**Python 3**

**Summary:**

* General purpose versatile popular programming language. Concise and easy to read, also good language to have in programmer stack as can be used for everything from web dev to software dev and data science

**SYNTAX**

* Comments
  + Texxt written in prog, but not run by cpu called comment.
  + Python interprets anything after a # as a comment.
  + Provides context for why something is written way it is
* Print
  + print( “whatever u want in here” )
  + print can print anything, no need to convert to string
* Strings
  + Blocks of text.
  + Can be surrounded with double or single quotes, **JUST BE CONSISTANT**
  + **Multi-Line Strings**
    - If want to write string spanning multiple lines use ‘’’ or “””
    - This lets compiler know string doesn’t end till next ‘’’.
* Variables
  + Way to store data for reuse. Assign variables with = sign in python.
    - Example
      * message\_string=”Hello There”
      * print(message\_string)
  + After var assigned to initial val, it CAN be reassigned to new val with dif data type from initial.
* Errors
  + When error occurs, python will point it out with a ^ character.
  + When prog throws error we don’t expect to encounter, call those BUGS.
  + TWO COMMON ERRORS
    - SyntaxError:
      * Something wrong with way your prog written
        + Punctuation does not belong
        + Command where not expected
        + Missing parenthesis
    - NameError
      * Python Interpreter sees word does not recognize.
      * Code that contains something that looks like var but was never defined will throw error
* Numbers
  + Python has few numeric data types.
  + Int
    - Whole number, no decimal point and contains counting numbers as well as negative and the number 0
    - An\_int=2
  + Floating-Point number (float)
    - Decimal number. Can be used to rep fractional quantities.
      * Ex: average test score, length of wall…
      * A\_float=2.1
* Calculations
  + Performs standard +, - , \* , /
  + When doing devision, result has a decimal place. This is because Pyton converts all ints to float before performing division. (2.8 and up) version
  + DIVISION CAN THROW ZERODIVISIONERROR
  + EXPONENTS
    - Use the notation \*\*
    - Example: print(2 \*\* 10) # 2 to the 10th power
  + Modulo
    - %
    - Give remainder of division calc
    - If number divisible, then modulo operator will be 0
      * Ex: 29%5=4
      * 25%5=0
  + CONCATENATION (+)
    - Doesn’t just add numbers, CAN ADD STRINGS
    - Concatenating strings can create brand new string compromised of contents of first and second string…
    - IF WANT TO CONCATENATE A STRING WITH A NUM HAVE TO MAKE NUM A STRING FIRST USING str()
    - Ex:
      * Age=10;
      * Print(“I am” + str(age)+” years old”)
  + PLUS EQUALS
    - Shorthand for updating vars.
    - When have num saved in var and want to add curr value of var can use +=
    - Example:
      * Num\_hiked=12
      * numHiked+=2;
    - CAN BE UESD FOR STRING CONCATENATION TOO