**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. 5 + 1 – 6
3. 0
4. 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.

a. 5 \* 30 / 6

b. 25.0

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.

a. 30 / 6

c. 5.0

b. 10 / 3

c. 3.3333333333333335

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.
2. 30 / 6
3. 5
4. 10/3
5. 3
6. Read through “Lesson 3: Math – Comparison Operators”.
   1. Why do you think Equals is “==” instead of “=”?
   2. What does “=” mean?

a b. “=” is used to help a variable either letter or entire word be represented by a number. “==” is meant to show is that variable equal to another number

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.
2. answer == ( 5 \* (6 + 8) / 14 + 6)

c. TRUE 9

1. answer == ( 3 + 1 – 2 \* (58 / 4))
2. FALSE -25.0
3. 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
4. Apple is not defined.
5. Also explain why “2 + 5” does not equal 7.

b. It is written in quotes which will cause python to write the expression not answer it.

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.

A. It does not mention why.

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

B. You can’t divide a word.

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.

N A M E

0 1 2 3

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”.
   2. The fourth letter is “o” since python starts counting from 0.
   3. What does print(“Hay, Bob!”[4]) print? For a hint try print(“Hay, Bob!”[3]) and print(“Hay, Bob!”[5])
   4. “b”
2. 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.
   2. There are less than 7 letters,

**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?
   2. error
   3. Why doesn’t typing kittens / 3 work?
   4. Kittens is a variable without an assigned value
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:
      * puppies
      * puppies = 36
      * puppies = puppies / 6

Variable get named, then it is assigned a value, and then used in an expression.

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

Error. You can’t add a colour to a number.

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  
        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. 4
   3. Write an expression using mynumber to return ‘r’
   4. Watermelon[mynumber – ]
3. 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. One is equal, the other is equal to.
   3. Create your own mnemonic to remember this difference.
   4. = equal, == equal to
4. Complete “Lesson 6: Errors – Examples” by typing the sample commands in the black area of the IDE.
   1. What doesn’t “friend” + 5 work?
   2. Friend in in quotation marks, or hasn’t been assigned a value.
   3. What is the difference between int and str?
   4. Int = integer, str = string
5. 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?

syntax

1. Read through “Lesson 6: Errors – Fixing Errors”.
   1. Use the ‘print’ command to print your first name and last name.
   2. Print(“First Last”)
2. 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”)
   2. Class string
   3. What is the value of: type( True )
   4. error
   5. Why is the result different?
   6. True is undefined

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?
   2. we use them in programming a lot when we need to make decisions about what to do in our code
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?
   2. An answer can either be right or wrong with A I, they are not programmed to compute queries that could possibly be both.

**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.

None

* 1. Explain how the AND operator is similar to a math operator and how it is different. It will tell if both strings are correct or not.

t does not however give an answer to why, only that the string is true or false.

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.

They are similar because they compare two or more expressions. They will answer with either true or false depending on the string. IF both options are true, or both are false, they will answer with “True” or False” respectively. The difference lies with how they answer other strings. In the “and” comparison, both expressions must be true in order for it to respond as “True”, otherwise it will always respond with “False”. In contrast, the “or” operator will always respond with “True” if at least one of the comparisons are correct, otherwise it will respond with “False”.

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.

They are similar because they compare two or more expressions. They will answer with either true or false depending on the string. The difference is that by using” not” and “or”, you get the exact opposite results as using the “and” operator by itself.

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) the not operator reverses the answer in the bracket
      2. not True or True it will still say true since one of them is true
   2. Explain why the following two Python statements give the same results.
      1. not (True and True)
      2. not True and True

the not operator overrides the and operator and reverses the answer.

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.

“answer” == “answer” “answer” == “answers” not(1==1 or 2==3)

* 1. Provide the results for your practice expressions

True False 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.

N/A

* 1. Assign your list to a variable.

N/A = 1

* 1. Confirm that your variable and your list 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.

0 1 2

* 1. In the tutorial, the error produced by typing “fruit[3]” is an example of:

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.

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

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”).

"**If** we have a name, I’ll be Not.  
Or **else** I'll be Applicable."

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 “else” statements to make a choice using at least 4 of your friends names.

"**If** we have name someone, we must call him this is.  
Or **else if** call him very!  
Or **else if** call him not!  
Or I’ll just call him applicable."