**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+6

=11

5-6

=-1

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.

6\*3

=18

6/3

=2

Answer\*

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

8/2

=4

* 1. And one division expression that gives a decimal number answer.

10/3

=3.3333333333333335

* 1. List your expressions and the results below.

(done above)

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(10/3)

=3

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

Typing == asks the computer is something equal to something. For example, typing “ answer==12” would be asking the computer if that statement is true or false.

* 1. What does “=” mean?

Typing a variable and value with a = sign in between gives that variable a value for any other times you’ll be typing that variable. For example if x=6 then, typing x\*2 would give you 12 if you used that variable again.

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\*4+6>=10/2+3

=True

2/2+1!=2\*3+0

=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.

Because if you want python to read a string, it has to be inside quotes.

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

Because math expressions can look like numbers but can also be strings if put inside quotes.

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.

Because we can only multiply how many times we show a string and add to put strings side by side, not subtract.

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

Multiplying two strings repeats how many times we see them. You cannot divide strings.

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.

G U R S I M R A T

0 1 2 3 4 5 6 7 8 9

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 index of l is not 4, it is 2 and 3. Indexes start at 0. H is zero, e is one, and so on.

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

That prints what seems like nothing, but is actually a space. The space is what character is at index 4. All the characters in a string are counted, even spaces

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.

There is no character at index 7. The indexes only go up to 5.

**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? 12
   2. Why doesn’t typing kittens / 3 work? Because Kittens has not been given a 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 = 36

Assigns the variable puppies to a value

* + - puppies = puppies / 6

Divides that value by 6

* + - puppies

Gives you the value of puppies

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”

Gives the variable colour the value of the string “red”

* + - puppies = 36

Gives the variable puppies the value of 36

* + - colour + puppies

Adds the two values 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

Entering the expression like this makes Python follow the order of operations to give a result

* + - ( Color + day ) \* fishes

Entering the expression like this tells Python to do the brackets first, which it wouldn’t do if they weren’t there

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”?

4

* 1. Write an expression using mynumber to return ‘r’  
     mynumber=5

fruit[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 for assignment statements

== is used for comparisons

* 1. Create your own mnemonic to remember this difference.

When we're assigning a value, we're saying "this equals that". That's a short sentence, so it only gets one equal sign: = But when we're comparing values, we're asking "is this thing equal to that thing?". And that's a longer sentence, so it gets two equal signs:

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?

Python cannot concatenate objects of different types

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

Int is an integer, which is a number, str means string or a character is other words

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.

Gursimrat Khaira

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

One is in quotations and one isn’t

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 is important because it is used in programming when we need to decide about what to do in 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 because either something works, or it does not. If there was a “maybe” in Boolean data, that would not help programmers make decisions about what to do in 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.

No there are not.

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

Only produces one answer, however in math the answer is an integer, and in the AND operator, the answer is either 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 like the AND operator and how it is different.

-Only produces an answer of either True or False

-If one thing is True, then the answer will be true unlike the AND operator where both must be true for the answer to be 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 like the AND operator by itself and how it is different.

All of them; answer will either be True or False

NOT operation- reverses the answer python gives, any operation that’s true can become false

AND- both must be true to get true as result

OR- if even only one is true, the response will be true

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)- False
      2. not True or True- True

for i), the response is false because it reversed the answer that it would give if in it brackets. For ii), it is true because it reversed first part of what is typed only.

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

Because not (true and true) is not (true) which is false. And not true and true is actually false and true which is also 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 like those in the tutorial.
   2. Provide the results for your practice expressions

True and 0==1

False

False or 0==0

True

True and “true”==”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. (I don’t watch any sports so I did my favorite food instead)
   2. Assign your list to a variable.
   3. Confirm that your variable and your list are the same.

Food= [“pizza”,”pasta”,”roti”]

Type(Food)

<class “list”>

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 for roti

* 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?

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 = "Alfred"

if myname == "Alfred":

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”).

me= "Gursimrat"

if me== "Gursimrat":

print("Hi Gursimrat!")

else:

print ("You are not Ms.Khaira")

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 = "Gursimrat"

if myname== "Gursimrat":

print ("Hi Gursimrat")

elif myname== "Amneet":

print ("Hi Amneet")

elif myname== ("Prabhnoor"):

print ("Go Away")

elif myname== ("Raman"):

print("Whats up")

else:

print ("Who are you")