**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.

**10+20 = 30**

**20-10 = 10**

1. 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.

**5 \* 2 = 20**

**10/5 = 2**

1. 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.

**3a). 10 / 2 = 5**

**3b). 10 / 4 = 2.5**

1. 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.

**Round (10/4) = 2**

**Round (10/2) = 5**

1. Read through “Lesson 3: Math – Comparison Operators”.
   1. Why do you think Equals is “==” instead of “=”?

**“==” compares the two numbers to see if it is true or false**

* 1. What does “=” mean?

**“=” means to assign something as something else/it is an Assignment Operator**

1. 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.

**True : 1==1 False: 4==6**

**True: 2\*2==2\*2 False: 5\*10==123**

**True:3/3==3/3 False: 123/12=1864**

**True: 56-10==46 False: 10-10==20**

**True: 10+10==20 False: 23+12=1412**

1. 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.

**For Python to read the string it must be in quotations**

* 1. Also explain why “2 + 5” does not equal 7.

**“2+5” does not equal 7 because they are in a string not as a separate equation**

1. 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.

**“Appl” + “e” works because the program can add a letter to the string but it does not allow to subtract the letter from the word**

* 1. Also explain why “Hello” \* 10 works but why “Hello” / 10 does work.

**“Hello” \* 10 works because it is just multiplying the same word 10 times but cannot divide hello by 10 since it has no numbers**

1. 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.

**A r j u n**

**0 1 2 3 4**

1. 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”.

**Because the indexes start counting from 0 and the 4th letter would be “o”**

* 1. What does print(“Hay, Bob!”[4]) print? For a hint try print(“Hay, Bob!”[3]) and print(“Hay, Bob!”[5])

**print(“Hay, Bob!”[4]) responds with a space while print(“Hay, Bob!”[3]  
comes out with a “,”**

1. 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.

**Print(“hello!”[7]) gives an error because there are only 5 characters (starting from 0)**

**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?

**When puppies/3 is typed it comes back as an err**or

* 1. Why doesn’t typing kittens / 3 work?

**When typing kittens/3 it does not work because a word cannot be divide but can be multiplied and added instead.**

1. 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:
      * puppies = 36
      * puppies = puppies / 6
      * puppies

**When putting in puppies=36 it tells the programing that 36 is the equaling of puppies, which means puppies equals 36. Puppies/6 basically means 36/6 as puppies is now equal to the number 36.**

1. Read through “Lesson 5: Variables – Rules”.
2. 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:
      * colour = “red”
      * puppies = 36
      * colour + puppies

**The first command makes the word “colour” be assigned the value ‘red’ while puppies = 36 makes the word “puppies” equal to 36. The last command gives an error because you are unable to add 36 to “red”**

1. 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:
      * Color + day \* fishes
      * **redmondaymondaymonday**
      * **The expression is color plus day times fishes. We know that color is yellow, day is Monday, and fishes is three. The multiplication starts first because order of operations and BEDMAS.**
      * ( Color + day ) \* fishes
      * **yellowMondayyellowMondayyellowMonday**
      * **This expression starts off within the brackets first because of BEDMAS and Order of Operations**
2. 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”?
   2. Write an expression using mynumber to return ‘r’

**The index of r would be fruit[4]**

**Expression=fruit[mynumber+1]**

1. 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 “==”?
   2. Create your own mnemonic to remember this difference.

**“=” is an assignment operator which assigns one thing to another of value**

**“==” is a comparison operator which compares the two things to see if it true of false**

1. Complete “Lesson 6: Errors – Examples” by typing the sample commands in the black area of the IDE.
   1. What doesn’t “friend” + 5 work?

**It does not work because the word “friend” cannot be added but can only be multiplied**

* 1. What is the difference between int and str?

**An integer has a numeric value while a string is any characters in quotations.**

1. 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 Syntax Error**

1. Read through “Lesson 6: Errors – Fixing Errors”.
   1. Use the ‘print’ command to print your first name and last name.

**Print(“Arjun 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”)

**<class ‘str’>**

* 1. What is the value of: type( True )

**<class ‘Bool’>**

* 1. Why is the result different?

**The result is different because type(“True”) has quotations around the characters which python identifies as a string while type(True) is without strings and first letter is capital which python identifies as a Boolean**

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?

**Having Boolean data type is important in computer programming because they make decisions about what to do in the code**

1. 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?

There is no “maybe” in Boolean data because of logic

**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.

**Do not know of any other True and False combinations**

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

**The AND operator is similar to the Math operator because they both compare the results and see if the two variables are true or false. They are different because the Math operator has different numeric results and more operations while the AND operator is a part of the Boolean rule and only spits back “false” except for True and True.**

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 and the AND operator are different because the OR operator results in 3 trues and 1 false (False or False = False) while the AND operator is the same but opposite. It results in 3 false and 1 true (True and True = 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 are similar to the AND operator and how it is different.

**The combination of the NOT & OR operators are similar to the AND operator because the results are the same (3 false and 1 true) but the sequence used to get the answer is flipped because of the NOT operator.**

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 following two Python statements give different results because in the first one, BEDMAS is involved so the brackets are done first then the NOT operator comes in. The result of inside the brackets is “True” but with the NOT operator the result will become “False”. The second one has no brackets so it is done from right to left**: **not True or True == False or True == True. The result is “True” because of the OR operator.**

Explain why the following two Python statements give the same results.

* + 1. not (True and True)
    2. not True and True

**The following two Python statements give the same results because in the first one, BEDMAS is involved which means the brackets will be done first, (True and True) which comes out as true. Then the NOT operator comes in and makes the True into False. The second statement starts from the NOT operator making the first True = False which then makes the equation, “False and True”. By the AND operator rule the result will be “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. Provide the results for your practice expressions

**5==25 – False**

**421 != 12 – True**

**“Arjun” != “Arjuns” – True**

1. 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.
   2. Assign your list to a variable.
   3. Confirm that your variable and your list are the same.

**Favouriteteams = [“Raptors” , “Cleveland” , “Celetics” , “Lakers” , “Spurs”]**

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 team

**The list index of the last team is [4]**

* 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?

**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.

NOTE: Starting with Lesson 9 you should use the WHITE area of the IDE for entering example code with multiple statements.

1. 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 = 3+2

if myname == 5:

print ("Hi Aflred!")

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 = "someonerandom"

if myname == "arjun":

print ("hello imposter")

else:

print("hello Arjun")

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.

myname = "arjun"

if myname == "arjun":

print ("hello arjun")

elif myname == "Jordan":

print("hey Jordan")

elif myname == "Christan":

print("Whats Up Christan?")

elif myname == "Sahaj":

print("hi Sahaj!, nice weather")

elif myname == "Sanvir":

print("hey Sanvir!")

else:

print("hello imposter")