Java: Day 3

with
Project Include

Review - Booleans (True/False)

```
boolean myBoolean = true;
```

Equality of Strings:

varString.equals(“string”);

Relationships Between Integers

== Equals

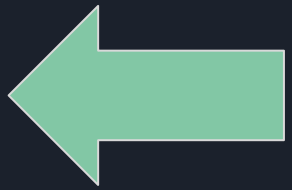
!= Does not equal

> Greater than

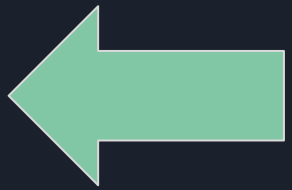
< Less than

Review - Conditional Statements (If / Else)

```
if (condition) {
    yourCodeHere ;
}
else {
    yourCodeHere ;
}
```



Executes code if condition is true (if the condition is met)



Executes code if condition is not met

Review - Multiple Conditions

And (&&)

```
if (condition1 && condition2) {
        yourCodeHere ;
        }
```

Or (||)

```
if (condition1 || condition2) {
        yourCodeHere ;
        }
```

Not (!)

```
if (condition1 ! condition2) {
        yourCodeHere ;
        }
```

Review - Loops

While Loops

Run through code as long as a condition is true

```
while(condition) {

    yourCodeHere ;

}
```

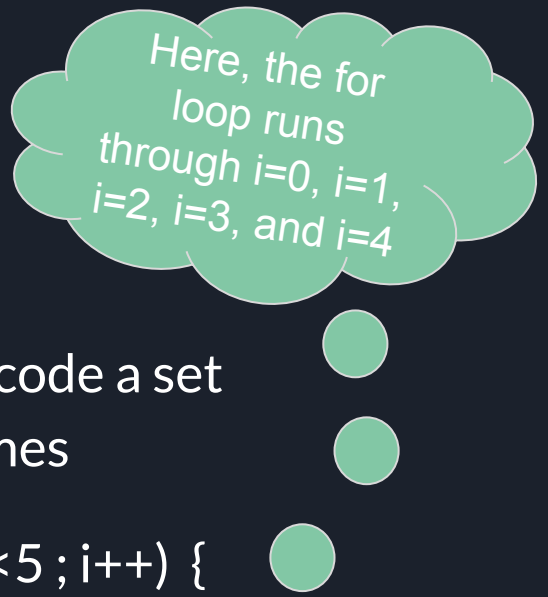
For Loops

Run through code a set number of times

```
for  (int i=0; i<5 ; i++) {

    yourCodeHere ;

}
```



Your Code

tiny.cc/join-class

or plain code: tiny.cc/java-day3

repl.it/languages/java

Arrays

Arrays: basic shelf for variables

- Fixed length, one variable type only
- Can access one element with an index
  - Think of finding box by shelf #
  - Indexes start at 0, not 1
- Can also loop over array elements

Arrays: basic shelf for variables

```
int[] numberArray = {1, 3, 5, 7, 9};
int[] blankArray = new int[10];

System.out.println(numberArray[2]);
System.out.println(blankArray[7]);
```

- TRY: loop over numberArray printing out every element!
- hint: you can find the length of numberArray by using numberArray.length

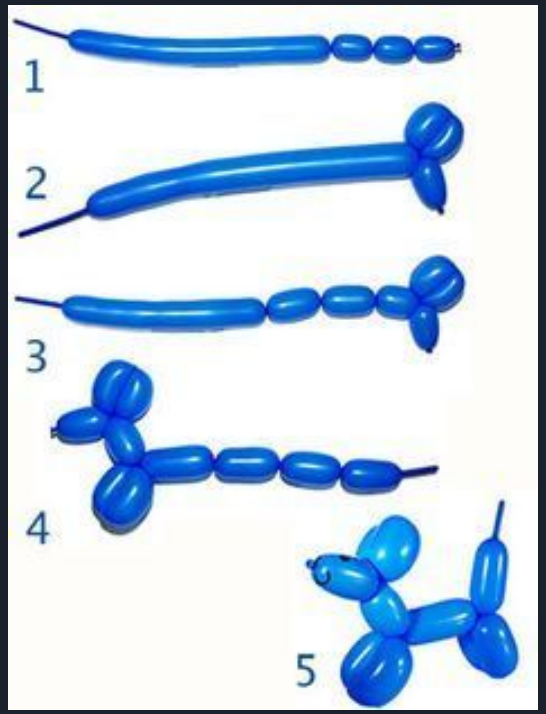
Arrays: basic shelf for variables

```
for (int i = 0; i < numberArray.length; i++) {
  System.out.println(numberArray[i]);
}
```

```
for (int currentNumber : numberArray) {
  System.out.println(currentNumber);
}
```

Functions

Functions: separating the work



- Write a function once and use it forever
- Saves space
- Many have already been written for you

Functions: separating the work

```
public static int exampleFunction(int inputNumber, String inputWord) {

  // Do stuff here using inputNumber and inputWord
  int outputNumber = 17;

  return outputNumber;
}
```

TRY: find the function called “numberTransformer” in the code. Follow the instructions inside, then try using it

Functions: separating the work

```
System.out.println(numberTransformer(5));
```

```
public static int numberTransformer(int inputNumber) {

  // Do something here with inputNumber to create outputNumber
  int outputNumber = inputNumber * inputNumber + 356;

  return outputNumber;
}
```

Characters

Characters: single letters/numbers

```
char letter = 'q';
char number = '7';
```

- Remember: use single quotation marks!
- Every String is just a collection of chars

String ---> Array of Characters

```
myString.length();
myString.charAt(integerIndex);
```

- Try: Use these functions and a FOR loop to create a new array and fill it with the characters from a String

String ---> Array of Characters

```
String helloString = "hello";
char[] helloArray = new char[helloString.length()];

for (int i = 0; i < helloString.length(); i++) {
  helloArray[i] = helloString.charAt(i);
}
```

- Is there an easier way?

String ---> Array of Characters

```
helloArray = helloString.toCharArray();
```

- There are tons of functions to help you!

(just use Google)

Array of Characters ---> String

```
helloString = new String(helloArray);
```

- Converting back is easy with a function!

ArrayLists

ArrayLists: advanced shelves!

```
ArrayList<Character> newArrayList = new ArrayList<>();
```

- Can modify length and hold anything!
- Has lots of functions to do work for us

ArrayLists: adding elements

```
newArrayList.add(someCharacter);
```

- Use this function to add a character
- Use a FOR loop to go through an Array and add all of the characters to the ArrayList!

Result

```
char[] oldArray = {'n', 'i', 'f', 't', 'y'};
ArrayList<Character> newArrayList = new ArrayList<>();

for (char oldLetter : secret) {
  newArrayList.add(oldLetter);
}
```

ArrayLists: checking contents

```
newArrayList.contains(guessLetter);
```

- This function returns “true” or “false”!
- Use an IF statement to tell the user if a certain letter is in our ArrayList

Result

```
char guessLetter = 't';

if (newArrayList.contains(guessLetter)) {
  System.out.println("You guessed it!");
}
else {
  System.out.println("Nope, not this time");
}
```

Feedback

tiny.cc/PI_feedback