RUNNING A SCRIPT VS THE INTERACTIVE INTERPRETER

- python [script_name.py] [args]
  - save to any directory
  - argparse module
- python
  - primary prompt
  - interactive
  - >>>

IF STATEMENTS:

- Comparison Operators
    - ==, !=, <>, >, <, >=, <=
- Logical Operators
    - and, or, not

ELIF & ELSE

- elif (else if)
- else (default or last case)
- same syntax as if statement

OPERATORS

- Membership operators
    - in
    - not in
- Identity operators
    - is
    - is not

SIMPLE FOR LOOP

- for x in [iterable_object]
    - Strings, lists, etc
    - Ex. Iterate through a list

FOR LOOP

- for x in range():
  - xrange()
  - range(start, stop, [step])

WHILE LOOP

- while 1:
- while f.readline():
- while count < 10:

CONTINUE, BREAK & PASS

- continue to next iteration of loop
- break out of inner most
- pass, does nothing
  - used when something is required but no action is required
    - ex try, except block

LOOP ELSE

- used when iterating through object
- when you get to the end of the object without breaking out

FUNCTIONS

- def func():
- indent

ARGUMENTS

- Required, Keyword, Default, Variable-length
  - required
    - def func(string):
  - keyword
    - def func(string):
      - func(string = “String”)

ARGUMENTS

- default
  - def func(string = “Default”):
    - like keyword can be out of order
- variable-length
  - def func(*var_len):
    - for item in var_len:
      - print item

ARGUMENTS THAT ARE CHANGED WITHIN THE FUNCTION

- Change item in list example

RETURN

- can return an expression
  - ex. return a + b
- Can return multiple variables (via a tuple)

LAMBDA (ANONYMOUS FUNCTION)

- lambda [arg1 [, arg2,…..argn]]:expression
- sum = lambda arg1, arg2: arg1 + arg2
- sum(10, 10)

EXTRA OPERATORS

- ** Exponent
- // Floor Division
- +=, -=, /=, *=, %=, **=, //=,