```
1 Algoritmo "SomaMatriz"
2
3 Var
4   i,j: inteiro
5   minhavariavel: inteiro
6   matrizA: vetor[1..10,1..10] de inteiro
7   matrizB: vetor[1..10,1..10] de inteiro
8   matrizSoma: vetor[1..10,1..10] de inteiro
9
10 Inicio
11   ESCREVAL ("Digite a variável que criará as matrizes")
12   LEIA(minhavariavel)
13
14   ESCREVAL (" ")
15
16   ESCREVAL ("Matriz A:")
17   PARA i de 1 ate 10  faça
18     PARA j de 1 ate 10  faça
19       matrizA[i,j]:= minhavariavel + 2*i + j
20       ESCREVA("  |",  matrizA[i,j])
21     FIMPARA
22   ESCREVA("  |")
23   ESCREVAL (" ")
24 FIMPARA
25
26 ESCREVAL (" ")
```



**3-** Implemente o programa anterior acrescentando uma matriz de soma das duas criadas.

```
27
28 ESCREVAL ("Matriz B:")
29   PARA i de 1 ate 10  faça
30     PARA j de 1 ate 10  faça
31       matrizB[i,j]:= minhavariavel + i - j
32       ESCREVA("  |",  matrizB[i,j])
33     FIMPARA
34   ESCREVA("  |")
35   ESCREVAL (" ")
36 FIMPARA
37
38 ESCREVAL (" ")
39
40 ESCREVAL ("Matriz SOMA:")
41   PARA i de 1 ate 10  faça
42     PARA j de 1 ate 10  faça
43       matrizSoma[i,j]:= matrizA[i,j]+ matrizB[i,j]
44       ESCREVA("  |",  matrizSoma[i,j])
45     FIMPARA
46   ESCREVA("  |")
47   ESCREVAL (" ")
48 FIMPARA
49
50 Fimalgoritmo
```



# Matrizes *Vamos praticar?* *Respostas*

4- Faça um algoritmo que leia 9 valores para uma matriz 3x3. Posteriormente, percorra esta matriz e imprima somente os números que são pares.

```
1 Algoritmo "MatrizPares"
2
3 Var
4   i,j: inteiro
5   matrizNove: vetor[1..3,1..3] de inteiro
6
7 Inicio
8   ESCREVAL ("Digite os valores da matriz:")
9   PARA i de 1 ate 3 faca
10     PARA j de 1 ate 3 faca
11       ESCREVAL ("MATRIZ[",i,",",j,"] = " )
12       LEIA (matrizNove[i,j])
13     FIMPARA
14   FIMPARA
15
16   ESCREVAL (" ")
17
18   ESCREVAL ("Os números pares da matriz são:")
19   PARA i de 1 ate 3 faca
20     PARA j de 1 ate 3 faca
21       SE (matrizNove[i,j] mod 2 = 0) entao
22         ESCREVAL ("MATRIZ[",i,",",j,"] = ",matrizNove[i,j])
23       FIMSE
24     FIMPARA
25   FIMPARA
26
27 Fimalgoritmo
```

# Matrizes *Vamos praticar?*

## *Respostas*

5- Faça um algoritmo que possua duas matrizes 4x4 de números inteiros. Posteriormente, imprima somente os números que estão nas duas matrizes.

```
28
29 ESCREVAL ("Os números que estão nas duas matrizes:")
30
31 PARA m de 1 ate 4 faca
32   PARA n de 1 ate 4 faca
33     PARA i de 1 ate 4 faca
34       PARA j de 1 ate 4 faca
35         SE (matrizA[m,n] = matrizB[i,j]) entao
36           ESCREVAL ("MATRIZ[",i,"",j,"] = ",matrizA[i,j])
37         FIMSE
38       FIMPARA
39     FIMPARA
40   FIMPARA
41 FIMPARA
42
43 Fimalgoritmo
```

```
1 Algoritmo "MatrizPares"
2
3 Var
4   i,j,m,n: inteiro
5   matrizA: vetor[1..4,1..4] de inteiro
6   matrizB: vetor[1..4,1..4] de inteiro
7
8 Inicio
9   ESCREVAL ("Digite os valores da Matriz A:")
10  PARA i de 1 ate 4 faca
11    PARA j de 1 ate 4 faca
12      ESCREVAL ("MATRIZ[",i,"",j,"] = " )
13      LEIA(matrizA[i,j])
14    FIMPARA
15  FIMPARA
16
17 ESCREVAL (" ")
18
19 ESCREVAL ("Digite os valores da Matriz B:")
20  PARA i de 1 ate 4 faca
21    PARA j de 1 ate 4 faca
22      ESCREVAL ("MATRIZ[",i,"",j,"] = " )
23      LEIA(matrizB[i,j])
24    FIMPARA
25  FIMPARA
26
27 ESCREVAL (" ")
```

# Métodos do Array (JavaScript)

**Array.find()**



O método `find()` retorna o valor do primeiro elemento do array que satisfizer a função de teste provida. Caso contrario, `undefined` é retornado.

**Array.forEach()**



O método `forEach()` executa uma dada função em cada elemento de um array.

**Array.includes()**



O método `includes()` determina se um array contém um determinado elemento, retornando `true` ou `false` apropriadamente.

**Array.reverse()**



O método `reverse()` inverte os itens de um array. O primeiro elemento do array se torna o último e o último torna-se o primeiro.

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array)

[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)