

Vetores Unidimensionais (array)

tipo **variavel**[] = new **tipo**[tamanho do vetor];

- **variavel** = Pode ser qualquer nome válido e escrito conforme as boas práticas de programação;
- **tipo** = Pode ser qualquer tipo de dado ou classe;

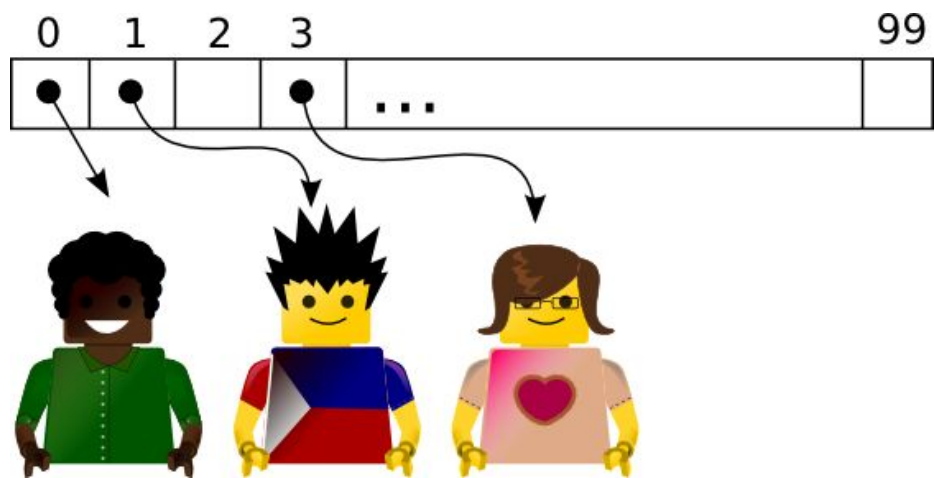
Exemplo:

String **nomes**[] = new **String**[100];

Atribuímos e acessamos valores para cada posição de forma individual:

```
nomes[0] = João;  
nomes[1] = Pedro;  
nomes[3] = Maria;  
nomes[...]= <String>;
```

```
system.out.println(nomes[0]);  
system.out.println(nomes[1]);  
system.out.println(nomes[3]);  
system.out.println(nomes[...])
```



Declarando um vetor

- **int** **vec1**[];
 vec1 = new **int**[4];
 vec1[0] = 1;
 vec1[1] = 2;
 vec1[2] = 3;
 vec1[3] = 4;
- **int** **vec1** = {1,2,3,4};

Atividades

- Criar um vetor de string com tamanho 10
 - Atribuir nomes/palavras para todas as posições
 - Imprimir os nomes/palavras
- Criar um vetor de inteiro com tamanho 8.
 - Atribuir valores para todas as posições
 - Imprimir os valores
 - Imprimir a soma dos valores contidos
- Criar um vetor de double com tamanho 12.
 - Atribuir valores para todas as posições
 - Imprimir os valores
 - Imprimir a soma dos valores pares contidos
 - Imprimir a multiplicação dos valores impares

Vetores Bidimensionais (matriz)

Os vetores bidimensionais também conhecidos como matrizes seguem o mesmo princípio do unidimensional, a diferença é que no caso do bidimensional temos linhas e colunas, ou seja, uma mesma variável terá posições na vertical e na horizontal, observe o exemplo abaixo:

```
double notas[][] = new double[6][4];
```

	0	1	2	3
0	7.5	5.3	7.2	7.7
1	5.6	2.3	4.5	6.7
2	8.8	9.1	9.4	8.3
3	3.5	4.2	7.0	6.7
4	1.6	2.3	4.5	4.7
5	4.8	7.1	8.4	8.3

```
system.out.println(nomes[0][1]);  
system.out.println(nomes[1][3]);  
system.out.println(nomes[4][3]);
```

Declarando uma matriz

- **int** mI [][];
mI = new **int**[2][2];
mI[0][0] = 1;
mI[0][1] = 2;
mI[1][0] = 3;
mI[1][1] = 3;
- **int** mI = {{1,2} {3,4}};