**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.

**5+4+3+2+1-1-2-3-4-5+6**

**6**

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.

**8\*4\*5\*9\*9\*1/3/7/8/4/1**

**19.285714285714285**

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.

**10/5**

**2.0**

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

**This is because “=” is used to assign values to a variable**

* 1. What does “=” mean?

**“=” are generally used in assignment statement to assign variables a value.**

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

**1 + 1 - 1 \* 1 / 1 == 1 / 1 \* 1 - 1 + 1**

* 1. And an expression using 5 different operators that returns a “False” result.

**5 \* 5 + 5 / 5 - 5 + 1 == 1 + 5 - 5 / 5 + 5 \* 5**

* 1. List your expressions and the results returned below.

**1 + 1 - 1 \* 1 / 1 == 1 / 1 \* 1 - 1 + 1**

**True**

**5 \* 5 + 5 / 5 - 5 + 1 == 1 + 5 - 5 / 5 + 5 \* 5**

**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.

**This happens because if I want python to read a string it must be in quotes.**

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

**This happens because since I’ve put all the values within the quotes it gets read as a string rather than an operation.**

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.

**This happens because “appl” + “e” has a + operation which puts both strings together whereas “apple” - “e” has a - operation which is an unsupported operand type in this case.**

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

**This happens because “Hello” \* 10 has a x operation which duplicates the string 10 times whereas “Hello” / 10 has a / operation which is an unsupported operand type in this case.**

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.

**"D" + "a" + "k" + "s" + "h" + "a"**

**'Daksha'**

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”.

**This happens because it uses a U+201D : RIGHT DOUBLE QUOTATION MARK {double comma quotation mark} which isn’t recognized by Python, when instead it should use a U+0022 : QUOTATION MARK.**

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

**It displays a SyntaxError once again due to the quotation marks used.**

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.

**This happens because “Hello!” only has 6 character whereas you are asking for a 7th character which is**

**nonexistent.**

**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?

**puppies / 3**

**12.0**

* 1. Why doesn’t typing kittens / 3 work?  
     **It doesn’t work because you’ve assigned values to puppies, not 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 are assigned a value of 36. Puppies are then assigned a value that is Puppies divided by 6, which is 36 divided by 6, making the new value of puppies 6.**

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 is assigned a string of red. Puppies are assigned an integer of 36. Colour + Puppies doesn’t compute since strings and integers can’t be added together.**

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

**Assuming Python uses BEDMASS, this can be explained by “Color + day \* fishes” multiplying fishes and day before adding Color whereas “( Color + day ) \* fishes” adds Color and day within the brackets before multiplying with fishes.**

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”?

**5**

* 1. Write an expression using mynumber to return ‘r’

**Dog = "Bark"**

**MyNumber = 3**

**Dog[MyNumber-1]**

**'r'**

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 to assign values, == is used to compare values.**

* 1. Create your own mnemonic to remember this difference.  
     **One equal sign assigns, two equal signs compare.**

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?

**Integers and Strings can’t be added.**

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

**int = Integer**

**str = String**

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?

**A Logic Error.**

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

**print("Faheem")**

**Faheem**

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”)

**String.**

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

**Boolean**

* 1. Why is the result different?

**The first command has quotation marks around True and the second one does not.**

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?

**It allows the programmer to implement decision making commands.**

**ex:("If this expression is True, do something; if the expression is False, do something else instead.")**

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?

**This is because then there wouldn’t be a clear command for the program to follow which will restrict it from operating efficiently, computers aren’t human.**

**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

**True and True**

**True**

**True and False**

**False**

**False and True**

**False**

**False and False**

**False**

* 1. Explain if there are any other combinations of True / False.

**No**

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

**AND is similar to = in how it checks if the comparison is true, but it is different in how it checks if only both comparisons are true, than the who expression is true as well.**

**ex( 3 == 3 and 1 == 1, 3 == 3 is True, 1 == 1 is True, therefore the whole expression is true,**

**3 == 4 and 1 == 1, 3 == 3 is False, 1 == 1 is True, therefore the whole expression is false,**

**3 == 4 and 1 == 2, 3 == 3 is False, 1 == 2 is False, therefore the whole expression is 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
      2. True or False
      3. False or True
      4. False or False

**True or True**

**True**

**True or False**

**True**

**False or True**

**True**

**False or False**

**False**

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

**OR is similar to AND where it checks if there is a comparison but it is different in the way how it needs two false commands to regard the whole thing as 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)
      2. not (True or False)
      3. not (False or True)
      4. not (False or False)

**not (True or True)**

**False**

**not (True or False)**

**False**

**not (False or True)**

**False**

**not (False or False)**

**True**

* 1. Explain how the combination of the NOT & OR operators is similar to the AND operator by itself and how it is different.

**NOT & OR operators compare 2 commands just like the AND operator does, yet it gives the opposite answer to AND.**

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

not (True or True): (True or True) = True, NOT gives the opposite answer making it False

not True or True: Not True is False, Not gives the opposite answer making it True

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

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 5 == 2.5 \* 2**

**True**

**True or 5 == 2.5 \* 2**

**True**

**not (True or 5 == 2.5)**

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

**favouriteteams = ["Toronto Raptors", "Los Angeles Lakers", "Cleveland Cavaliers"]**

**favouriteteams**

**['Toronto Raptors', 'Los Angeles Lakers', 'Cleveland Cavaliers']**

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?
      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

**pi = "3.14159265359"**

**if pi == "3.14159265359":**

**print("I like pie!")**

**I like pie!**

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 = "Faheem"**

**if myname == "Faheem":**

**print("Hi Faheem!")**

**else:**

**print("Hi Mr. Hanif!")**

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="Faheem"

myname1="Bob"

myname2="Miles"

myname3="Rohan"

if myname=="Faheem":

print("Hi Faheem xD")

elif myname1=="Bob":

print("Go away Bob.")

elif myname2=="Miles":

print("Ur Trash Miles.")

elif myname3=="Rohan":

print("Hello young one.")

else:

print("Good Afternoon.")