**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.

3+4=7

12-7=5

9+1=10

6-2=4

7-2=5

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.

55/5 =11

14\*3=42

54/9=6

7\*101=707

80/40=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.

144/12=12.0

75/6=12.5

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(144/12)= 12

round(75/6)=12

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

Because “==”is used to see if the answer is actually the variable in the expression or not whereas “=” just means that the answer must be the variable in the expression

* 1. What does “=” mean?

It means that the term on the left side of the equal sign is the same as the term on the right side of the equal sign.

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.

3+4=7 -> True

12-7=5 -> True

9+1=10 -> True

14\*3=42 -> True

54/9=6 -> True

3+4=9 -> False

12-7=465 -> False

9+1=8 -> False

14\*3=0 -> False

54/9=1 -> False

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.

When typing “apple”, the quotation marks allow Pthon to read the string, However without the quotation marks, the string is considered as an error as Python is unable to read it.

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

“2+5” cannot be equal to 7 as the expression is only being read as a string by Python. The expression cannot be added to give a value of 7 unless the expression is without the quotation marks.

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.

The process, Concatenate, allows strings to be put together side by side. However, removing string out of another string cannot be done as the process allows strings to be put together.

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

“Hello”\*10 works because multiplying controls the number of times the string can be shown. However, decreasing the number of times a string can be shown is an error as there is only one string.

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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **M** | **u** | **k** | **t** | **i** | **k** | **a** |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** |

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 “l” has an index of 3 and the first letter from the string is counted starting with 0.

* 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]) is the space in between the coma and “B”.

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.

Because there is no seventh position in this string.

**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?

Because puppies was given a value of 6\*6 (36).

* 1. Why doesn’t typing kittens / 3 work?  
     Because no value was assigned to kittens.

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

Puppies was assigned a value of 36 at first. Then it was reassigned with another value which is the first value divided by 6. Therefore the new value was given when “puppies” was typed in.

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

Colour was assigned the string of “red” and puppies was assigned the value of 36. When both colour and puppies were added together, the result proved to be an error.

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
      * ( Color + day ) \* fishes  
        In the result of the first string, the command given was to put the colour and three times of the day together. However, the result for the second string, the command given was to put three pairs of the colour and day string together.
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”?

Since fruit is assigned to be “watermelon”, then the index of “r” in “watermelon” is fruit[4].

* 1. Write an expression using mynumber to return ‘r’  
     If my mynumber is 3, then the index of “r” in “watermelon” is 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 “==”?

“=” is used when assigning a value or string to something whereas “==” is to find out if something is equal to a value or not.

* 1. Create your own mnemonic to remember this difference.  
     “=” Me=3

“==” Me==3 -> True

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?

Because “friend” is a string and 5 is a value

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

Int means integer such as 3, 4, 5 and many other numbers where str means string such as “friend”, “hello” and many other words with quotation marks.

1. Read through “Lesson 6: Errors – Parts of an Error Message”.
   1. Is “friend” + 5 an example of:
      1. A Syntax Error? Yes
      2. A Runtime Error? No
      3. A Logic Error? No
2. Read through “Lesson 6: Errors – Fixing Errors”.
   1. Use the ‘print’ command to print your first name and last name.

firstname=“Muktika”

lastname=“Mundboth”

print(firstname+lastname)

MuktikaMundboth

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 )

Error

* 1. Why is the result different?

The results are different because the first data is a string whereas the second one is none of the data.

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?

Because Boolean data type allows us to make decisions for the future steps in our 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?

Because “maybe” is not a clear answer that will help in making decisions. It is neither positive nor negative. So it makes us unsure about what decisions to take in our code.

**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.

There are no other combinations of True/False

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

Examples of how AND operator is similar to math operator:

2==2 True

3==8 False

3<=3 True

2!=7 True

True and True True

True and False False

False and True False

False and False False

The difference between AND operator and math operator is that math operators can give different results that depend on the operator used. However for AND operator, the result depends if any of the values are wrong or not.

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.

Both of them give either an answer of true or false. But for AND operator, if any of the values are wrong, the answer would false and for OR operator, if any of the values are wrong, then the answer would still 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.

Similarities: Have either an answer to True or False

Differences: NOT & OR operators use “or” when comparing whereas AND operator uses “and” when comparing.

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) - It means that the answer is either not True or not True which leaves us with the answer of False
      2. not True or True - It means that the answer is False or True which leaves with an answer of True.
   2. Explain why the following two Python statements give the same results.
      1. not (True and True)
      2. not True and True

When using AND operator, the answer for the comparisons gives us False as both comparisons indicate that they have a wrong result.

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

True and True -> True

3==3 or 4==9 ->True

not(1!=0 or 2==2) ->False

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.

[“Manchester United”, “Blue Jays”]

* 1. Assign your list to a variable.

sportsteams = [“Manchester United”, “Blue Jays”]

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

They are the same

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.
   2. In the tutorial, the error produced by typing “fruit[3]” is an example of:
      1. A Syntax Error? No
      2. A Runtime Error? No
      3. A Logic Error? Yes
2. Complete “Lesson 8: Lists – Practice” and “Lesson 8: Lists – Practice Answers” by typing the sample commands in the black area of the IDE.

colours=[“red”, “blue”, “green”]

print(colours[0])

red

print(colours[1])

blue

print(colours[2])

green

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

myfavouritenumber = 7

if myfavouritenumber == 7:

print("my favourite number is 7")

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 = "Muktika"

if myname == "Muktika":

print("Hi Muktika")

else:

print("Stop lying impostor")

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 = "Muktika"

if myname == "Muktika":

print("Hi Muktika")

elif myname == "Abira":

print("Hey Abira")

elif myname == "Vinne":

print("Hey Vinne")

elif myname == "Hasrat":

print("Hi Hasrat")

else:

print("Well, hello there :)")