**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.

12+3-2  
=13

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.

2\*2/3

1.3333333333333333

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.

9/3

3.0

10/3

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.

Round(10/4)

2

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

== means this thing is equal to that thing

= means this is equal to that

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\*2+10-2>5+3/2

true

2\*20-5+3<10-2\*2

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

If you want python to read a string it must be in quotes.

Since python reads it as a string it is not added.

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.

Appl+e works since we are adding 2 strings. Apple-e does not work because you cannot subtract strings.

Hello\*10 works since you can multiply strings but hello/10 does not work because 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.

Tejveer

0123456

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

It does not print “I” since the 4th index is “o”

It prints the space after the comma

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.

Since there are indexes only up to 5, so it cannot give any result since there is no seven.

**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?

The results you would get for puppies/3 would be 12

Typing kitten/3 doesn’t work since it does not have a set value

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  
        So first “puppies” is to set to have a value of 36 Then the value is set to puppies/6 which is 6, then puppies = 6.
2. Read through “Lesson 5: Variables – Rules”.
3. 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

Color is set to the value “red” and puppies is set to 36, and once you add them you get error, since you can’t add a string and a integer

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

Since whatever is in the bracket is done first the results are different.

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 is 4 in watermelon.

mynumber=6

fruit="watermelon"

fruit[mynumber-2]

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.

When we used one “=” we are saying this equals that. When we use two = then it means this thing is equal to that thing.

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?
   2. Wht is the difference between int and str?

Friend + 5 doesn’t work since you cannot add an integer and string.

An integer is a number and a string is text.

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?

Syntax error

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

print("First and last name”)

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”)
   2. What is the value of: type( True )
   3. Why is the result different?

The value of (“true”) is string. The value of (true) came as error. The results are different since it does not recognize the word true without quotations.

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?

Since it can help you choose between two different things.

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 what would the program do if It is maybe, there would be no option for the program.

**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 unless you use more than 2 true/false.

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

An AND operator needs both inputs true for the result to be true. It is similar to a math operator because it tells python to compare two sides. It is different because you don’t really input numbers, you input 1(true) or 0(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.

The OR operator needs at least on input to be true for the result to be true. It is similar to the AND operator since it somewhat uses a similar concept of true and refresh. It is different because it requires different inputs for the results to be true or false, it needs at least one input true for the result 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 similar to the AND operator by itself and how it is different.

Not and Or is similar to the And operator since they use similar concepts of true and false and they are different since they require different inputs for the results to be true or false. NOT and OR are similar to each other since they are the same operators but one inverts the input.

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 the first one python first solves the OR operator and gives true, then the NOT operator changes it to false. In the second one python takes the first true and changes it to false and then the expression becomes false or true, then the OR operator outputs true. The results are different because when you put something into brackets it is solved first, so for the first one true or true is solved which equals to true, and then the not operator changes it to false. For the second expression there is no brackets so it does the not operator first and then the equation left is false or true, then OR operator outputs true.

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

The first expression, solves the and operator first since it is in brackets then reverses the answer outputting to false. The second expression does the not operator first changing the first true to false, then is left with false and true, then the and operator outputs false. The results are the same since one expression has brackets and the other doesn’t and because the AND operator needs both inputs to be true for the output to be 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

“Puppies” == “Puppies”=True

“Puppy1” == “Puppy2”=false

1==1=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.
   2. Assign your list to a variable.
   3. Confirm that your variable and your list are the same.

sports = ["FC barcelona", "Chelsea", "Toronto Fc"]

print(sports)

['FC barcelona', 'Chelsea', 'Toronto Fc']

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?

The index is 2 for the last team in my list of favorite sports teams. It is an index error since there is no index of 3.

1. Complete “Lesson 8: Lists – Practice” and “Lesson 8: Lists – Practice Answers” by typing the sample commands in the black area of the IDE.

Color=[“Red”, “green”]

Color[0]

‘Red’

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

Alfred=10

if Alfred == 10:

print("Hi")

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”).

Hi = 10

if Hi == 10:

print("TEJ")

else:

print("singh")

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="Morgan"

if myname == "Jasjot":

print("Hi Jasjot!")

elif myname == "Raakin":

print("Hi Raakin!")

elif myname == "Tp":

print("Hi Tp!")

elif myname == "Jaskaran":

print("Hi Jaskaran!")

else:

print("Who are you?!?")