

MODULE 1: INTRODUCTION TO PROGRAMMING

Loops and Arrays



Yesterday

- What is an **expression**?
- What is a **statement**?
- What is a **code block**?
- What is a **method**?
- What is a **Boolean expression**?



Working with variables

TYPE	DESCRIPTION	DEFAULT	SIZE	EXAMPLE LITERALS	RANGE OF VALUES
boolean	true or false	false	1 bit	true, false	true, false
byte	twos complement integer	0	8 bits	(none)	-128 to 127
char	unicode character	\u0000	16 bits	'a', '\u0041', '\101', '\W', '\l', '\n', '\beta'	character representation of ASCII values 0 to 255
short	twos complement integer	0	16 bits	(none)	-32,768 to 32,767
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2	-2,147,483,648 to 2,147,483,647
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F	upto 7 decimal digits
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d	upto 16 decimal digits

Wild Animals



What is an array?

- An **array** is a data structure that is a collection of variables of the same type



Declarative syntax

```
String[ ] animals;
```



Element Type



Variable Name



Initialization syntax

```
animals = new String[6];
```



Variable Name



Number of elements

syntax

```
String[ ] animals = new String[6];
```



Element Type



Variable Name



Number of elements



Assigning values to elements

```
String[ ] animals = new String[6];
```

```
animals[0] = "Tiger";
```

```
animals[1] = "Polar Bear";
```

```
animals[2] = "Giraffe";
```

```
animals[3] = "Koala";
```

```
animals[4] = "Moose";
```

```
animals[5] = "Panda";
```

Accessing values in an array

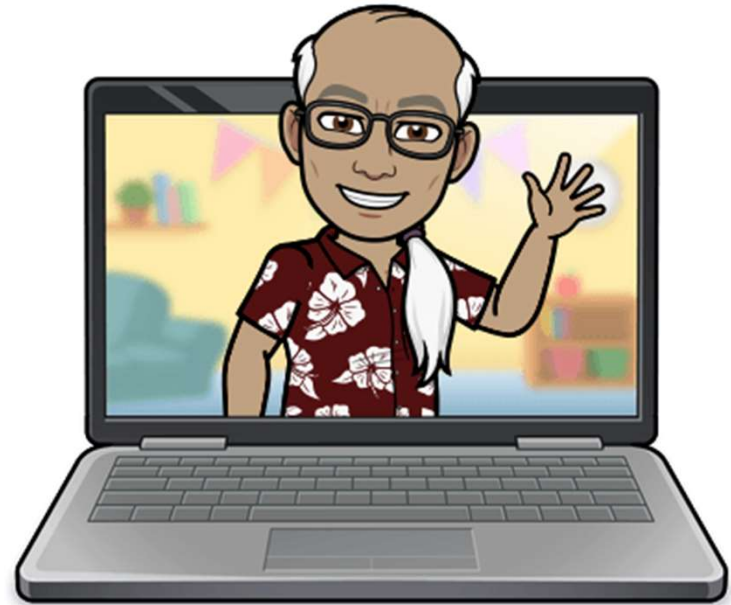
- Write the 3rd value from the array to the console:
 - `System.out.println(animals[2]);`

How do we know the size of an Array?

- `String[] animals = new String[6];`
- `animals.length`



LET'S CODE!



What are arrays good for?

- Keeping related data together for processing
 - Names of students in a classroom
 - `String[] students = new String[20];`
 - Par values on a golf score card
 - `int[] parValues = new int[18];`



The gotchas

- Arrays are zero indexed
- Arrays are fixed length
- Arrays must be initialized before used
- Array are reference types

Scope

```
{  
    int length;  
    int width;  
    int area;  
    area = length * width;  
}
```

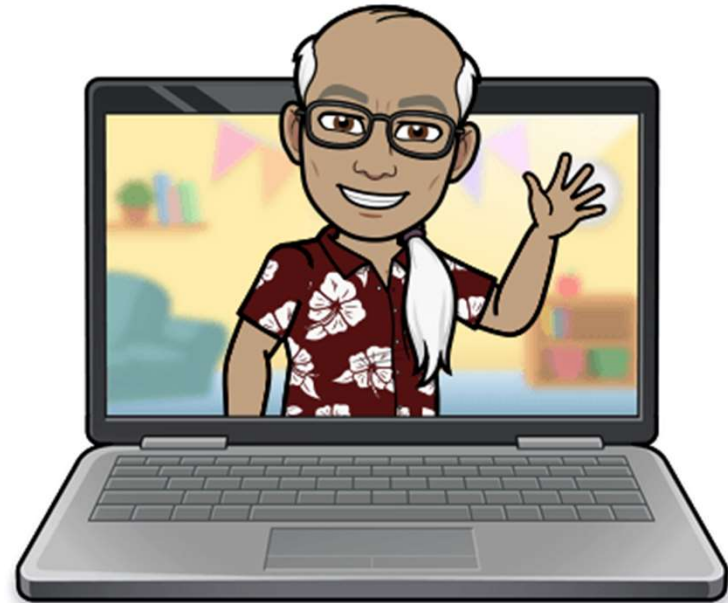
A variable's **scope** defines where in the program that the variable exists (i.e. can be referenced). When code execution reaches a point where a variable is no longer referenceable, the variable is said to be *out of scope*.



Rules of Scope

- Variables declared inside of a function or block `{..}` are local variables and only available within that block. This includes loops.
- Blocks can be nested within other blocks and therefore if a variable is declared outside of a block, it is accessible within the inner block.

LET'S CODE!



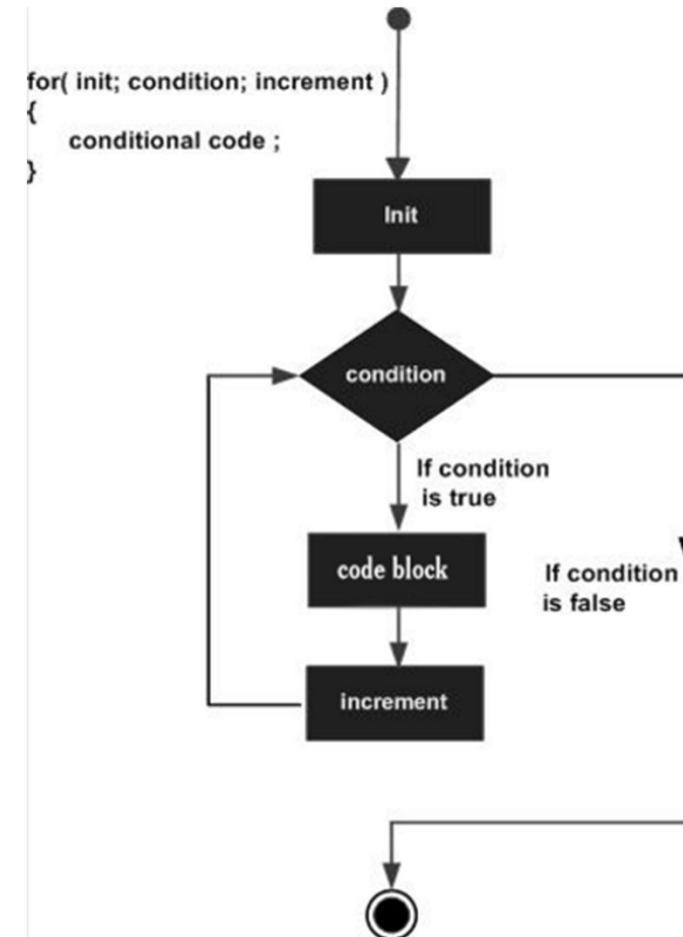
Finding Animals

- How would we find the Moose?



Accessing Elements in an Array

- For loop allows you to check each element in an array.



Memorize this code segment!!

```
for(int i = 0; i < animals.length; i++) {  
    ...  
}
```

Memorize this code segment!!



What is it really doing?

```
String[] animals = new String[5];  
for(int i = 0; i < animals.length; i++) {  
    System.out.println(i);  
}
```

0
1
2
3
4

What is it really doing?

```
String[] animals = new String[5];  
for(int i = 0; i < animals.length; i+=2) {  
    System.out.println(i);  
}
```

0
2
4

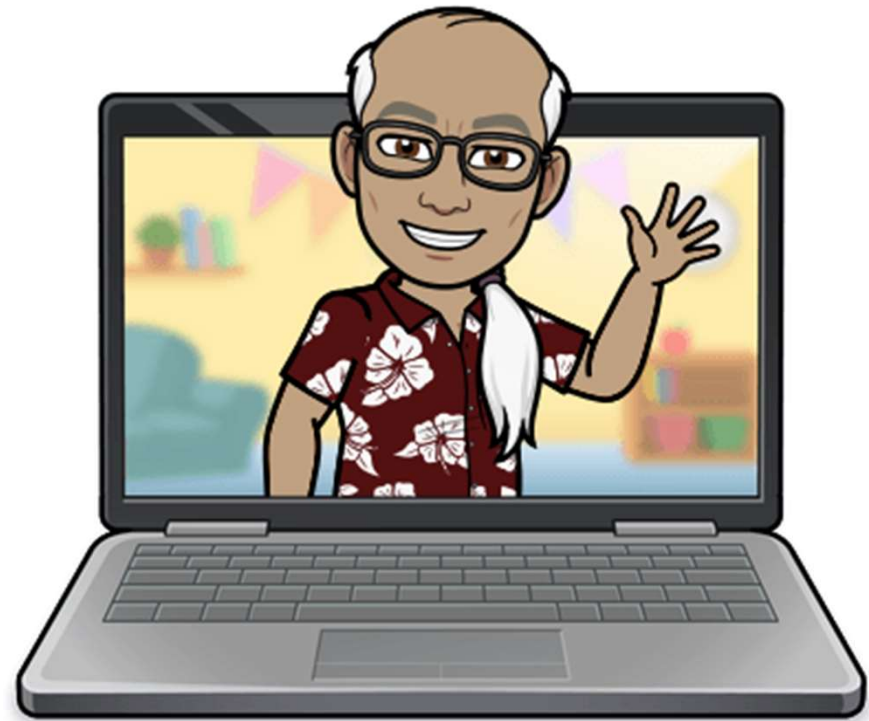


What is it really doing?

```
String[] animals = new String[5];  
for(int i = 0; i < animals.length; i++) {  
    System.out.println(animals[i]);  
}
```

Polar Bear
Giraffe
Moose
Tiger
Koala

LET'S CODE!



Shorthand Notation

- Increment variable x by 1:
 - $x++$ is the same as $x = x + 1$
- Decrement variable x by 1:
 - $x--$ is the same as $x = x - 1$
- Assignment shortcuts
 - $x += n$ is the same as $x = x + n$
 - $x -= n$ is the same as $x = x - n$

Danger of Shorthand notation

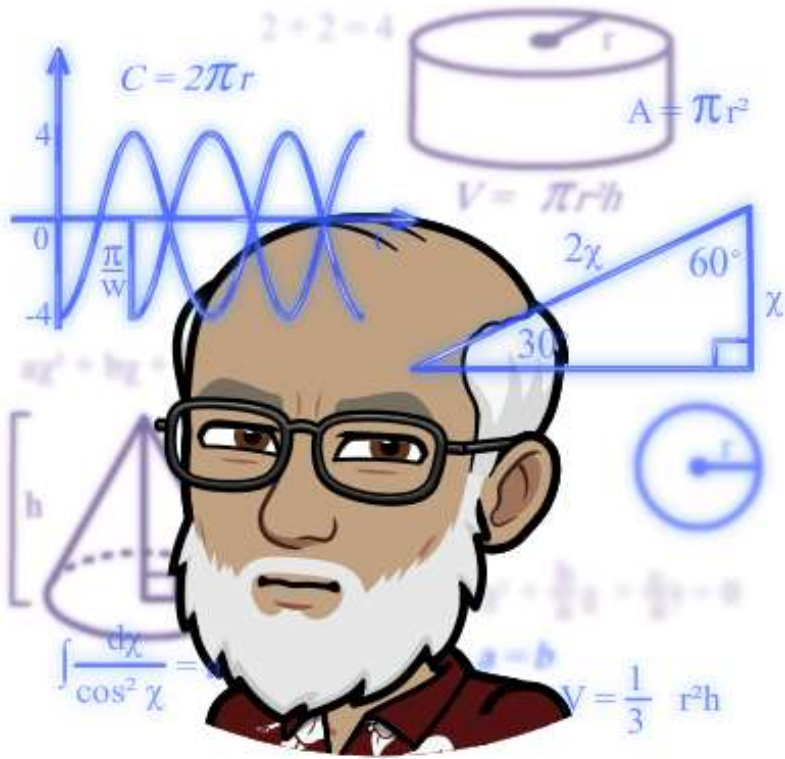
```
{  
    Int x = 0;  
    Int y = 0;  
    Int z = 0;  
    y = x++;  
    z = ++x;  
}
```



WHAT QUESTIONS DO YOU HAVE?



Code Reviews



ELEVATE  YOURSELF



Building Command Line Programs

