**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.

5+6

22-5

* 1. List your expression and the result below.

5+6=11

22-5=17

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.

5\*6

5/5

* 1. List your expression and the result below.

5\*6=30

5/5=1.0

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

Round (4/2)=2

* 1. And one division expression that gives a decimal number answer.
  2. 3/2=1.5
  3. List your expressions and the results below.

2

1.5

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.

Round (6/5)=1

Round (5/2)=2

* 1. List your “round()” expressions and the results they return below.

1

2

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

I think Equal is == instead of = because if you type in a number and use == it would give the answer and it would give it true.

* 1. What does “=” mean?

The = mean that is an error it would give you nothing it would say Syntax Error instead

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. **5<6**
3. **5 !=6**
4. **5<=6**
5. **guess==5**

5==6

* 1. And an expression using 5 different operators that returns a “False” result.
     + MyPet is “-“ “dog”
     + 5==6
     + 5>+6

1. **5!=5**
   * + 6==7
   1. 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.

Apple works with quotes because it tells python what to do when you add the quotes. The reason that it doesn’t work without quotes is that the python doesn’t know if you do not add quotes when you type in words.

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

I think “2+5” does not equal 7 because probably the quotes don’t work when you use number to add or subtract.

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.

Typing ‘appk” + “e” works maybe because it not a full word and it work when you do it. And when you type in the full word of apple and – to e it doesn’t work because your probably using the full word.

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

Because probably when you use the multiplication sign then it works when you use / 10 it probably doesn’t work.

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.

“R” + “a” + “j”

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”.

Print(“Hello!”[4]) doesn’t print “I” because python doesn’t know what it is trying to say.

* 1. What does print(“Hay, Bob!”[4]) print?

For a hint try print(“Hay, Bob!”[3]) and print(“Hay, Bob!”[5]) for print(“Hay, Bob!”[5]) it says error as well

Print(“Hay, Bob!’[4]) prints nothing. For print(“Hay, Bob!”[3]) it shows and error it doesn’t work.

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 it doesn’t match what the word said

**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?

last command try adding them, but, was missing a few characters

* 1. Why doesn’t typing kittens / 3 work?  
     it doesn’t work because it probably doesn’t have a brackets around

1. Complete “Lesson 5: Variables – Assign a New Value” by typing the sample commands in the black area of the IDE.

First you assign a variable to puppies, then you divide it by 6 to get the answer 6.0.

The first command multiplied day and fishes and then add color. Although the second command added color and day together because they were in brackets and then multiplied it by fishes.

* 1. Explain how the following sequence of commands works:
     + puppies = 36
     + puppies = puppies / 6
     + puppies

1. Read through “Lesson 5: Variables – Rules”.

The index of r in watermelon is 4.

The index of r in mynumber is 7.

1. 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 and puppies were both given a variable, but the last command try adding them, but, was missing a few characters

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  
The first command multiplied day and fishes and then add color. Although the second command added color and day together because they were in brackets and then multiplied it by 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”?

The index of r in watermelon is 4.

Write an expression using mynumber to return ‘r’  
The index of r in mynumber is 7.

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 “==”?

The sign = assigns a variable to something, while == compares the characters on both sides

Create your own mnemonic to remember this difference.  
Single

   Equal sign

   Assigns

   Variables

   Double

   Equal sign

   Compares

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?

It doesn’t work because friend is a string and 5 is an integer which are 2 different types of data. Python doesn’t know if were trying to add numbers or concatenate strings because it’s not that smart enough to make its own decisions so we need to clarify.

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

They are 2 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?

Its a Syntax Error

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

print(name)

Raj Shah

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”)

It’s a string

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

It’s a boolean

* 1. Why is the result different?

The result is different because the first line uses apostrophes which is used in strings, while the second is just a word which is a boolean.

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 very important in computer programming because they help make 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?

There is noMaybe data value in computer programming because that makes another possible outcome which gives you 3 different outcomes, but booleans are better because it helps computers choose between 2 different outcomes.

**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

* + 1. True and False

false

* + 1. False and True

False

* + 1. False and False

False

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

There are no other comparisons

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

It is similar because the and operator adds 2 variables like an addition sign.

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

* + 1. True or False

True

* + 1. False or True

True

* + 1. False or False

False

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

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

Its similar because if both sides are true, then the answer is true. The difference is if one side is true for AND comparisons then its false, but for OR comparisons if one side is true then its 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

* + 1. not (True or False)

false

* + 1. not (False or True)

false

* + 1. 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.

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 comparisons can tell its false if you type in not(False or True), AND comparisons will give you the same result if you type in False and 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)
      2. not True or True

The first statement has brackets so it deals with that first True or True equals, but not reverses the answer so you get False. The second statement does not have and brackets so not True gives us False and so False or True gives us True. The brackets is the reason why you get different answers.

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

The first statement has brackets so True and True equal True, but its reversed because of the not so you get False. The second statement has no brackets but not True equals False, then False and TRue equals 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

not(True or False and 1==2)

    =False

       3==3 and False

    =False

      not(False and True or False)

    =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", "Juventus FC", "Toronto Blue Jays"]  
   Sports  
=> ['FC Barcelona', 'Juventus FC', 'Toronto Blue Jays']**

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.

The list index of the last team is 2.

* 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.

**colour=["Blue", "Green", "Purple", "Gray"]  
   colour[0]  
=> 'Blue**

**colour[1]  
=> 'Green'**

**colour[2]  
=> 'Purple'**

**colour[3]  
=> 'Gray'**

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

Modify the tutorial code to print “Hi Alfred!” based on a decision using numbers

**A logic Error**

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”).

**yourName=input("What is your name?")**

**myName="Raj"**

**if(yourName==myName):**

**print("That's my name too!")**

**else:**

**print("Well its not a great name, but it will do.")**

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.

**if myname == " Raagaventhan":**

**print("Hi Raagaventhan!")**

**elif myname == "Raj":**

**print("Hi Raj!")**

**elif myname == "Raul":**

**print("Hi Raul!")**

**elif myname == "Karan":**

**print("Hi Karan!")**

**elif myname == "Andrew":**

**print("Hi Andrew!")**

**else:**

**print("Who are you?!?")**