Arrays

Java
Mr. Poole

Data Storage - What we know

Variable

int number

Value: 5

A variable with a primitive data type can store **1 value**.

Object Variable

Dog temp

Name: “Barney”
Age: 3
Breed: “Corgi”

A variable with a Class Object data type can store **multiple values of different data types** in this one variable.

Array

???

*primitive data types are: **int, double, boolean, char** and a few more (short, long, byte, float)

Averaging

How many variables in a program before it gets too hard to perform the operation?

- Averaging 3 scores (3 variables?)
- Averaging 10 scores (10 variables?)
- Averaging 100 scores (100 variables?)

Variable Average

```
int score0 = 2;
int score1 = 10;
int score2 = 9;
int score3 = 10;
int score4 = 7;
int score5 = 8;
int score6 = 9;
int score7 = 9;
int score8 = 9;
int score9 = 9;
double avg = score0+score1+score2+score3+score4+score5+score6+score7+score8+score9;
avg = avg / 10.0;
System.out.println(avg);
```

This is not it.

Introducing **Arrays**

Arrays are used to **store multiple values in a single variable**, instead of declaring separate variables for each value.

Data Storage - Arrays

Variable

int number

Value: 5

A variable with a primitive data type can store **1 value**.

Object Variable

Dog temp

Name: “Barney”
Age: 3
Breed: “Corgi”

A variable with a Class Object data type can store **multiple values of different data types** in this one variable.

Array

int [] test = new int [10];

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

A **singular variable** that holds **multiple values** of the **same data type.**

Constructing Array Variables

These brackets indicate that this variable is an array

```
int temp;
String temp2;
double temp3;
Dog temp4;
```

```
int [] arr;
String [] arr2;
double [] arr3;
Dog [] arr4;
```

Single Variables
with one value

Array Variables
with multiple value

Initializing Array Variables

Arrays MUST have a defined size when created.

```
int [] arr = new int[10];
String [] arr2 = new String[15];
double [] arr3 = new double[5];
Dog [] arr4 = new Dog[5];
```

These values define **how many values** the array will store.

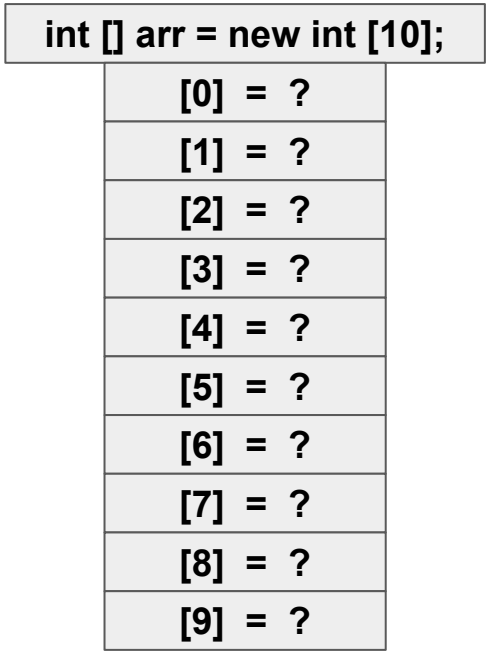
Notice all Arrays are made with the keyword **“new”**!

Initializing Array Variables

```
int [] arr = new int[10];
```

We **reserve** 10 spots to be used!

They don’t have values yet though!



*Try seeing what default values are printed when an array isn’t assigned values.

Assigning Array values

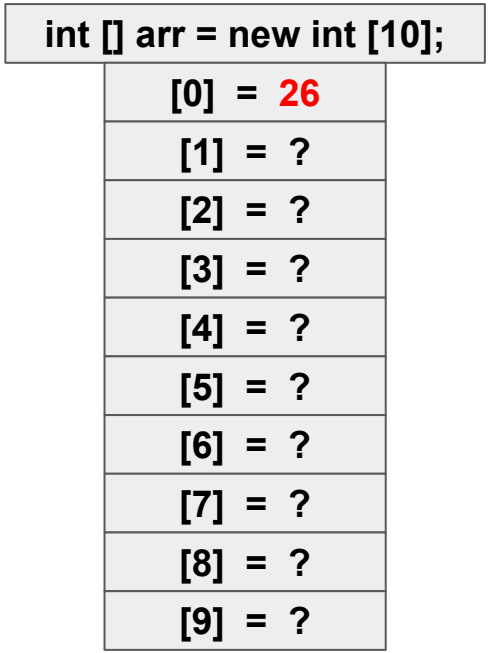
```
int [] arr = new int[10];
arr[0] = 26;
```

Just the variable name

Here we start at the first **index, 0.**

Always start at 0!

This **assigns spot** (index) **0** to 26.



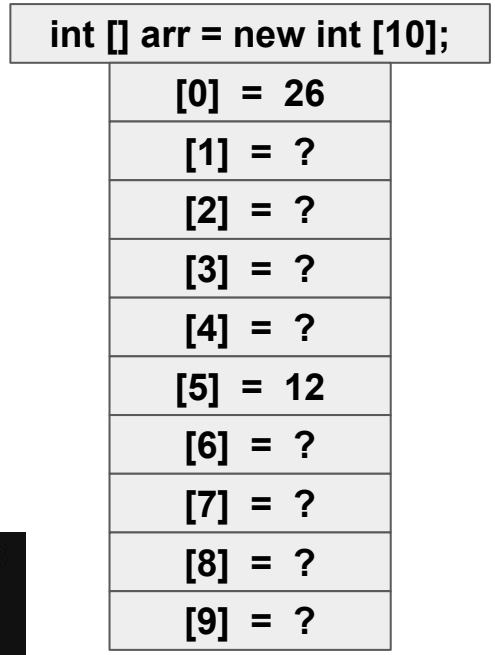
Index Out Of Bounds

```
int [] arr = new int[10];
arr[0] = 26;
arr[5] = 12;
arr[10] = 9;
```

Now we go to far!

This causes an error.

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:
 Index 10 out of bounds for length 10
        at test.main(test.java:30)
```



[10] ???

Knowing your **Length**

```java
int [] arr = new int[10];
int len = arr.length;
System.out.println(len);
```

`arr.length`
gives the length of how many elements are in the array.

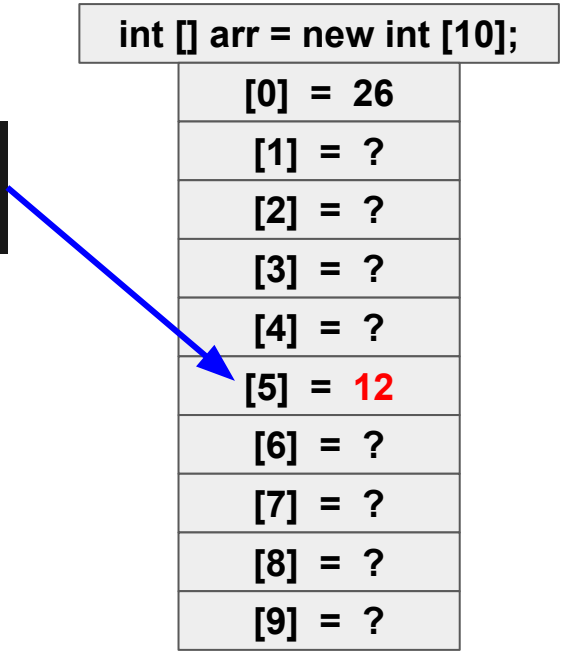
This prints 10, meaning 0 to 9 inclusive.

Using your array

Treat arrays just like normal variables!

```
System.out.println(arr[5]);
```

This prints the 5th index's value, in the case above, it was **12**.



Lab: Arrays

- Make an array of integers
  - Set the size to 10
- Give all 10 elements the following:
  - Respectively 9 to 0
  - Ex: index 0 has 9 and
    index 9 has 0
  - Print out all numbers