

# Algoritmos e Programação : Arrays

Prof. Ovídio Francisco



## Plano de aula

Conceito de arrays

Índices

Iteração em arrays

Exemplos

Exercícios

# Introdução

# O Contexto

Praticamente todo software utiliza a memória para armazenar dados na forma de variáveis. É muito comum também um programa precise lidar com um conjunto de dados de maneira organizada e prática. Por exemplo:

- Lista de nomes
- Notas de turma uma de alunos
- Salários dos funcionários
- Pontuações do jogadores
- Vendas de um período

## O Contexto

Nesse sentido, as linguagens de programação fornecem recursos para armazenar múltiplos dados em uma mesma estrutura.

Os arrays (ou vetores) são conjuntos de elementos homogêneos e estruturados.

## O Contexto

Enquanto uma variável simples armazena um único valor como em:

```
num    = 7;  
letra  = 'A';
```

## O Contexto

Enquanto uma variável simples armazena um único valor como em:

```
num    = 7;  
letra  = 'A';
```

Um **array** pode ser visto como uma coleção dados do **mesmo tipo**.

Por exemplo, para armazenar todos os produtos de uma loja, podemos guardá-los em um **array**.

## O Contexto

Considere a necessidade de armazenar uma nota de um aluno. Nesse caso, podemos declarar uma variável como mostrado abaixo:

```
nota = 10;
```

nota

10

Essa variável pode ser representada assim:

## O Contexto

Considere agora a necessidade de armazenar as notas de 10 alunos. Nesse caso, será necessário **mais espaço** na memória.

```
notas = [];
```

```
notas[0] = 9;  
notas[1] = 7;  
notas[2] = 8;  
notas[3] = 2;  
notas[4] = 0;  
notas[5] = 3;  
notas[6] = 1;  
notas[7] = 4;  
notas[8] = 3;  
notas[9] = 5;
```

notas

9	7	8	2	0	3	1	4	3	5
---	---	---	---	---	---	---	---	---	---

Um array pode ser representada assim:



## Criando arrays em C

## Criando arrays em C

Para criar um array, assim como as variáveis simples, devemos declará-lo.

- Após o nome temos um par de colchetes. São eles que determinam que trata-se de um array.

```
dados = [];
```

# Criando arrays em C

Para criar um array, assim como as variáveis simples, devemos declará-lo.

- Após o nome temos um par de colchetes. São eles que determinam que trata-se de um array.

`dados = [];`

Colchetes

Nome do array

## Criando arrays em C

Outros exemplos são:

```
scores      = [];  
salarios    = [];  
alturas     = [];  
pesos       = [];  
populacoes = [];
```

## Acessando elementos do array

# Criando arrays em C

Os elementos de um array podem ser acessado por meio de **índices**.

- Um índice é um **número** inteiro que indica qual a **posição** de um elemento.
- Um elemento é apontado na forma **array[índice]**.
- O primeiro elemento é indicado por **0**.
- O último elemento é indicado por **tamanho-1**.

Exemplos:

```
notas = [];
```

```
notas[0] = 9;
```

```
notas[1] = 7;
```

```
notas[2] = 8;
```

```
notas[3] = 2;
```

# Criando arrays em C

Os elementos de um array podem ser acessado por meio de **índices**.

- Um índice é um **número** inteiro que indica qual a **posição** de um elemento.
- Um elemento é apontado na forma **array[índice]**.
- O primeiro elemento é indicado por **0**.
- O último elemento é indicado por **tamanho-1**.

Exemplos:

```
notas = [];
```

```
notas[0] = 9;
```

```
notas[1] = 7;
```

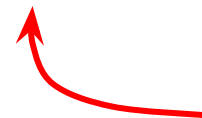
```
notas[2] = 8;
```

```
notas[3] = 2;
```

Índices



Valores



# Criando arrays em C

Internamente, os dados se organizam linearmente na memória, um após o outro.

Valores

9	7	8	2	0
---	---	---	---	---

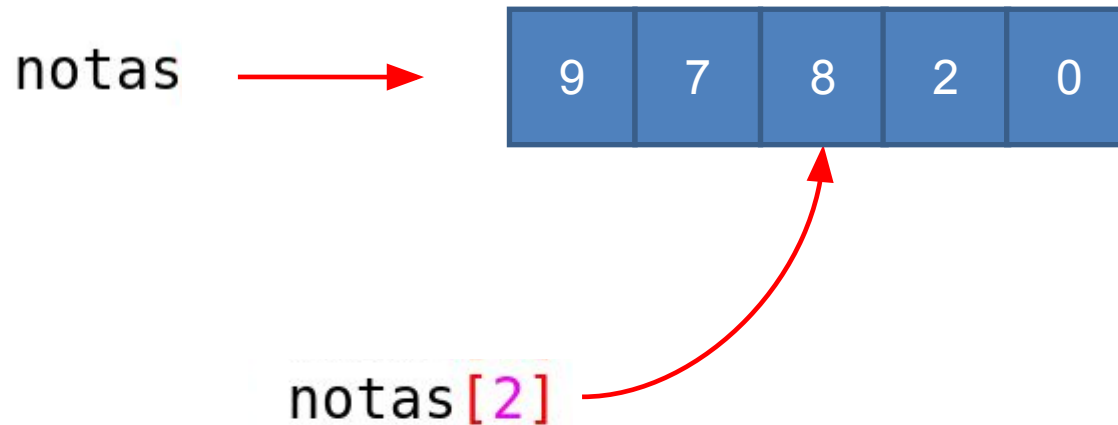
Índices

0      1      2      3      4



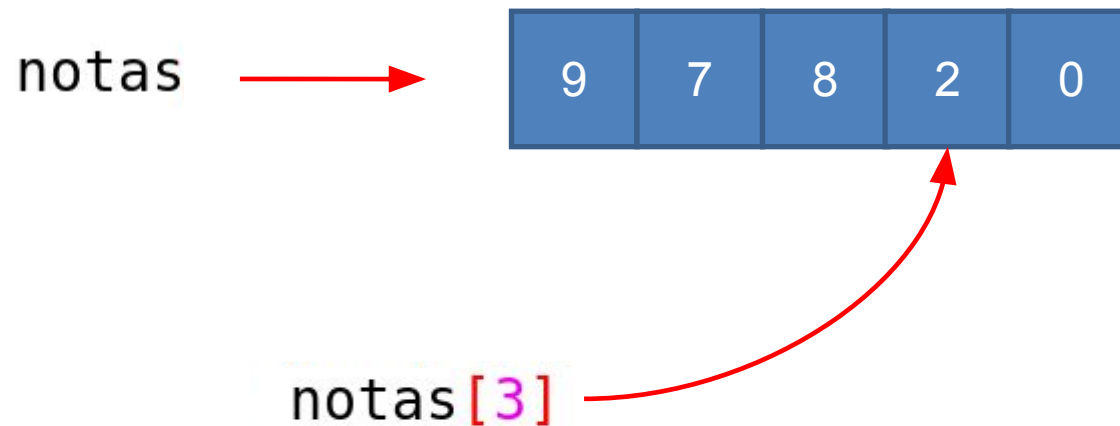
## Criando arrays em C

Internamente, os dados se organizam linearmente na memória, um após o outro.



## Criando arrays em C

Internamente, os dados se organizam linearmente na memória, um após o outro.



## Criando arrays em C

Um índice também pode ser uma variável:

```
i = 5;
```

```
dados[i] = 42;
```

O elemento 5 do array dados recebeu o valor 42.

## Criando arrays em C

Os elementos de um array podem trocar valores com variáveis simples e com outros elementos de arrays:

```
i = 5;  
j = 7;  
v = 8;
```

```
dados[i] = 42;  
i = 6;
```

```
dados[i] = v;  
dados[j] = dados[5];  
dados[9] = dados[j] + 3;
```

```
v = dados[j];
```

```
alert(v);
```

## Criando arrays em C

Os elementos de um array podem trocar valores com variáveis simples e com outros elementos de arrays:

```
i = 5;  
j = 7;  
v = 8;
```

This page says

42

OK

```
dados[i] = 42;  
i = 6;
```

```
dados[i] = v;  
dados[j] = dados[5];  
dados[9] = dados[j] + 3;
```

```
v = dados[j];
```

```
alert(v);
```

## Iterando em um array

## Criando arrays em C

Uma prática muito comum é acessar todos os elementos de um array por meio de um loop:

```
num = []  
  
for(i=0; i<10; i++) {  
    num[i] = i;  
}
```

O código acima cria um array e preenche com números de 0 a 9.

## Criando arrays em C

Uma prática muito comum é acessar todos os elementos de um array por meio de um loop:

```
num = []
```

```
for(i=0; i<10; i++) {  
    num[i] = i;  
}
```

```
for(i=0; i<10; i++) {  
    document.write(num[i]+"<br>");  
}
```

O código acima cria um array e preenche com números de 0 a 9.



## Criando arrays em C

Suponha agora a necessidade de ler notas de 5 alunos e em seguida exibir essas notas. Para isso o código a seguir é uma solução que atende à tarefa.

```
notas = [];  
  
for(i=0; i<5; i++) {  
    notas[i] = Number(prompt("Informe uma nota: "));  
}  
  
for(i=0; i<5; i++) {  
    document.write("Nota: " + notas[i] + "<br>");  
}
```

## Para saber mais...

- <https://www.javatpoint.com/c-array>
- [https://www.tutorialspoint.com/cprogramming/c\\_arrays.htm](https://www.tutorialspoint.com/cprogramming/c_arrays.htm)
- <https://www.programiz.com/c-programming/c-arrays>
- <https://www.geeksforgeeks.org/arrays-in-c-cpp/>
- <http://linguagemc.com.br/vetores-ou-arrays-em-linguagem-c/>
- [https://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-18\\_Arrays.html](https://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-18_Arrays.html)

# Exercícios



# Exercícios

1. Crie um array e armazene todos os números inteiros entre 0 e 100.
2. Crie um array para armazenar todos o números pares entre 0 e 100. Em seguida exiba esses dados.
3. Crie um array para armazenar as letras da palavra ETEC.
4. Escreva um programa que recebe a notas de 10 alunos e as armazena em um array. Em seguida exiba
  - a. A média dessas notas.
  - b. A maior nota.
  - c. As notas acima da média.
5. Crie um array com todos os números ímpares entre 0 e 30 e os exiba em ordem decrescente.
6. Crie um programa que recebe a altura de 10 pessoas e as armazene um array. Em seguida copie esses dados em um segundo array.