

Objetivo:

- I. Tipo de dado Array;
- II. Array bidimensional;
- III. Percorrer os elementos do array.

Observação: Para fazer os exemplos recomenda-se o uso do VS Code ou da interface de programação online <https://replit.com/>.

I. Tipo de dado Array

Uma variável pode receber somente um valor por vez, mas um array pode manter um conjunto de valores.

Um array é representado por um conjunto de valores separados por vírgulas entre um par de colchetes, cada elemento do array pode ser de um tipo de dado distinto. No exemplo a seguir cada elemento possui um tipo de dado distinto:

```
vetor = ["Boa noite", 20, 2.5, true]
```

Cada elemento de um array possui uma posição e precisamos dessa posição para acessar os elementos do array.

O 1º elemento está na posição zero. No exemplo a seguir acessamos cada posição do array para leitura:

```
vetor = ["Boa noite", 20, 2.5, true]
console.log("1o:", vetor[0])
console.log("2o:", vetor[1])
console.log("3o:", vetor[2])
console.log("4o:", vetor[3])
```

```
1o: Boa noite
2o: 20
3o: 2.5
4o: true
```

Os elementos do array podem ser acessados para leitura e escrita. No exemplo a seguir a 3ª posição do array foi acessada para escrita:

```
vetor = ["Boa noite", 20, 2.5, true]
vetor[2] = "terceira"
console.log("1o:", vetor[0])
console.log("2o:", vetor[1])
console.log("3o:", vetor[2])
console.log("4o:", vetor[3])
```

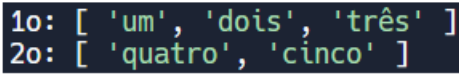
```
1o: Boa noite
2o: 20
3o: terceira
4o: true
```

II. Array bidimensional

Cada elemento do array pode ser de qualquer tipo de dado, inclusive pode ser outro array. No exemplo a seguir o array possui 2 elementos, sendo que cada elemento é outro array:

```
matriz = [
  ["um", "dois", "três"],
  ["quatro", "cinco"]
]

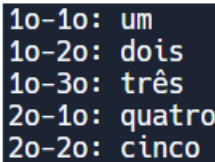
console.log("1o:", matriz[0])
console.log("2o:", matriz[1])
```



Para acessar os elementos da lista interna precisaremos de dois índices, um para acessar a posição no 1º nível e outro índice para o 2º nível:

```
matriz = [
  ["um", "dois", "três"],
  ["quatro", "cinco"]
]

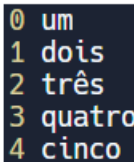
console.log("1o-1o:", matriz[0][0])
console.log("1o-2o:", matriz[0][1])
console.log("1o-3o:", matriz[0][2])
console.log("2o-1o:", matriz[1][0])
console.log("2o-2o:", matriz[1][1])
```



III. Percorrer os elementos do array

Como as posições dos elementos do array começam em zero e são sequenciais, então podemos utilizar uma estrutura de repetição while para prover os índices das posições. No exemplo a seguir a variável **pos** receberá os valores 0, 1, 2 e 3. Veja que utilizamos os valores da variável **pos** como índice de posição no **vetor**:

```
vetor = ["um", "dois", "três", "quatro", "cinco"]
pos = 0
while( pos < vetor.length ){
  console.log(pos, vetor[pos])
  pos = pos + 1
}
```



A propriedade **length** possui a quantidade de elementos do array que está na variável **vetor**.

Precisaremos de dois índices para acessar os elementos de um array bidimensional. No exemplo a seguir as variáveis **linha** e **coluna** são usadas para receber os índices dos arrays. Sendo que a variável **linha** terá os índices do array externo e a variável **coluna** terá os índices do array interno.

Aqui criamos arrays de uma e duas dimensões, porém um array pode ter inúmeras dimensões.

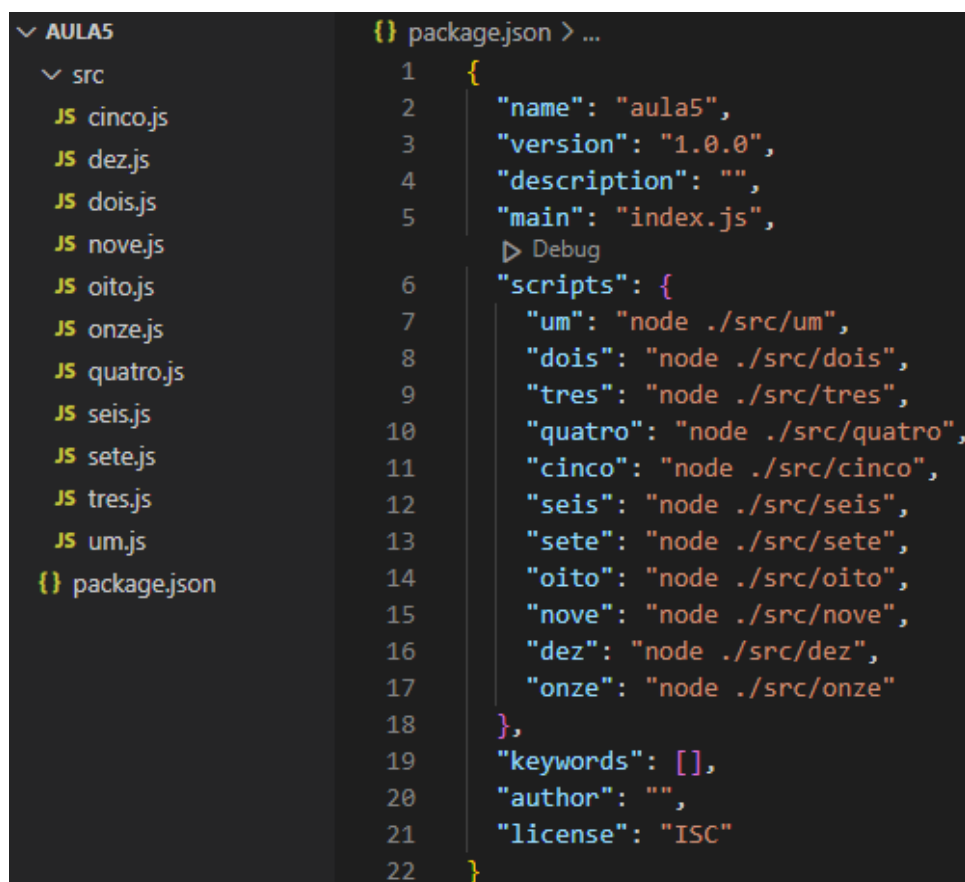
```
matriz = [
  ["um","dois","três"],
  ["quatro","cinco"]
]
linha = 0
while( linha < matriz.length ){
  coluna = 0
  while( coluna < matriz[linha].length ){
    console.log("linha:",linha,"coluna:",coluna,":",matriz[linha][coluna])
    coluna = coluna + 1
  }
  linha = linha + 1
}
```

```
linha: 0 coluna: 0 : um
linha: 0 coluna: 1 : dois
linha: 0 coluna: 2 : três
linha: 1 coluna: 0 : quatro
linha: 1 coluna: 1 : cinco
```

Exercícios

Veja o vídeo se tiver dúvidas nos exercícios: <https://youtu.be/A3Yuj3FQdSE>

Para fazer os exercícios crie um projeto de nome **aula5** no VS Code assim como é mostrado a seguir. Cada programa deverá estar num arquivo separado da pasta **src**. Crie uma propriedade para cada exercício na propriedade **scripts** do arquivo **package.json**. Para rodar o arquivo use: **npm run propriedade**, onde propriedade será **um**, **dois**, **tres** (sem acento), **quatro**, ... e **onze**.



Exercício 1: Completar o código a seguir para imprimir na tela o primeiro elemento do array.

```
numeros = ["um", "dois", "três", "quatro"]
```

Exemplo de saída:

```
Primeiro um
```

Exercício 2: Completar o código a seguir para imprimir na tela o último elemento do array.

```
numeros = ["um", "dois", "três", "quatro"]
```

Exemplo de saída:

```
Último quatro
```

Exercício 3: Completar o código a seguir para imprimir na tela o os elementos do array na ordem invertida.

Dica: use a estrutura de repetição while.

```
numeros = ["um", "dois", "três", "quatro"]
```

Exemplo de saída:

```
3 quatro
2 três
1 dois
0 um
```

Exercício 4: Completar o código a seguir para imprimir na tela o somatório dos elementos do array.

Dica: use a estrutura de repetição while.

```
numeros = [8, 3, 4, 7, 5]
```

Exemplo de saída:

```
Somatório 27
```

Exercício 5: Completar o código a seguir para imprimir na tela os elementos do array que possui valor ímpar.

Dica: use a estrutura de repetição while.

```
numeros = [8, 3, 4, 7, 5, 6, 4]
```

Exemplo de saída:

```
Ímpar 3
Ímpar 7
Ímpar 5
```

Exercício 6: Completar o código a seguir para imprimir na tela os elementos do array que estão em posições ímpares.

Dica: use a estrutura de repetição while.

```
numeros = [8, 3, 4, 7, 5, 6, 4]
```

Exemplo de saída:

```
1 : 3
3 : 7
5 : 6
```

Exercício 7: Completar o código a seguir para imprimir na tela a multiplicação dos elementos dos arrays pesos e notas.

Dica: use a estrutura de repetição while.

```
pesos = [0.4, 0.2, 0.1, 0.3]
```

```
notas = [8.2, 5.0, 10.0, 9.1]
```

Exemplo de saída:

```
0.4 * 8.2 = 3.28
0.2 * 5 = 1
0.1 * 10 = 1
0.3 * 9.1 = 2.73
```

Exercício 8: Alterar o Exercício 7 para imprimir na tela o somatório das multiplicações.

Dica: use a estrutura de repetição while.

Exemplo de saída:

```
Nota final 8.01
```

Exercício 9: Completar o código a seguir para imprimir na tela os elementos de maior valor em cada posição nos arrays de entradas e saídas.

Dica: use a estrutura de repetição while.

```
entradas = [9,2,4,5,8]
```

```
saidas = [3,5,2,7,4]
```

Exemplo de saída:

```
0 : 9
1 : 5
2 : 4
3 : 7
4 : 8
```

Exercício 10: Completar o código a seguir para imprimir na tela os elementos da diagonal principal da matriz.

Dica: use a estrutura de repetição while.

```
matriz = [
    [9,2,4],
    [6,5,7],
    [2,1,3]
]
```

Exemplo de saída:

```
9
5
3
```

Exercício 11: Completar o código a seguir para imprimir na tela o somatório dos elementos da matriz.

Dica: use a estrutura de repetição while.

```
matriz = [
    [9,2,4],
    [6,5,7],
    [2,1,3]
]
```

Exemplo de saída:

```
Somatório: 39
```