For Each Loops

For Each Loops Purpose

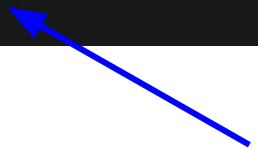
For each loops are an EASY way to **iterate through EVERY value in an array**.

They cannot skip values!

For each structure

This says “for each piece of data **in** the data structure”

```
for( Data type : Data Structure)
```



The colon represents the word “**in**”

Matching Data Types

Notice: nums is an **array** of **“ints”**

The variable “x” is also an “**int**”

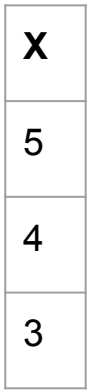
```
int[] nums = {5,4,3};
for(int x : nums)
```

Iterating through a for each loop

This loop has no index or counter.

It goes through all items in a data structure one at a time.

```
int[] nums = {5,4,3};
for(int x : nums)
```



X becomes a copy of each value one at a time.

Warning: for each loops cannot change values in the original data structure.

Output of for each

x is NOT an index, it is a value.

```
int[] nums = {5,4,3};
for(int x : nums){
    System.out.println(x);
}
```

This loop prints out the following:

5
4
3

Output of for each w/ Strings

```
String[] words = {"hi", "hello", "yo"};
for(String word : words){
    System.out.println(word);
}
```

This loop prints out the following:

hi
hello
yo

Output of for each w/ 2D arrays

```
int [][] nums = {
    {5, 4, 3},
    {2, 1, 0},
    {9, 8, 7}
};
for(int[] rows: nums){
    for(int val: rows){
        System.out.print(val + " ");
    }
}
```

This loop prints out the following:

5 4 3 2 1 0 9 8 7

This says:
“for each integer array in nums,
for int in that array”

Lab

This lab does not necessarily use for each loops.

It can if needed.

Read the comments of the starter.java file.