

Arrays in Java

CSCI 111

Data Storage

- How do we currently store data?

Data Storage

- How do we currently store data?
Variables!

Data Storage

- How do we currently store data?

Variables!

- How would we store the ages of everyone in this class?

Data Storage

- How do we currently store data?

Variables!

- How would we store the ages of everyone in this class?
- How would we store Employee instances for a company?

Arrays

- So, it would be nice to have an efficient way to store lots of data without having to create a variable for each piece of data.

Arrays

- So, it would be nice to have an efficient way to store lots of data without having to create a variable for each piece of data.
- Arrays are data structures used to store a fixed number of values of the same data type (i.e. You can't mix Strings, ints, and Students).

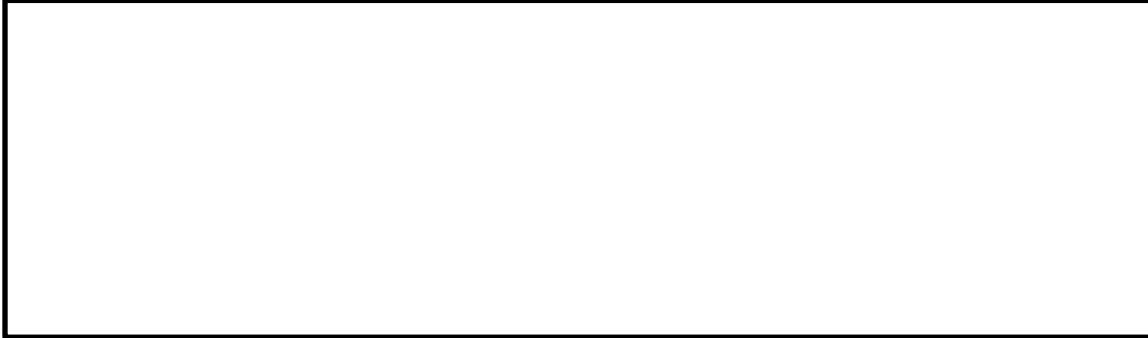
```
int x = 5
```

```
int[] x = 

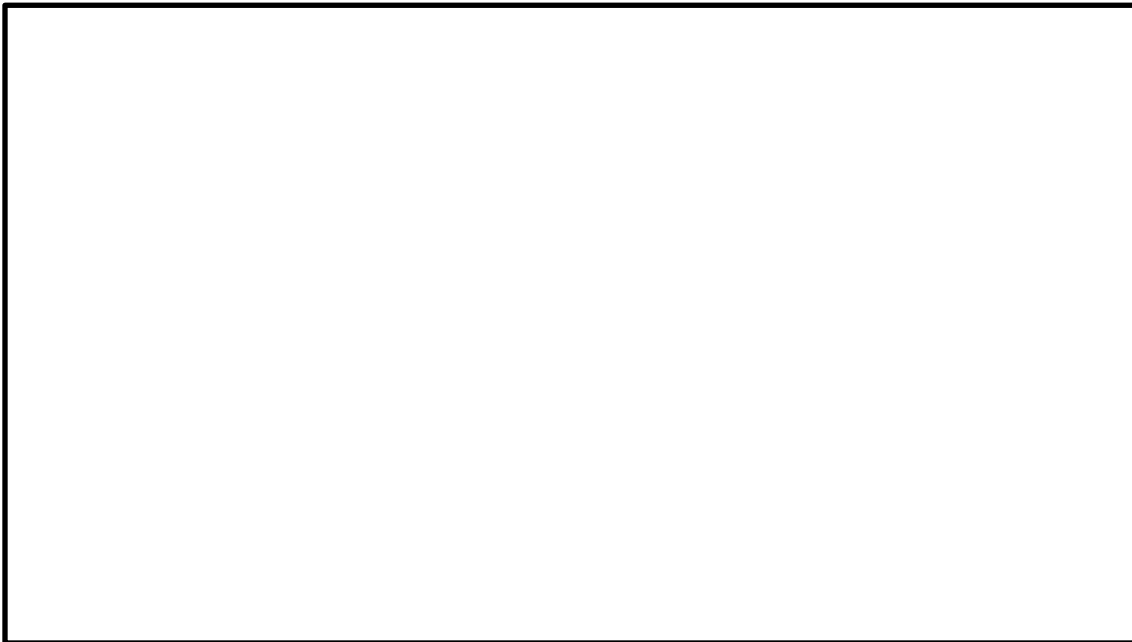
|   |   |    |    |   |   |     |   |   |   |
|---|---|----|----|---|---|-----|---|---|---|
| 0 | 2 | 21 | 37 | 6 | 4 | 291 | 3 | 4 | 5 |
|---|---|----|----|---|---|-----|---|---|---|


```

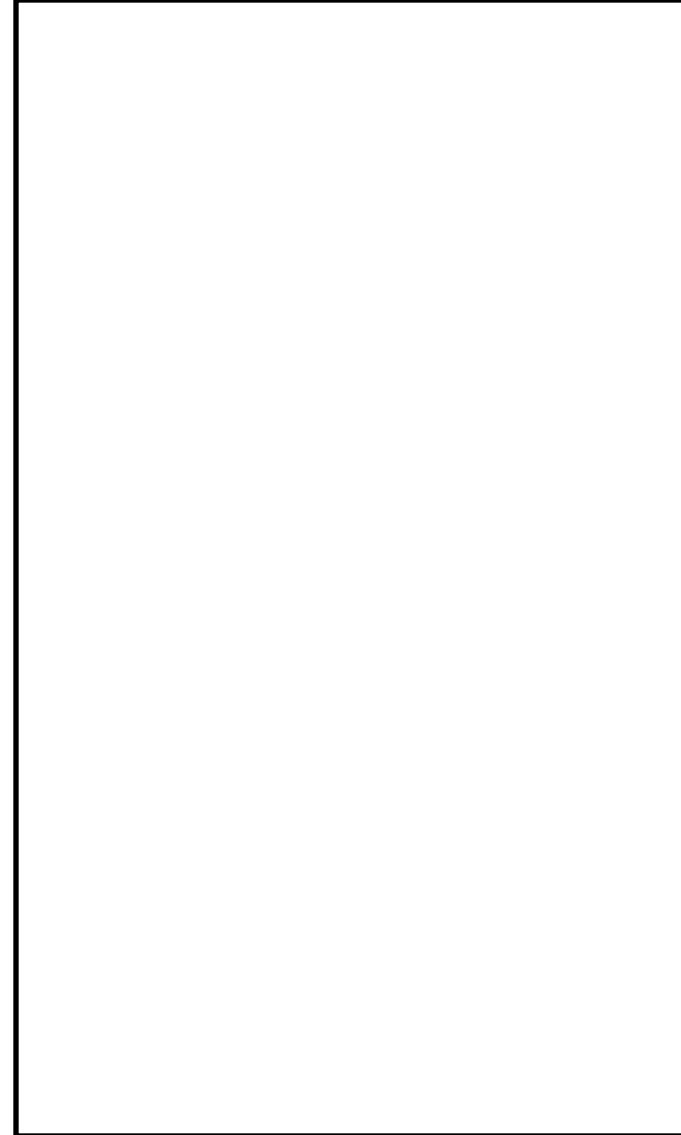
Java Code



What Happens?



Computer
Memory



Java Code

```
int x;
```

Computer Memory

X

What Happens?

A new variable, `x`, is created. `x` can ONLY hold an integer (variable declaration).

Java Code

```
int x;  
x = 5;
```

Computer Memory

x = 5

What Happens?

A new variable, `x`, is created. `x` can ONLY hold an integer (variable declaration).

`x` is assigned the value 5 (variable assignment).

Java Code

```
int x;  
x = 5;  
  
int[] ages;
```

What Happens?

A new variable, `ages`, is created. `ages` can ONLY hold an integer array (variable declaration).

Computer Memory

`x = 5`

`ages`

Java Code

```
int x;  
x = 5;  
  
int[] ages;  
ages = new int[4];
```

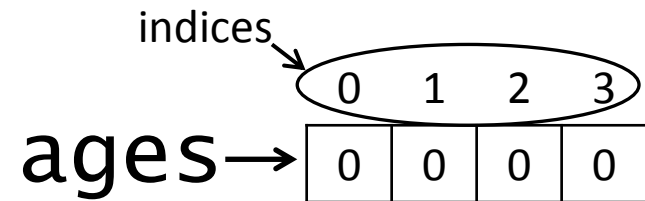
What Happens?

A new variable, `ages`, is created. `ages` can ONLY hold an integer array (variable declaration).

A new integer array with 4 slots is created and `ages` is set to point to it.

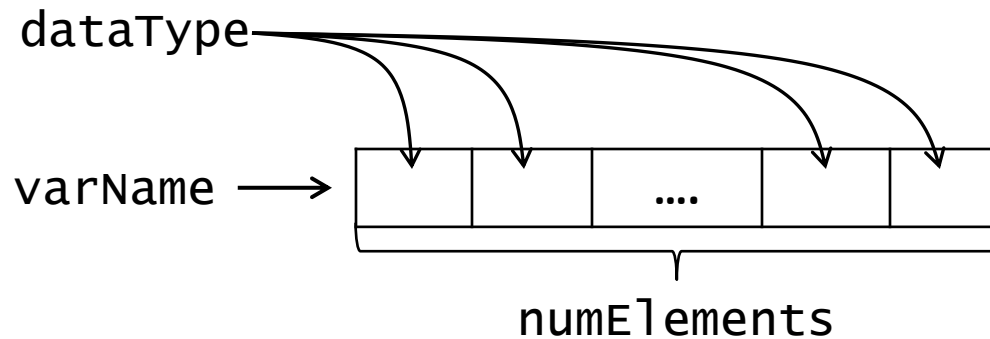
Computer Memory

`x = 5`



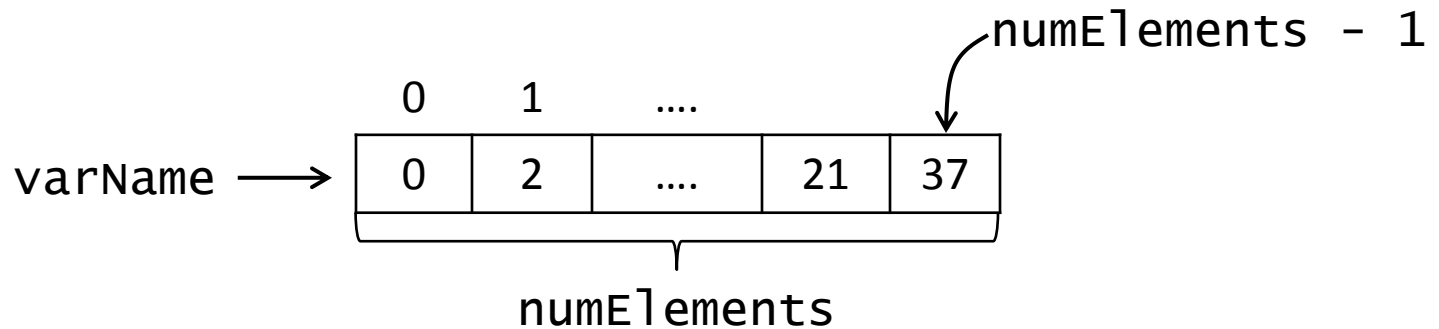
Array Structure

```
dataType[] varName = new dataType[numElements];
```



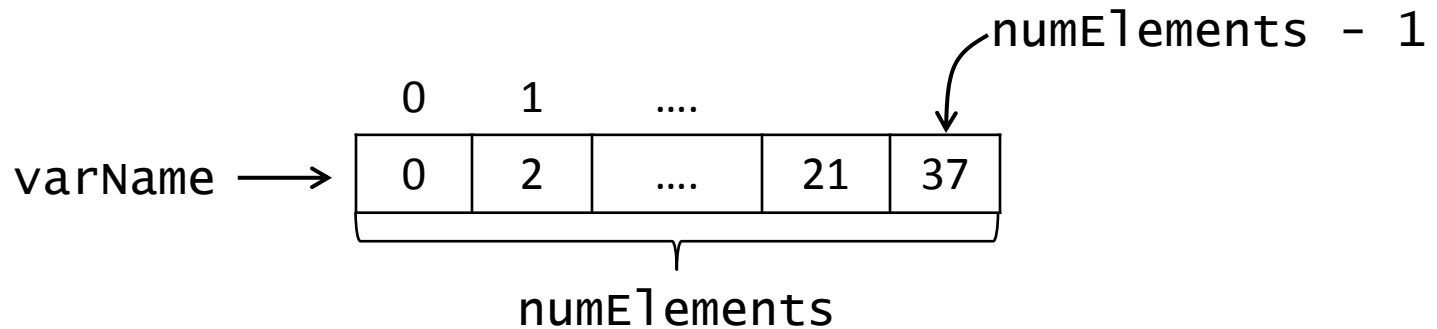
Array Structure

```
dataType[] varName = new dataType[numElements];
```



Array Structure

```
dataType[] varName = new dataType[numElements];
```



`varName.length` is the number of elements.

Java Code

```
int[] ages;  
ages = new int[4];  
  
ages[1] = 8;
```

What Happens?

Computer Memory

ages →

0	1	2	3
0	0	0	0

Java Code

```
int[] ages;  
ages = new int[4];  
  
ages[1] = 8;
```

What Happens?

The value in the slot in `ages` at index 1 is set to 8.

Equivalent to variable assignment.

Computer Memory

`ages` →

0	1	2	3
0	8	0	0

Java Code

```
int[] ages;  
ages = new int[4];  
  
ages[1] = 8;  
ages[0] = 4;  
ages[3] = 9;
```

What Happens?

Computer Memory

ages →

0	1	2	3
4	8	0	9

Java Code

```
int[] ages;  
ages = new int[4];  
  
System.out.println(ages[3]);
```

What Happens?

We go to the slot at index 3 and retrieve (and print) the value.

So, 9 is printed.

Computer Memory

ages →

0	1	2	3
4	8	0	9

Java Code

```
int[] ages;  
ages = new int[4];  
  
System.out.println(ages[3]);  
System.out.println(ages.length);
```

What Happens?

How many elements are in the array that `ages` is pointing to? 4 elements.

So, 4 is printed.

Computer Memory

`ages` →

0	1	2	3
4	8	0	9