

Portugol Studio - Dominando a Lógica de Programação

Rodrigo R. Silva

{Portugol  Studio}

- Chamamos de vetores os conjuntos de variáveis agrupadas do mesmo tipo.

- Chamamos de vetores os conjuntos de variáveis agrupadas do mesmo tipo.
- Um exemplo prático da utilização de vetores: Imagine-se que precisasse armazenar as notas de 50 alunos.

- Chamamos de vetores os conjuntos de variáveis agrupadas do mesmo tipo.
- Um exemplo prático da utilização de vetores: Imagine-se que precisasse armazenar as notas de 50 alunos.
- Na forma como vínhamos declarando as variáveis, teríamos que declarar 50 variáveis, ficando assim: nota1, nota2, nota3, nota4,.....até nota50

- Chamamos de vetores os conjuntos de variáveis agrupadas do mesmo tipo.
- Um exemplo prático da utilização de vetores: Imagine-se que precisasse armazenar as notas de 50 alunos.
- Na forma como vínhamos declarando as variáveis, teríamos que declarar 50 variáveis, ficando assim: nota1, nota2, nota3, nota4,.....até nota50
- Com vetores, a declaração fica da seguinte maneira: **real** nota [50]. Isso é equivalente a criar 50 variáveis do tipo real.

Como funciona

A única diferença entre a declaração de vetores e a declaração de variáveis simples é que, ao declarar vetores, especificamos ao fim da declaração e entre colchetes [] o tamanho do vetor, ou seja, sua capacidade de armazenamento.

Um vetor é como uma coleção de caixinhas numeradas. Cada caixinha é capaz de armazenar um valor e tem o seu “endereço”. O “endereço” de cada caixinha é conhecido como índice e serve para identificar qual posição do vetor queremos acessar. Abaixo temos a representação gráfica no nosso vetor de notas de dez posições declarado acima:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	8.5	7.6	9.3	10.0	6.3	4.7	8.8	9.1	3.4	10.0

Um vetor é como uma coleção de caixinhas numeradas. Cada caixinha é capaz de armazenar um valor e tem o seu “endereço”. O “endereço” de cada caixinha é conhecido como índice e serve para identificar qual posição do vetor queremos acessar. Abaixo temos a representação gráfica no nosso vetor de notas de dez posições declarado acima:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	8.5	7.6	9.3	10.0	6.3	4.7	8.8	9.1	3.4	10.0

No vetor representado acima temos, por exemplo, armazenado no índice “2” o valor “9.3”. Note que, como declaramos um vetor com 10 posições, os índices variam de 0 a 9.

Vetor

Um vetor pode ser visto como uma variável que possui diversas posições, e com isso armazena diversos valores, porém todos do mesmo tipo.

- Como referenciar as posições do vetor e armazenar dados nas mesmas?

- Como referenciar as posições do vetor e armazenar dados nas mesmas?
- A sintaxe para utilizar uma posição do vetor é:

- Como referenciar as posições do vetor e armazenar dados nas mesmas?
- A sintaxe para utilizar uma posição do vetor é:
- **nome_do_vetor[indice];**

- Como referenciar as posições do vetor e armazenar dados nas mesmas?
- A sintaxe para utilizar uma posição do vetor é:
- **nome_do_vetor[indice];**
- Assim, para armazenar o valor 8.5 na primeira posição do vetor, seria utilizado o comando: `notas[0] = 8.5;`

- A única diferença entre a atribuição de valores quando utilizamos variáveis e quando utilizamos vetores é que, usando vetores, precisamos indicar o índice a ser utilizado.

- A única diferença entre a atribuição de valores quando utilizamos variáveis e quando utilizamos vetores é que, usando vetores, precisamos indicar o índice a ser utilizado.
- Trabalhar com vetor, significa trabalhar com estruturas de repetições.

- A única diferença entre a atribuição de valores quando utilizamos variáveis e quando utilizamos vetores é que, usando vetores, precisamos indicar o índice a ser utilizado.
- Trabalhar com vetor, significa trabalhar com estruturas de repetições.
- O motivo é simples: trabalharemos com muitas variáveis e deixaremos o trabalho mais pesado para a estrutura de repetição.

- A única diferença entre a atribuição de valores quando utilizamos variáveis e quando utilizamos vetores é que, usando vetores, precisamos indicar o índice a ser utilizado.
- Trabalhar com vetor, significa trabalhar com estruturas de repetições.
- O motivo é simples: trabalharemos com muitas variáveis e deixaremos o trabalho mais pesado para a estrutura de repetição.
- A melhor estrutura de repetição para trabalhar com vetores é o **para**, pelo simples motivo de sempre ter-se o número exato dos elementos do vetor.

```
programa
{
    funcao inicio()
    {
        inteiro vetor[5] = {15,22,8,10,11}

        escreva(vetor[0])

        escreva(vetor[1])

        escreva(vetor[4])

        real outro_vetor[10]

        caracter nome[] = {'P','o','r','t','u','g','o','l'}

    }
}
```

OBRIGADO!



github.com/Prof-Rodrigo-Silva



[@rodrigo-rosa-da-silva-472928138](#)

{Portugol  Studio}