Java Tutorial

Basic Rules - 1: Demo

1. All code in Java must be part of **a class**
2. For code to run you need to have

```
public static void main(String[] args)
```

1. We mark the *beginning* and *end* of segments of code using `{` and `}`
2. All statements in Java must end in a semi-colon: ;

Basic Rules - 2

1. Before Java variables can be used, they must be ***declared***
2. Java variable must have a specific type:
   a. int, String, double, boolean etc
3. Types can *never* change
4. Types are verified *before* the code even runs
   a. Big difference between Python and Java

Defining Functions. Basic Rules

1. Functions must be declared as part of a class in Java
   a. A function that is inside a class is called a "method"
   b. All functions in Java are methods
2. To define a function in Java we use "public static"
   a. Other ways are later
3. All parameters must have a declared type
4. Return value of the function must have a declared type
5. Functions in Java return only one value

Discussion Question - 1

How many errors can you find in the code on the right?

A: 1
B: 2
C: 3
D: 4
E: 5 or more

```
public class Discussion {
    public static void main(String[] args) {

        double y = 5.6;
        x = 10;

        if (x < y):
            System.out.println(y is smaller);
        else {
            x = x * y;
            System.out.println(y);
        }

}
```

1 2 int 3 { 4 " " 3 } int double int 3 5

Multiply double & int
↳ store in an int ⇒ Lossy conversion ERROR
① change x to double
② casting result to an int : acknowledging potential loss
x = (int) (x * y)

For Loop in Java

(Plain old for loop)

Declaring and Initializing loop control variable — Checking condition — Incrementing loop control variable

```
for (int i =0; i<10 ; i++) {

// Loop statements to be executed

}
```

What is the output? *

A: 1, 3, 5, 7, 9

B: 1, 3, 5, 7, 9, 11

C: 1, 4, 7, 10

D: 1, 4, 7

E: None of the above

** Assume that the output does not have commas and each number is on a new line.*

```
public class Discussion {

    public static void main(String[] args) {

        for (int i=1; i<10; i=i+2){
            System.out.println(i);
            i = i + 1;
        }
    }
}
```

~~system.out.println(i)~~

i = 0 ← initializer
while i < 10: ← ending condition
    i = i+1 ← increment

Enhanced for loop exists!

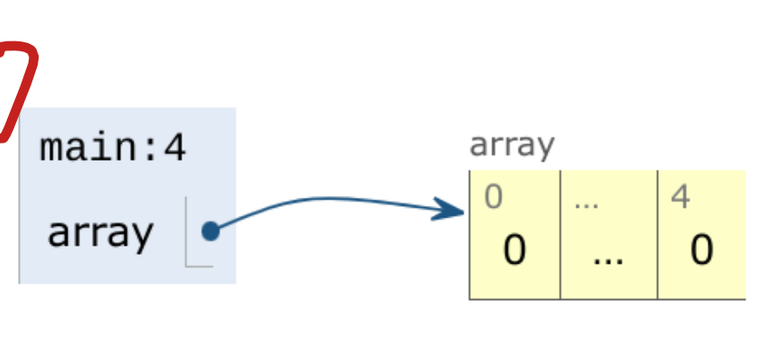
```
for (List<Integer> entry : entries.values()) {
    // result.append(len(entry))
    result.add(entry.size());
}
```

Short Practice

Write a function *expand* that takes an integer and returns an integer array with numbers 1, 2.. up to (including) the parameter:

```
int[] array = new int[5];
array[0] = 1;
array[1] = 14;
array[3] = array[1];
array[4] = 5;
```

new int[5]



Example:
 Input: 5
 Output: [1, 2, 3, 4, 5] ⇐ Array, not a list!

public static int[] expand (...) {

```
import java.util.Arrays;

public class Discussion {
    public static int[] expand(int num) {
        int[] result = new int[num];
        for (int i = 1; i <= num; i = i + 1) {
            result[i - 1] = i;
        }
        return result;
    }
    // Depending on the specification, you may sometimes
    // want to return type void (returning nothing)

    public static void main(String[] args) {
        int[] result = expand(5);
        // Arrays.toString(result)
        // str(result)
        System.out.println(Arrays.toString(result));
    }
}
```

}

Q: ~~i++ (Post increment) vs. ++i (preincrement)~~

i = i+1   IMO just use a separate statement!