Crash Course 2: Variables and Data Types

ST. MARY'S HIGH SCHOOL

ST. MARY'S HIGH SCHOOL

Welcome to Variables and Data Types!

Here's what we'll be covering:

- What is a variable?
- Variable names/identifiers
- What is a data type?
- Declaring the variable
- Initializing the variable
- Using the variable
- Modifying variables
- Examples
  - Number examples (int, float, double)
  - Logic examples (boolean)
  - Text examples (char and String)
- Summary

What is a variable?

- A variable is used to store one piece of information at a time
- When you need a variable, follow these steps:
  - 1) Declare the variable
  - 2) Initialize the variable
  - 3) Use the variable
- Variables are useful in code because they allow us to write flexible programs. Rather than manually entering a piece of data into a program, we store the data in a variable, and use and modify it as necessary.

Variable names/identifiers

- Every variable has a name or identifier
- While you can name your variable whatever you like, it is best to name it something meaningful that is related to its purpose
- For multiple word names, two good conventions are areaOfCircle (using lowerCamelCase) or area_of_circle (using underscores)
- Note that identifiers cannot contain characters such as:

  / , ; . & * + - # = “ ‘
- Names can begin with a letter or an underscore, but not a number

What is a data type?

- When defining a variable, the programmer must specify the type of information to be stored. This is the “data type” of the variable
- Data types classify the type of information in your variable, and lets the computer know what type of information is being stored in that variable

What is a data type? (2)

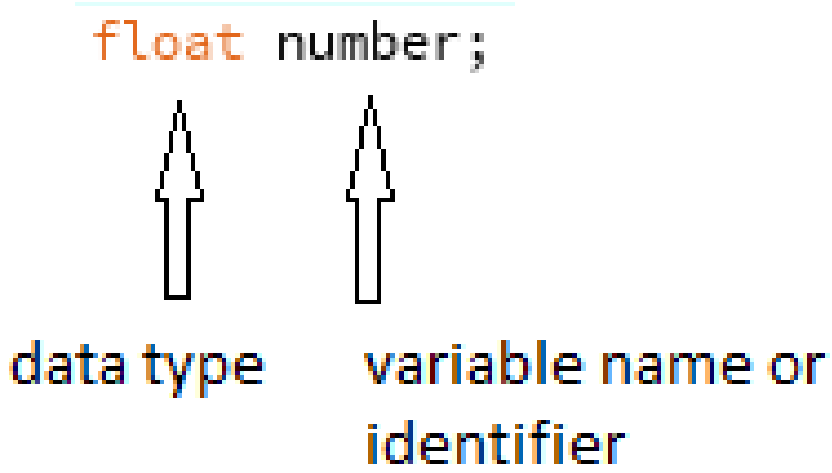
- Here are the most commonly used data types:

Data Type	Description
int	denotes an *integer* meaning that your variable can **only** take on integer values (i.e. -2, -1, 0, 1, 2, etc.) and as such may only be accepted by certain functions capable of accomodating that specific type
float	Number with decimals (can be thought of as a number with a "floating" *decimal point*)
char	one single character (could be anything on your keyboard)
String	a *sequence or "string" of characters* (note capital S)
boolean	either *"true" or "false"* and are often used to make decisions

- Most other types are extensions of these, with either a greater or smaller range

Declaring the variable

- To declare a variable (i.e. bring it into existence), follow this syntax:



Initializing the variable

- To initialize your variable, set your variable equal to some initial value after declaring it

Why initialize?

- When you create/declare a variable, the computer allocates some memory to store it
- Sometimes, the memory it dedicates to your variable already holds other data (leftover from some previous process), but that doesn't stop it from being used
- If you don't assign any value to your variable (i.e. initialize it) and use it in your program, the computer will use the leftover data in the memory that it was assigned
- Initializing the variable ensures that the above phenomena doesn't occur

Initializing the variable (2)

- If you don't know what value to assign your variable, simply put some default value like 0 (in the case of int, float or double)
- Initialize it by assigning a value to it like this (provided that it's already been declared earlier):

```
number = 0;
```

- The value you assign to your variable must be appropriate for the data type

```
String myName;
myName = "Mary's";

char firstInitial;
firstInitial = 'M';

boolean lovesCoding;
lovesCoding = true;
```

Initializing the variable (3)

- When declaring and initializing variables, there are some special rules to keep in mind
- For a char data type, the character must be in single quotations ‘ ’
- For a String data type, the word or sequence of characters must be in double quotations “ ”
- **Declaring and initializing variables can be done in one step as shown below:**

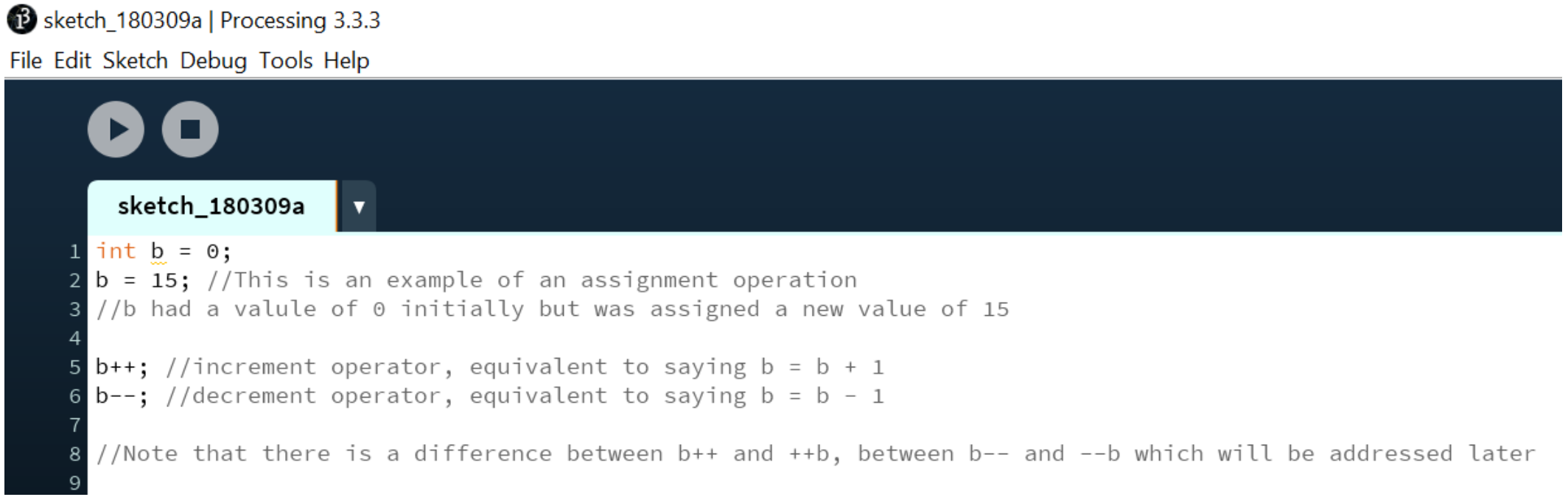
```
float number = 0;

String myName = "Mary's";

char firstInitial = 'M';

boolean lovesCoding = true;
```

Using the variable

- How to use your variable depends on the data type
- Usage revolves around using the data stored in a variable or modifying it to take on a new value
- Do not specify the data type (e.g. int, boolean, char, etc.) of a variable more than once (the compiler will not allow this)
- Call/use your variables by using their identifier/name

Modifying variables

- Variables can be changed by:
  - Assigning values to them
  - Incrementing or decrementing them, or performing some other operation



Number examples

- When your variable is of type int, float, or double, you're essentially dealing with regular numbers
- An int can take on any integer or whole number value
- Remember that integers can't have decimals and PEDMAS applies if performing several operations at once

Number examples (2)

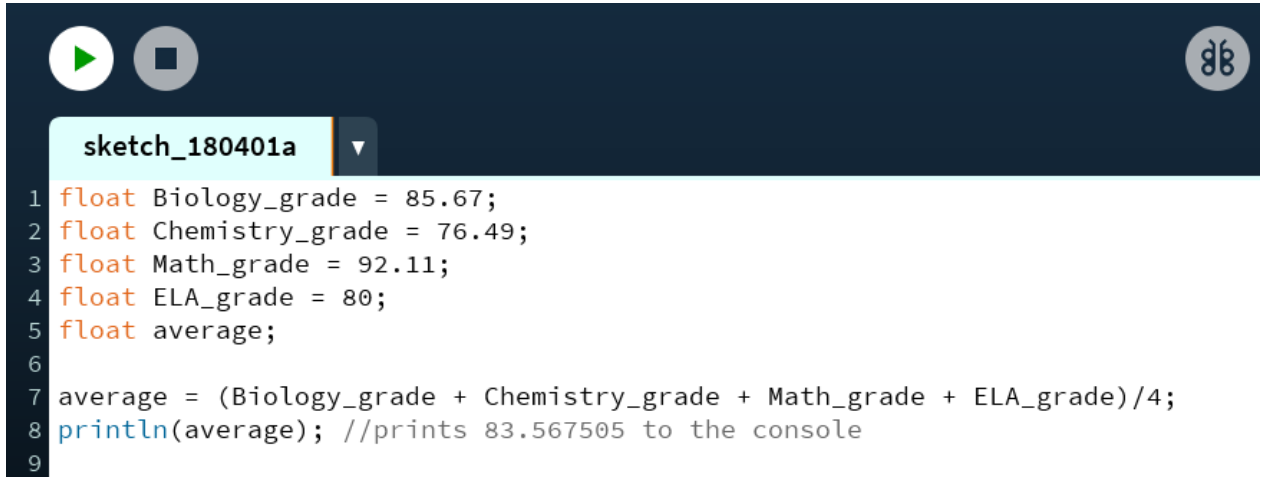
- For example, the current year, someone's birth year, age, or student ID are types of information that would be saved as integers in computer memory (discrete data)

Number examples (3)

- A float or double is used for numbers that require decimal places (continuous data)
- The only difference between a float and a double is that a double can take on bigger or more precise values
- An example of data saved under this form would be someone's bank account balance, academic average or commute time

```
sketch_180326a
1  float totalCommute = 0;
2  float busRide = 20;
3  float trainRide = 30.2;
4  float walkFromTrain = 7.6;
5
6  totalCommute = busRide + trainRide + walkFromTrain;
7
8  println(totalCommute);
9
10
```

Number examples (4)



- These data types are used for doing all the calculations in your program
- Note that in the program implementation above, the ELA grade was declared like an integer, without a decimal
- This is allowed and is interpreted as 80.000000….

Number examples (5)

- Note: You can convert a variable from one data type to another by a method known as “type casting”

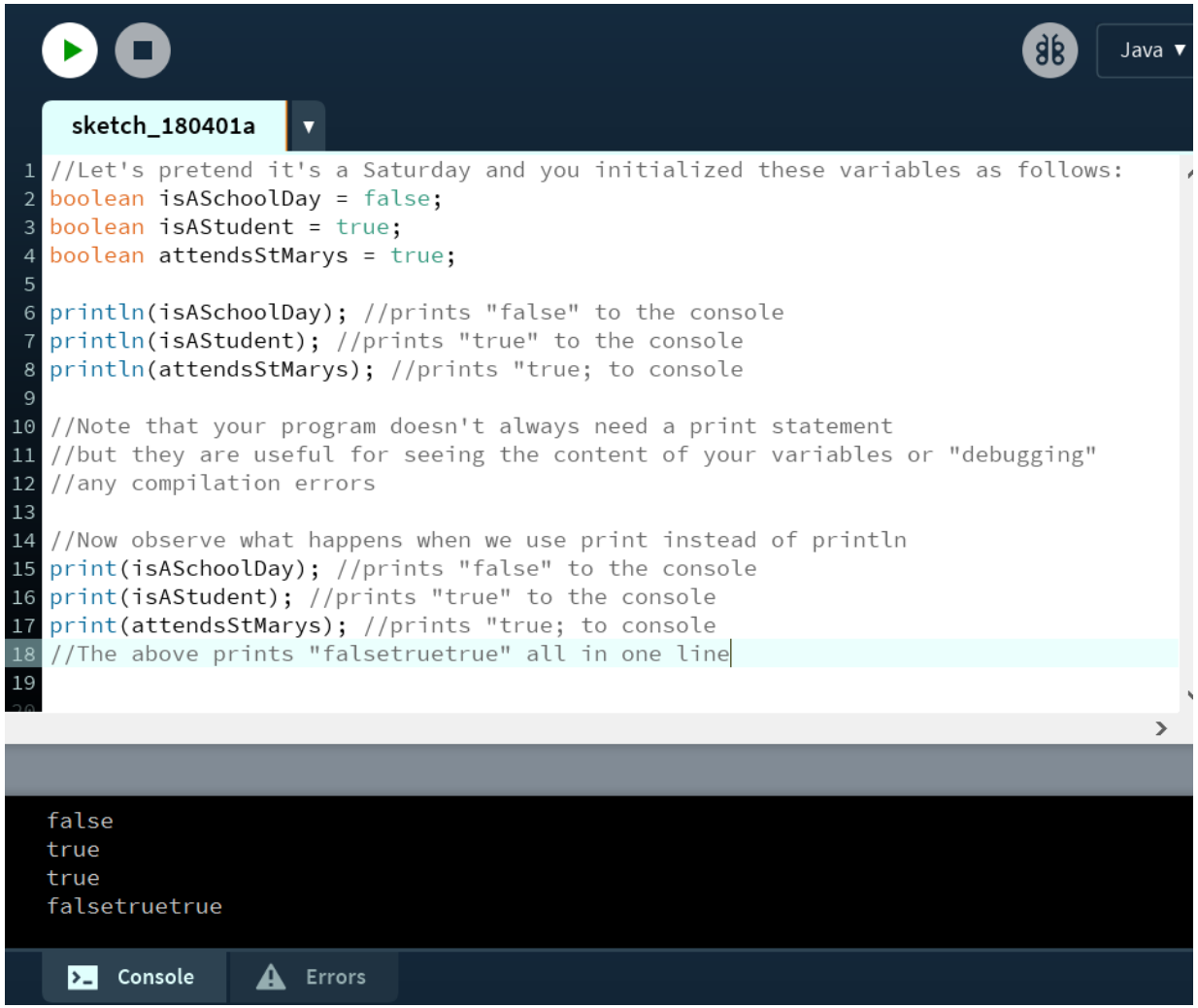
```
20
21 int number = 5;
22 float sum = (float)number + 15.0;
23 //We are not converting the variable number to a float but rather just treating it that way for this operation
24 //This is what is meant by typecasting
25 println(sum);
26
```

```
20.0
```

Logic examples

- When your variable is of type boolean, you're essentially dealing with computer logic (true vs. false)
- boolean variables are most commonly used for if() statements
- if() statements are at the core of decision making (more on this in a later crash course)
- Remember that a boolean data type can only take one of two values: true or false
- So if your data answers a true or false question, it should probably be saved as a boolean

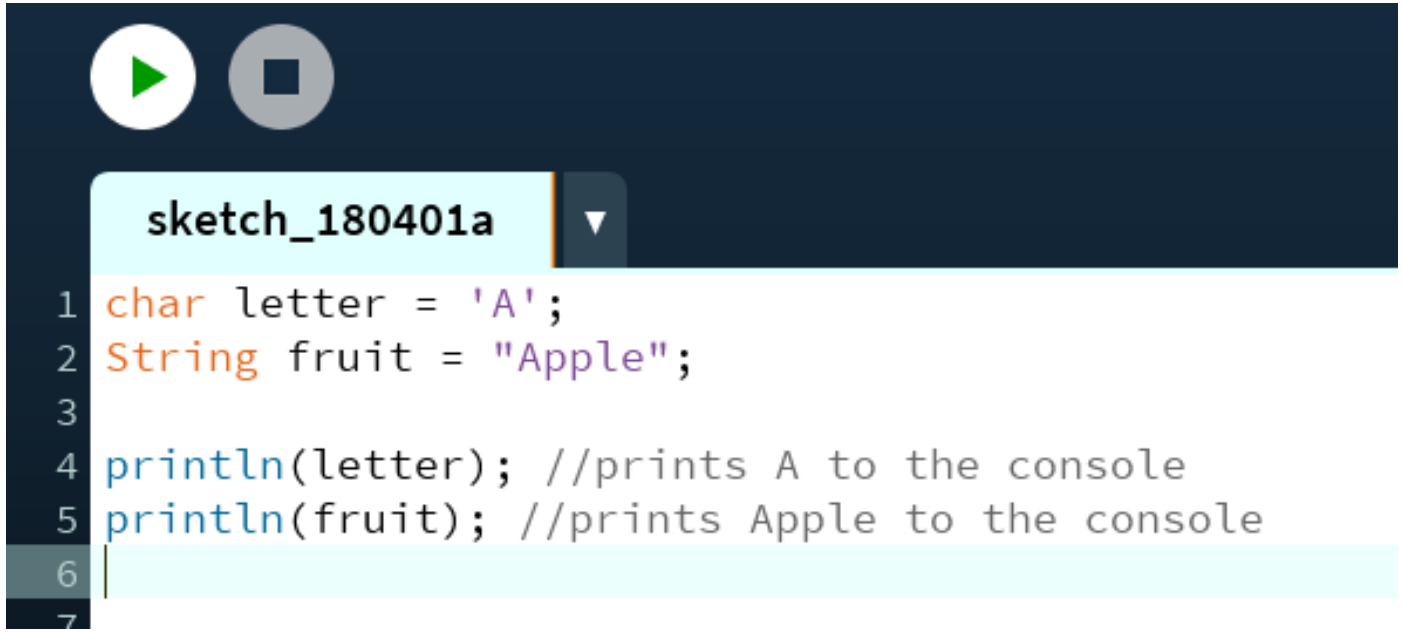
Logic examples (2)



Text examples

- When your variable is of type char or String, you're essentially dealing with text
- A char variable can be any character on your keyboard but only one of them
- Initialize a char with ' ' (single quotations)
- Think of a String as several chars, each in their own box
- We label these boxes or cells with indices, starting at 0
- Declare a String with " " (double quotations)
- Note that the word String in Processing, when referring to the data type, has a capital S

Text examples (2)



Text examples (3)

- For example, consider the String called “Processing”

Character	P	r	o	c	e	s	s	i	n	g
Index	0	1	2	3	4	5	6	7	8	9

- There are 10 characters indexed from 0 to 9
- The dot operator (a period after the variable name) can be used to access certain information about your String variable
- .charAt() can be used to access a character at a certain index
- .length() can be used to return the length of the String

Text examples (4)

```
19
20
21 String name =  "Processing"; //Inializing a string called "Processing"
22 println(name.charAt(3)); //Printing the character at the third index
23 //Since we start with "P" at index 0, this will be the fourth letter, "c"
24
25 println(name.length()); //Printing the length of the string, which is 10 characters long
26
```

```
c
10
```

Text examples (5)

- If we now try printing the character at index 10 (which is the length of the String), look what happens

```
31
32
33 println(name.charAt(name.length()));
34
```

```
StringIndexOutOfBoundsException: String index out of range: 10
```

Text examples (6)

- Remark how the last line of code causes an "Out of bounds exception error" which prevents compilation
- This is because the length of the String is 10, and there is no character at index 10 (this would imply an eleventh letter)
- You have 10 letters, labelled with indices 0 through 9, therefore index 10 is out of range

Summary

- Great job! You now wield the power of storing information in a computer as you please! Let's summarize what we've learned today.
- In this crash course, we learned about variables and data types
- A variable is used to store a piece of information of a certain data type
- Each variable has a name, also known as an identifier
- Data types classify what kind of information is stored in a variable

Summary (2)

- Data types include, but are not limited to: int, float, char, String, and boolean
- int and float are used to store numbers
- char and String are used to store text
- boolean is used to store either true or false
- A variable can be declared and initialized all in a single line, which is the preferred method
- You have completed: **2. Variables and Data Types**
- Up next: **3. Decision Structures**