Introduction to Python

What is Python?

Python is a general purpose programming language that has a wide variety of applications in areas such as engineering, life sciences, business, economics and many more. Some of its most notable uses are for data analysis, web development, artificial intelligence, machine learning, etc. That being said, however, Python can be used for much more than this and by anyone, including you! Compared to other programming languages, Python is considered to be very human readable and easier to pick up and work with for beginners. Let’s get started!

Printing

- To have something displayed on the screen we can use the print function

  Example: `print("Hello, World!")`

- When you run a program with the above statement, the text will show up on your screen
- This is the most basic program for any language! Print the phrase/line, “Hello, World!”

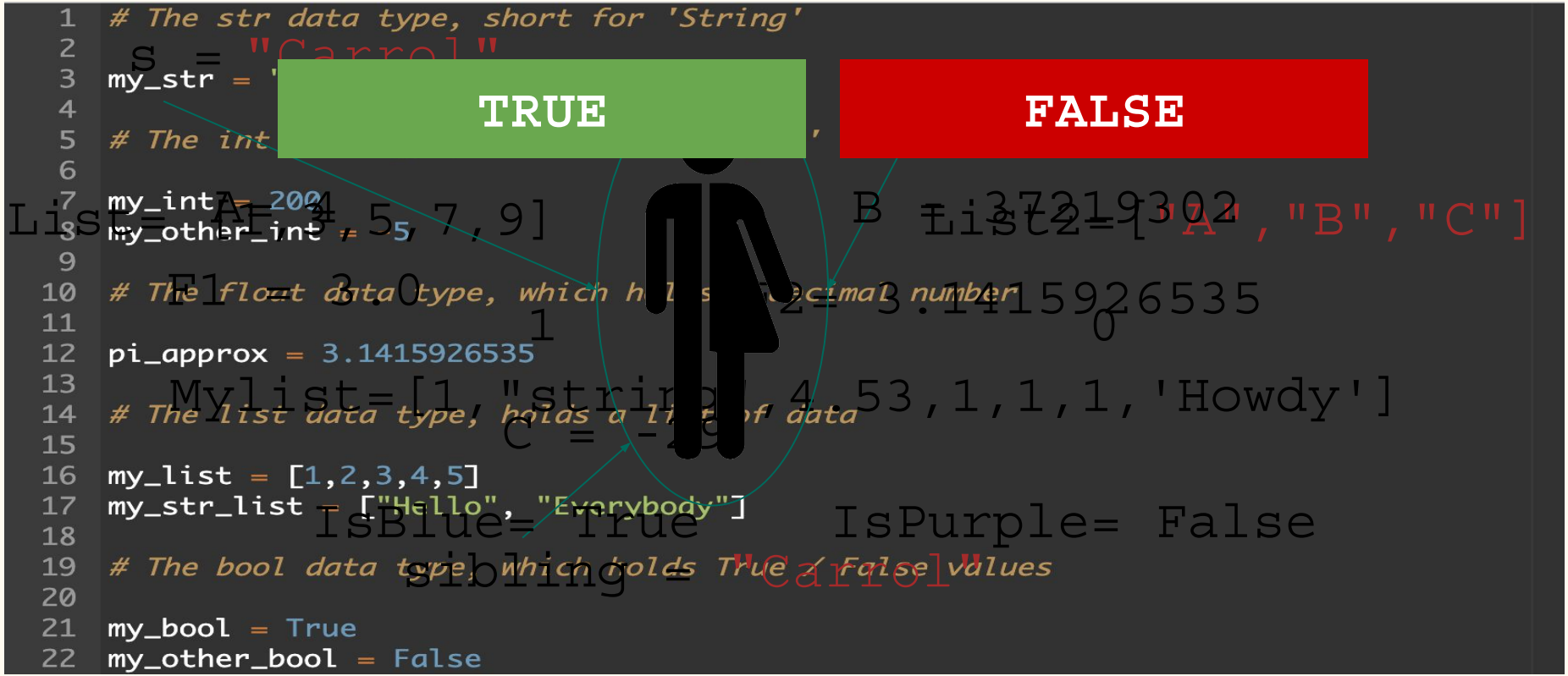
Printing demo

Link to online coding environment: Online Python Compiler (Interpreter)

```
1 print("Hello, World") # This is a comment, ignored by the program
2
3 print('Hello, World!') # You can print text to the screen using "" or ''
4
5 print("I can print anything I want in here, such as aius23qasf%%#")
6
7 # In short, you can print nearly anything you want to the screen using the print function!
8
```

Variables and data types

VariableName = data



Which, if any, of these variable names would throw an error?

1	WheneverICode	2	The Thing I do
3	1st_is_always	4	GETREADY4
5	TheFUNadventure!	6	InJ4v4

Which, if any, of these variable names would throw an error?

1 WheneverICode

2 The Thing I do

3 1st_is_always

4 GETREADY4

5 TheFUNadventure!

6 InJ4v4

Variable Example

What we want to know:

Carrol, Age 14

Arjit, Age 7

Sum of their ages

Difference of their ages

Product of their ages

Quotient of their ages

Then sum of the numbers above

?

Variable Example continued

We could do

14 +7

14-7

14*7

14/7

(14 +7)+(14-7)+(14*7)+(14/7) or 21+7+98+2

Variable Example: Manipulation of variables

BUT we can use **variables** instead!!

$$\text{Carrol\_age} = 14$$

$$\text{Arjit\_age} = 7$$

$$\text{Sum} = \text{Carrol\_age} + \text{Arjit\_age}$$

$$\text{Difference} = \text{Carrol\_age} - \text{Arjit\_age}$$

$$\text{Product} = \text{Carrol\_age} * \text{Arjit\_age}$$

$$\text{Quotient} = \text{Carrol\_age} / \text{Arjit\_age}$$

$$\text{allTogether} = \text{Sum} + \text{Difference} + \text{Product} + \text{Quotient}$$

What's the difference?

- Same number of lines
- A little more typing in the second one

We forgot Arjit just had a birthday!

So updating the first way would be updating

every

single

line

The second way just updates **one** line: the second line

String Variable Manipulation

String + String = concatenation

METHOD 1
Variables Only

```
S1 = "My name is"
S2 = "Lee"
S3= S1+S2
Print (S3)
```

METHOD 1
Strings Only

```
Print ("My name is" +"Lee")
```

METHOD 3
Strings and Variables

```
S1 = "My name is"
Print (S1,  "Lee")
```

Concatenation Result

This will make “My name isLee”

How can we fix this?

Print and Variable demo

Time for a live demo!

If statements (if, elif and else)

- If statements are used to execute different blocks of code in your program depending on certain condition(s)
- This is where we will look at boolean expressions and comparison operators
- Boolean expressions are expressions that yield either true or false, such as $5 > 2$ (true), $4 <$ -99 (false)
- Remember $\leq$ and $\geq$ from math? In programming, they are represented by $<=$ and $>=$ respectively
- Some other ther comparison operators are $<$, $>$, ‘and’, ‘or’, $==$

Difference between ‘=’ and ‘==’

- These symbols look pretty similar, though when it comes to programming they have different functionalities
- The ‘=’ operator is an assignment operator, meaning it assigns a given value to a variable
- Example: my_variable = 5 (I am setting the value of my_variable here to 5)
- The ‘==’ operator, on the other hand, is a comparison operator and checks the truth of a statement
- Example: print(5==5) would print True to the screen, while print(5==9) would print False

If Statements Demo

```
1  my_age = 22
2  your_age = 20
3
4  if (my_age >= 21 and your_age >= 21): # Will run if we are both at least 21
5      print("We can both have a drink")
6
7  elif (my_age < 21 and your_age < 21): # Will run if neither of us are at least 21
8      print("Neither of us can have a drink")
9
10 else: # Will run by default if the other two statements do not run
11     print("One of us is old enough to have a drink, the other is not")
```

- The ‘elif’ and ‘else’ statements stand for ‘else if’ and ‘else’ respectively
- If the initial ‘if’ statement did not execute, these statements are checked to see if they are true or not
- If the initial ‘if’ statement did execute, these statements are ignored and not checked

A large problem with a small solution

- Say that I wanted to print out all of the numbers between 1 and 1,000 to my computer screen using Python
- From what we have learned, that would look something like this...

```
1  print(1)
2  print(2)
3  # Skip a few thousand more...
4  print(9999)
5  print(10000)
```

- There must be a better way to do this, right???

Loops: Don’t repeat yourself

- Loops give us a quick and easy way to solve our problem in just a few lines of code!!!

```
1 i = 1
2 while (i <= 10000):
3     print(i)
4     i += 1 # This adds 1 to the current value of our int variable i
5
```

- Using a while loop (one of the two loops available to you in Python), we have written a program that prints the numbers 1 to 10,000 to our computer
- In short, this loop runs over and over again (hence the name loop) until the boolean expression (i <= 10000 in our case) is no longer true

For Loops

- For loops in Python are used to iterate over a collection of elements - I know that's a bit wordy, so I think an example will better demonstrate the idea

```
1 list_of_fruit = ["Apple", "Banana", "Orange"] # list of strings representing fruits
2 for fruit in list_of_fruit: # for each string in the list
3     print(fruit) # print the current fruit
4
```

- Using a for loop, we are going through our list, called of list_of_fruits, and we are printing out each fruit in the list. For loops in Python follow the general syntax of "for variable in collection"

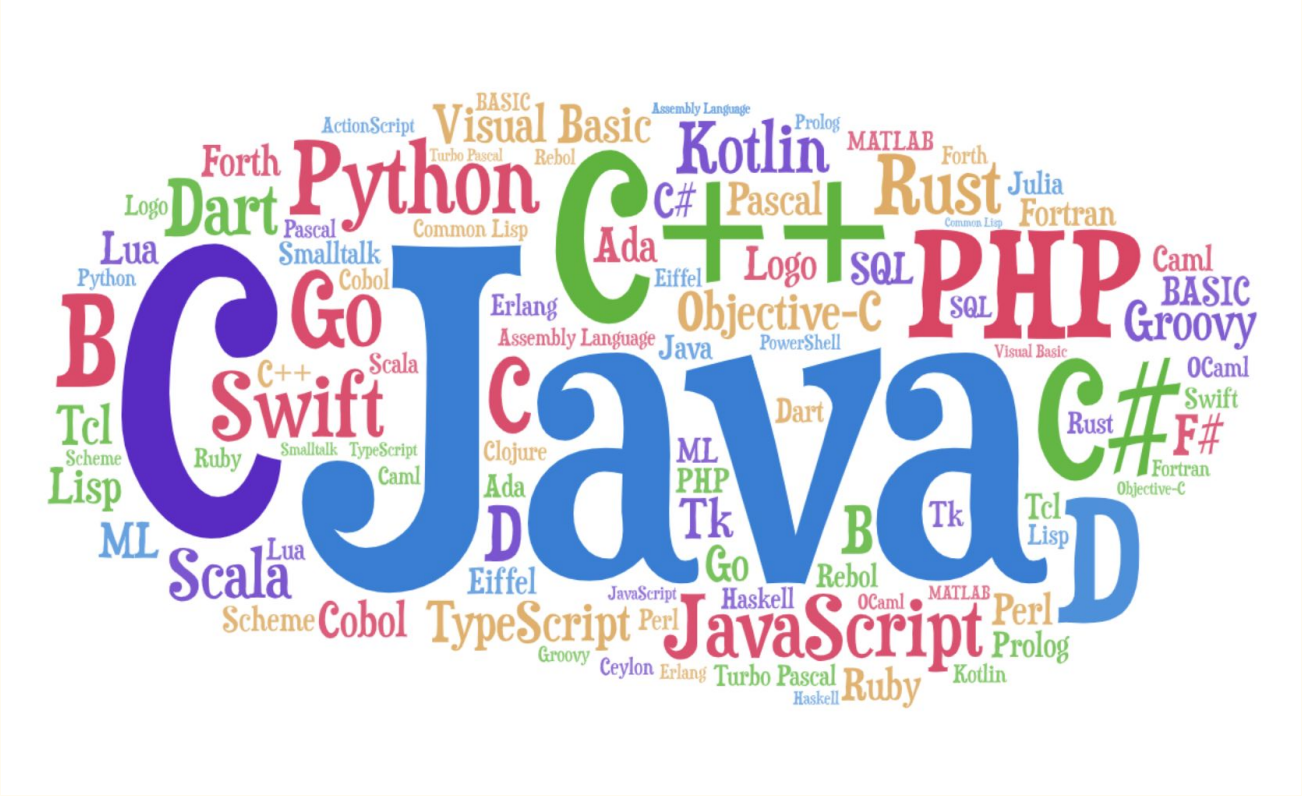
While Loops

- While loops are used to run a block of code while a given boolean expression is true (remember, a boolean expression is something like $5 > 3$ (true) or $4 < 1$ (false)
- While loops are great to use when you want to repeatedly execute a block of code while some condition is true hence the name while

```
1 my_age = 10
2 while (my_age < 22): # This loop will run while I am younger than 22
3     print(my_age) # Each time the loop runs, I am printing my age
4     my_age += 2 # After printing my age, I add two to it
```

- This program looks at my age, which is initially 10 and runs the while loop “while” my age is less than 22. The code block in the for loop prints my age to the screen and adds two to it

Python vs Others Languages



Python's (relatively) easy installation

- Download Python for Windows: Python Releases for Windows
- Download Python for Mac: Download Python
- Download Python for Linux/Unix: Python Source Releases
- If you need any help installing Python, there are tons of tutorials online to help you. Below is just one of the many I've seen:
- Python 3 Installation & Setup Guide – Real Python