Introduction to the Python Interactive Shell

And Variables

1. Activate your Python virtual environment
   1. What command did you use?
   2. source ~/py3/bin/activate
2. To enter the **Interactive Shell** Type python<ENTER>
3. Mathematical operations in Python follow the PEMDAS process you learned in math class. Parenthesis, exponents, multiplication, division, addition, subtraction. Style guides indicate you should place a space before and after operators.
   1. **Table 1-1:** Math Operators from Highest to Lowest Precedence

| **Operator** | **Operation** | **Example** | **Evaluates to . . .** |
| --- | --- | --- | --- |
| \*\* | Exponent | 2 \*\* 3 | 8 |
| % | Modulus/remainder | 22 % 8 | 6 |
| // | Integer division/floored quotient | 22 // 8 | 2 |
| / | Division | 22 / 8 | 2.75 |
| \* | Multiplication | 3 \* 5 | 15 |
| - | Subtraction | 5 - 2 | 3 |
| + | Addition | 2 + 2 | 4 |

* 1. Enter the following in the interactive shell and view the results
     1. 1 + 1<ENTER>
     2. 5 + 4 \* 3<ENTER>
     3. (5 + 4) \* 3<ENTER>
        1. Why is this expression different than the one above?
        2. It calculates 5+4 first.
     4. ((9 - 3) \* (3 + 2)) / (8 - 6)
        1. What is the answer to the above and how does Python arrive at that solution?
        2. 15, (6\*5)/2, 30/2, 15

1. Common Python datatypes include Integers, floating point numbers, and strings.
   1. Integers store negative and positive whole numbers
   2. Floating point numbers are used for decimal values
   3. Strings are used for character data and are enclosed in quotes.
   4. Python operators behave differently depending on the data type.
   5. Enter the following in your Interactive shell.
      1. 2 + 2<ENTER>
      2. 2 + '2'<ENTER>
         1. Explain the results.
         2. TypeError: unsupported operand type(s) for +: 'int' and 'str'
         3. Can’t plus str and int together
      3. 3 \* 2<ENTER>
      4. 3 \* '2'<ENTER>
         1. Explain the results.
         2. TypeError: unsupported operand type(s) for +: 'int' and 'str'
         3. Str can’t \* int variables
      5. 'Brown' + 'Dog'<ENTER>
         1. Explain the results.
         2. ‘BrownDog’
         3. Making 2 str together is possible
      6. 'Brown' \* 5
         1. Explain the results.
         2. It writes the word ‘Brown’ five times.
      7. 'Brown' \* 5.0
         1. Explain the results.
         2. can't multiply sequence by non-int of type 'float'
         3. Can’t use float type on str type
2. Variables are containers for storing values. They are given values through **assignment statements** using an **assignment operator** which is the equals sign.
   1. Python is **CASE SENSITIVE,** so spam, Spam, and SPAM are all different variables.
   2. Style guides for Python indicate you should place a space before and after your assignment operator (equal sign).
   3. Python variable names *cannot* contain hyphens, spaces, or special characters. Underscores are acceptable.
   4. Python variables are *not* preceded by an assignment character like dollar sign in other languages.
   5. Python variables cannot begin with a number, but a number can be used anywhere after the first character.
   6. Style guides indicate variables should start with a lower case letter.
3. Enter the following into your interactive shell:
   1. spam = 40<ENTER>
   2. Spam = 20<ENTER>
   3. SPAM = 10<ENTER>
      1. What value will you get if you type Spam<ENTER>?
      2. 20
      3. What value will get if you type sPam<ENTER>?
      4. NameError: name 'sPam' is not defined
   4. eggs = 2<ENTER>
   5. spam + eggs<ENTER>
      1. Explain the results.
      2. 42, 2 int type plus together 40+2
   6. SPAM\*eggs<ENTER>
      1. Explain the results.
      2. 20, 10\*2 all int type variables
   7. spam = spam + 2<ENTER>
   8. spam<Enter>
      1. Explain the results.
      2. 42, as the command put ‘spam’ equal to ‘spam’ itself plus 2
4. None is the Python equivalent of Null. Enter the following into your interactive shell.
   1. spam = 'None'
   2. spam
      1. Explain the results.
      2. It equals the letters None
   3. spam = None
   4. spam
      1. Explain the results.
      2. None made spam into a ‘NoneType’
5. To exit the interactive shell type:
   1. exit()<ENTER>