**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?

The double equal sign works because it indicates whether a statement is true or not and the single equal sign means not equal to whereas the double equal sign means equal to

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. 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.

Typing apple with quotes around it works because the program is used to words with quotes around it since it can define it but typing apple without quotes comes up as an error since the program language is used to quotes around words. Typing “2+5” doesn’t equal seven because Repl will treat it like a word because it has quotes around 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.
   2. Also explain why “Hello” \* 10 works but why “Hello” / 10 does work.

Typing “appl” + “e” works because the program treats it as two words and since there’s an addition sign it’ll concatenate. Typing “apple” – “e” doesn’t work because it comes as an unsupported operand type. Typing “hello” \* 10 works since it can be repeated but typing “hello” / 10 doesn’t work because a word cannot be split a certain amount of times.

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.
2. 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])

It doesn’t print the exclamation mark because at the beginning of this string, ‘H’ is at the index or position zero, then index 1 will be ‘e’, indexes 2 and 3 will be l and l and index 4 will be the letter o and since there’s 4 in the statement, it will come out as just ‘o’. Typing print(“Hay, Bob!” [4]) prints the letter ‘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.

It gives an error because the string index is out of range since the word “Hello” only has 4 indexes.

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

You get a name error since the word puppies is not defined. Typing kittens/3 doesn’t work because of the same reason, it gives an error since the word kittens is not defined.

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

The first sequence is basically assigning a value to the word puppies, then Repl takes the value of puppies and registers it as 36 so typing puppies = puppies / 36 will basically be 36/6 and typing just puppies with no quotes, Python treats it as a string so it knows that it’s a variable and has a value.

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

First up is assigning the word colour as red and then assigning the value of 36 to the word puppies and then adding the two together comes up as 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

It gives different results because the commands are being typed in differently. One of them has brackets around it and the other one doesn’t

1. 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’ in the word “watermelon” is 4 because the beginning starts with 0 which is the letter ‘w’ then counting in the sequence, the letter ‘r’ will be the index of 4.

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.

The difference between the equal signs are that when assigning a value, it’s saying that this equals that and that’s a short sentence so it requires one equal sign. Using two equal signs means comparing values which is a longer sentence so it requires two equal signs. Short sentences require one equal sign and longer sentences require two equal signs.

1. Complete “Lesson 6: Errors – Examples” by typing the sample commands in the black area of the IDE.
   1. Why doesn’t “friend” + 5 work?
   2. What is the difference between int and str?\

“Friend” + 5 doesn’t work because the program says that it must be a str, not an int. And Python isn’t able to concatenate the two objects are two different types of data. The difference between int and str is that an int is an integer and an str is a string and they’re two different types of data.

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?

Typing “friend” + 5 gives a TypeError in the black area since it can’t concatenate two different types of data.

1. Read through “Lesson 6: Errors – Fixing Errors”.
   1. Use the ‘print’ command to print your first name and last name.
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. What is the value of: type( True )
   3. Why is the result different?

It comes as <class ‘str’> for, type(“True”). For type(true), it comes up as an error. The result is different because one statement has quotations around it and the other one has no quotations around it since Repl registers words that have quotations around them.

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?

Booleans are a type of data used in programming when we need to make a lot of decisions about what to do 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?

Maybe because Python knows that it should treat them like Booleans instead of strings or integers.

**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
      2. True and False
      3. False and True
      4. False and False
   2. Explain if there are any other combinations of True / False.
   3. Explain how the AND operator is similar to a math operator and how it is different.

There are no other combinations of true/false. The AND operator and the math operator are similar because in both these lessons, there are statements that are either true or false and it uses numbers to compare the statement. The two operators are different because they’re two different types of data used in computer programs.

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
      2. True or False
      3. False or True
      4. False or False
   2. Explain how the OR operator is similar to the AND operator and how it is different.

The AND & OR operators are similar because they both compare statements to see whether they are true or not. They are different because of the words used in between the statements.

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)
      2. not (True or False)
      3. not (False or True)
      4. not (False or False)
   2. Explain how the combination of the NOT & OR operators is similar to the AND operator by itself and how it is different.

The combination of the NOT & OR operators are similar to the AND operator because they all compare statements. The NOT & OR operators are different because the NOT operator can reverse the outcome of statement and make a true statement, false.

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 NOT comparison operator can reverse the result of a statement or any time you put not in front of a comparison. Any expression that is true can become false. Also the OR comparison operator makes these results different because, if at least one part of the expression is true, then the statement is true. If both are false, then it is false.

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

These expressions give the same results because it has the NOT in front it makes the statement from being true to 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

Some practice expressions came up as true some came up as false and some came up as an error.

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.
2. 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?
      2. A Runtime Error?
      3. A Logic Error?

The list index of the last team in my list of favourite sports teams is 2. In the tutorial, the error produced is 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
2. 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”).
3. 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.