**Outline**

Access the Python Development environment and follow the tutorial to gain an initial exposure to a programming language. Begin to develop an familiarity with basic programming concepts.

**Objectives**

* Use correct terminology to describe programming concepts;
* Describe the types of data that computers can process and store (e.g., numbers, text);
* Explain the difference between constants and variables used in programming;
* Use variables, expressions, and assignment statements to store and manipulate numbers and text in a program

**Materials**

* Python3 Development Environment at: //repl.it/
* Python Tutorial at:<http://www.letslearnpython.com/learn/>

**Accessing the Python3 Web IDE Environment**

Accessing the IDE

* Go to:<https://repl.it/>
* Select Python3
* Sign-up / Create an account
* Make sure you can remember your account information for the rest of the course.

Using the IDE

* Use the black area like a calculator to try simple statements or commands
* Use the white area to create programs with multiple statements

**Accessing the Tutorial**

Accessing the Tutorial

* Go to:<http://www.letslearnpython.com/learn/>
* Read up to “Lesson 3: Math”

**Level 1: Basic Math & Strings**

Access the Tutorial and start at “Lesson 3: Math”.

Questions

1. Complete “Lesson 3: Math – Math Basics” by typing the sample commands in the black area of the IDE.
   1. Create your own expression using 5 “+” and “-“ operators.
   2. List your expression and the result below.
2. Complete “Lesson 3: Math – More Operators” by typing the sample commands in the black area of the IDE.
   1. Create your own expression using 5 “\*” and “/” operators.
   2. List your expression and the result below.
3. Complete “Lesson 3: Math – More Division” by typing the sample commands in the black area of the IDE.
   1. Create one division expression that gives a whole number answer
   2. And one division expression that gives a decimal number answer.
   3. List your expressions and the results below.
4. Complete “Lesson 3: Math – Floats” by typing the sample commands in the black area of the IDE.
   1. Use the “round()” function for the expressions you created in question #3 above.
   2. List your “round()” expressions and the results they return below.
5. Read through “Lesson 3: Math – Comparison Operators”.
   1. Why do you think Equals is “==” instead of “=”?
   2. What does “=” mean?
6. Complete “Lesson 3: Math – Practice” and “Lesson 3: Math – Practice Answers” by typing the sample commands in the black area of the IDE.
   1. Create an expression using 5 different operators that returns a “True” result
   2. And an expression using 5 different operators that returns a “False” result.
   3. List your expressions and the results returned below.
7. Complete “Lesson 4: Strings – Strings” and “Lesson 4: Strings – Examples” by typing the sample commands in the black area of the IDE.
   1. Explain why typing “apple” works and why typing apple without quotes gives an error.
   2. Also explain why “2 + 5” does not equal 7.
8. Complete “Lesson 4: Strings – Operators” by typing the sample commands in the black area of the IDE.
   1. Explain why typing “appl” + “e” works and why typing “apple” - “e” gives an error.
   2. Also explain why “Hello” \* 10 works but why “Hello” / 10 does work.
9. Complete “Lesson 4: Strings – Indexes” by typing the sample commands in the black area of the IDE.
   1. List the letters in your first name and the index for each letter in your first name.
10. Complete “Lesson 4: Strings – Indexes Examples” by typing the sample commands in the black area of the IDE.
    1. Explain why print(“Hello!”[4]) does not print “l”.
    2. What does print(“Hay, Bob!”[4]) print? For a hint try print(“Hay, Bob!”[3]) and print(“Hay, Bob!”[5])
11. Complete “Lesson 4: Strings – Rules” by typing the sample commands in the black area of the IDE.
    1. Explain why print(“Hello!”[7]) gives an error.

**Level 2: Booleans & Variables**

Access the Tutorial and start at “Lesson 5: Variables”

Questions

1. Complete “Lesson 5: Variables – Save a Value” by typing the sample commands in the black area of the IDE.
   1. What do you get if you type puppies / 3? 12.0
   2. Why doesn’t typing kittens / 3 work? Because kittens is not the parable for 6\*6.
2. Complete “Lesson 5: Variables – Assign a New Value” by typing the sample commands in the black area of the IDE.
   1. Explain how the following sequence of commands works:
      1. puppies = 36 Doing this assigns the value of 36 to the word/variable puppies.
      2. puppies = puppies / 6 Here when puppies=puppies is pressed the variable puppies holds no value. However when / 6 is added the value becomes 6.
      3. puppies Now the variable holds the value of 6.
3. Read through “Lesson 5: Variables – Rules”.
4. Complete “Lesson 5: Variables – Math Operators” by typing the sample commands in the black area of the IDE.
   1. Explain what happens for following sequence of commands:
      1. colour = “red” The value of colour becomes red.
      2. puppies = 36 The value of puppies becomes 36.
      3. colour + puppies An error occurred
5. Complete “Lesson 5: Variables – String Operators” by typing the sample commands in the black area of the IDE.
   1. Explain why the following commands give different results:
      1. Color + day \* fishes Since fishes holds the value of 3, day gets multiplied by 3.
      2. ( Color + day ) \* fishes With brackets around both words, fishes multiples both colour and day by 3.
6. Complete “Lesson 5: Variables – Indexes” by typing the sample commands in the black area of the IDE.
   1. What is the index of ‘r’ in “watermelon”? The index is [4]
   2. Write an expression using mynumber to return ‘r’ fruit [mynumber+1]
7. Complete “Lesson 5: Variables – Assignments or Comparisons” by typing the sample commands in the black area of the IDE.
   1. What is the difference between “=” and “==”? “=“ means “this equals that” and “==“ means “is this thing equal to that thing?”.
   2. Create your own mnemonic to remember this difference. “==“ gives you an the answer to the question, but “=“ sets up a value for your variable.
8. Complete “Lesson 6: Errors – Examples” by typing the sample commands in the black area of the IDE.
   1. What doesn’t “friend” + 5 work? Because you cannot add a number to a string.
   2. What is the difference between int and str? “int” stands for integer and “str” stands for string.
9. Read through “Lesson 6: Errors – Parts of an Error Message”.
   1. Is “friend” + 5 an example of:
      1. A Syntax Error?
      2. A Runtime Error?
      3. A Logic Error? It is a logical error.
10. Read through “Lesson 6: Errors – Fixing Errors”.
    1. Use the ‘print’ command to print your first name and last name.

Print (“Harneet”, “Sidhu”)

1. Complete “Lesson 7: Booleans – Types of Data” by typing the sample commands in the black area of the IDE.
   1. What is the value of: type(“True”) It is a string.
   2. What is the value of: type( True ) It is a booleans.

Why is the result different? The result is different because anything with quotations around it, python will read as a string. Therefore it read the first one as a string and the second one as a booleans because it had no quotations.

1. Complete “Lesson 7: Booleans – What Is A Boolean” by typing the sample commands in the black area of the IDE.
   1. Why do you think that having a Boolean data type is important in computer programming? I think it is important to have a Boolean data type in computer programming because it tells you if what you are doing is correct or not.
2. Complete “Lesson 7: Booleans – Trying Out Booleans” by typing the sample commands in the black area of the IDE.
   1. Why do you think that there is no Maybe” Boolean data value in computer programming? I think that there is not maybe because even if only a part of it is false, it still throws off the whole thing. It should only be true when the whole thing is correct.

**Level 3: Lists & Logic**

Access the Tutorial and start at “Lesson 7: Booleans”

Questions

1. Complete “Lesson 7: Booleans – AND Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. True and True. True
      2. True and False False
      3. False and True False
      4. False and False False
   2. Explain if there are any other combinations of True / False.

I don’t think there are any other combinations.

* 1. Explain how the AND operator is similar to a math operator and how it is different.

The AND operator is similar to a math operator because even in the math operator if there is one mistake the whole expression is false and the two are different because Booleans compares things whereas math operators do math.

1. Complete “Lesson 7: Booleans – OR Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. True or True True
      2. True or False True
      3. False or True True
      4. False or False False
   2. Explain how the OR operator is similar to the AND operator and how it is different.

The OR operator is similar to the AND operator because when both are false the result comes out as false. It is different because if there is at least one true than the result is true.

1. Complete “Lesson 7: Booleans – NOT Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. not (True or True) False
      2. not (True or False) False
      3. not (False or True) False
      4. not (False or False) True
   2. Explain how the combination of the NOT & OR operators is similar to the AND operator by itself and how it is different.

It is the similar because it gives the same answer on (True or False) and (False and True) as AND being on its own. It is different because the answers for (True or True) and (False or False) are switched.

1. Complete “Lesson 7: Booleans – Expressions” by typing the sample commands in the black area of the IDE.
   1. Explain why the following two Python statements give different results.
      1. not (True or True)
      2. not True or True

The two statements give different results because one has brackets around it but the other does not. With the brackets the whole statements become false, but without them only the first true becomes false, and with an OR statement we know if there is at least one true then the whole statement is true.

* 1. Explain why the following two Python statements give the same results.
     1. not (True and True)
     2. not True and True

Both of these python statements give the same results because the not before the brackets makes the whole expression False. The second is also false because the not in front of the first True makes it false , and when we compare with AND, even if one of them is false that makes the whole expression false.

1. Complete “Lesson 7: Booleans – Practice” by typing the sample commands in the black area of the IDE.
   1. Create three more practice expressions similar to those in the tutorial.
2. not True and False
3. 3!=0 and 1==1
4. not (False or True)
   1. Provide the results for your practice expressions
5. False
6. True
7. False
8. Complete “Lesson 8: Lists – A Collection of Objects” by typing the sample commands in the black area of the IDE.
   1. Create a list of your favorite sports teams.

teams=[“toronto raptors”, “toronto blue jays”, “toronto maple leafs”]

* 1. Assign your list to a variable.

The variable I used was teams.

* 1. Confirm that your variable and your list are the same.

list(teams) => [‘toronto raptors’’, ‘toronto blue jays’, ‘toronto maple leafs’]

1. Complete “Lesson 8: Lists – List Indexes” by typing the sample commands in the black area of the IDE.
   1. What is the list index of the last team in your list of favorite sports teams.

The index of the last team in my list is 2 for ‘toronto maple leafs’

* 1. In the tutorial, the error produced by typing “fruit[3]” is an example of:
     1. A Syntax Error?
     2. A Runtime Error?
     3. A Logic Error?

It was a Syntax error.

1. Complete “Lesson 8: Lists – Practice” and “Lesson 8: Lists – Practice Answers” by typing the sample commands in the black area of the IDE.
2. NOTE: Starting with Lesson 9 you should use the WHITE area of the IDE for entering example code with multiple statements.
3. Complete “Lesson 9: Logic – Making Decisions” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code to print “Hi Alfred!” based on a decision using numbers

myname = “2”

If myname == “2”:print(“Hi Alfred”)

1. Complete “Lesson 9: Logic – Adding A Choice” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code to print your first name or your last name based on a choice (using “else”).

myname= “Harneet”

if myname== “Harneet”: print(“Harneet”)

else: print(“Sidhu”)

1. Complete “Lesson 9: Logic – Adding Many Choices” and “Lesson 9: Logic – Practice” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code and “elif” statements to make a choice using at least 4 of your friends names.

if myname == “Harneet”: print(“Hi Harneet!”)

elif myname == “Sareena”: print(“Hi Sareena”)

elif myname == “Khushi”: print(“Hi Khushi”)

elif myname == “Kirishnavy”: print(“Hi Kirishnavy”)

else: print(“Who are you?!?”)