Command Line Arguments

By: Megan Avery

Updated December 2018

Before we begin

- Make sure you have Java & Java JDK downloaded on your computer
  - can run **`java -version`** in terminal to check
- Make sure you have IntelliJ downloaded on your computer

- *Suggested:* Watch previous Java tutorials

Where?

```
public class Example {
    public static void main(String[] args) {
        // code goes here
    }
}
```

Using String[] args

You can use the command line “args” array just like any other array of Strings. You access its elements in the same way and all other array methods apply.

String[] args will be split on the spaces of the words (or numbers) entered for the command line arguments.

Array Refresh

```
// array access
String x = args[3];

// array modification
args[2] = “4”;

// get array length
int x = args.length
```

Reminder: 0 Based Indexing

Arrays, like Strings and everything else in Java, are 0 indexed. Meaning the first element is at index 0, the second at index 1, etc.

How to enter command line input

Via IntelliJ:

Run **>** Edit Configurations **>** Choose program you want to enter command line input for **>** Enter command line input in the “Program arguments” section **>** Apply **>** Ok

Via the command line:

java <program name> <args separated by spaces>

Reminder: Converting Strings to Numbers

Convert from String to int:

- Integer.parseInt(numberString)

Convert from String to double:

- Double.parseDouble(numberString)

Watch Out For...

ArrayIndexOutOfBoundsExceptions

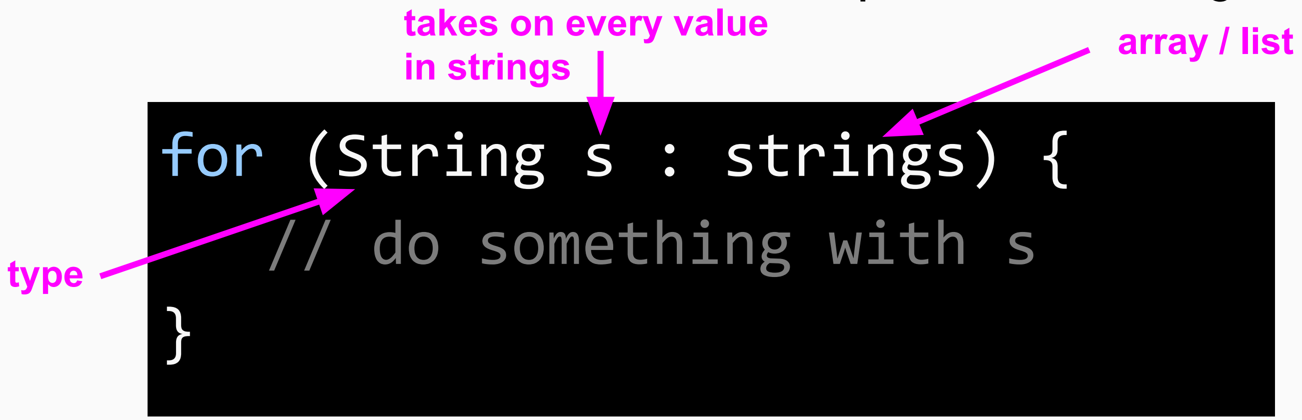
- These will get thrown if you try to access past the end of the array, you can avoid this by checking the size of your String[] args array before accessing its elements

When to use command line arguments

When there is some configuration you want to set up for your Java program. For example, you might have a program that does several different things and you want to give arguments to indicate which one. Or you might know your program does something like add together two numbers and you want to input them as command line arguments instead of using user input.

Bonus: Foreach Loops

For each loops are a way for you to quickly run over an array, or other list, of elements one at a time. An example with a String array is below.



```java
for (String s : strings) {
    // do something with s
}
```

Exercises

CLA Exercise #1

Time Limit: 7 minutes

Write a program called Average that finds the average of its command line arguments. It should be able to handle any number of CLAs and should catch any exceptions that might arise (for example if a CLA happens to not be a number). It should throw an “IllegalStateException” if there are no given command line arguments.

CLA Exercise #2

Time Limit: 7 minutes

Write a program called Echo that takes in the command line arguments and simply prints them back to the screen (like what the echo linux program does). This program should throw an IllegalStateException if there are no given CLAs.

Hint: Use a foreach loop to make things easier

Practice for Later

Edit your Hangman program to have the answer come from a command line argument instead of a constant at the top. It should be able to be single or multi word. If there are no command line arguments given throw an IllegalStateException with information explaining what went wrong.

The End