Week 2

1. Create a new document in Wing and enter the following python comman on the right in the upper window.
2. You should see that same statement print to the lower right console once you lick the “run button”
3. Assign a number to a variable using the “=”. Here, you’re storing a number, which is one type of data.
4. Create three more variables:
   i. Integer (no decimal)
   ii. Floating point (decimal)
   iii. boolean (true or false)
5. Now, declare a variable that is an integer
   i. print that variable
   ii. change the value
   iii. print it again
6. Now we’ll revisit functions. Functions require the following:
   i. a “def” or define statement
   ii. function name
   iii. parenthesis with input inside (). A function doesn’t require input...
   iv. return statement
7. You’ll notice the “#” in step 6, that allows you to comment what you’re doing.

```
3  print "Oh Hi!, This is a Python print statement"
```

```
4
5  var = 123
```

```
4  var_int = 45
5  var_float = 1.23
6  var_bool = True
```

```
3  var = 123
4  print var
5  var = 456
6  print var
```

```
2.7.8 (defaul
Python Type "
[evaluate unt
123
456
>>>
```

```
2  #define/create function
3  def func():
4      #create a variable
5      var = 10
6      #return variable as output from function
7      return var
8
9  #call function and print the result
10 print func()
```

Math Operations

1. Create the following variables:
   i. add (add two numbers)
   ii. subtract (subract two numbers
   iii. multi (multiply)
   iv. divide (divide)
8. Exponents
   i. use the ** to raise one number to the power of another
9. Modulo
   i. Use “%” to ger the remainder from a division and print the result
10. Use some Math to figure out what a bar bill would be in total using the following:
    i. bar tab is 15.75
    ii. local sales tax is 6.75%
    iii. you tip generously at 18%

```
2  add = 123 + 456
3  subract =  456 - 123
4  multi = 123 * 456
5  divide =  456 / 123
```

```
7  exp = 123 ** 456
```

```
9  mod = 5 % 3
10 print mod
```

```
2  barTab = 15.75
3  tax = 0.0675
4  tip = 0.18
5
6  #put your equation here
7  barTotal =
8
9  print barTotal
```

```
+---+---+---+---+
| T | E | X | T |
+---+---+---+---+
  0   1   2   3
```

11. Create a string variable set to some text
12. print the string statement
13. Strings can be contained in single ‘ ‘ or double “ “ quotes. But, if a an apostrophe appears in a line Python will take issue with it. You can tell Python to treat it correctly using a “\”.
14. Access parts of a string using the index and bracket notation.
    i. set a string equal to variable
    ii. access the 3rd letter (index 2)
15. Use the “len()” function to determine the length of a string
    i. set a string equal to a variable
    ii. print the length
16. use the “.lower()” and “.upper()” to convert a string to all lower and all upper
17. Convert a number to a string using the “str()” method
18. Concatenation of strings and formatting
    i. You can combine strings using the “+” operator
    ii. convert a number to a string and add it to another string
    iii. use the “%s” format to fill in blanks
19. Use the “raw_input” to store user input as a variable and then add it to a string

```
2  text = 'The Dude abides'
```

```
2  text = 'That\'s the Dude\'s rug!'
3  print text
```

```
1  text = 'The Dude Abides'
2  print text[2]
```

```
3  text = 'The Dude Abides'
4  print len(text)
5
```

```
3  text = 'The Dude Abides'
4  print text.lower()
5  print text.upper()
```

```
3  n = 132
4  text = str(n)
5  print text
```

```
3  text =  'the' + 'Dude' + 'abides'
4  answer = 'The answer to the univers is: ' + str(42)
5  mod = 'That %s really tied the %s together Dude' %('rug','room')
6
7  print text
8  print answer
9  print mod
```

```
2  name = raw_input("What's your name? ")
3
4  print "Hi %s. I'm HAL" %(name)
```

Comparisons

20. create some boolean variable using the different comparison operators and print them
    i. use equal to: ==
    ii. use not equal: !=
    iii. use less than: <
    iv. use greater than: >
    v. use greater than or equal to: >=
21. As an exercise, make b1 and b3 True. Make b4 and b5 false using comparison operators.
22. We’ll use the “and” and “or” operators to see how they can be used

```
3  b1 = 25 == 5**5
4  b2 = 4 != 5
5  b3 = 72 < 43
6  b4 = 72 > 43
7  b5 = 25 >= 25
8  print b1, b2, b3, b4, b5
```

```
b1 = 1>2 and 2>3
b2 = True and False
b3 = False or True
b4 = True and True
b5 = 2==2 and 3==3
print b1, b2, b3, b4, b5
```

23. Now, try the “or” operator and compare how that works with the “and” operator

```
3  b1 = 1>2 or 2>3
4  b2 = True or False
5  b3 = False or True
6  b4 = True or True
7  b5 = 2==2 or 4==3
8  print b1, b2, b3, b4, b5
```

Comparisons Continued

24. Using the "not" statement. Not, will return True for false statements, and False for True statements.
25. Work through the variables below to make them correct.

```
# Make me false!
b1 = 1 > 2 and 3==3

# Make me true!
b2 =  1 > 0 and not False

# Make me false!
b3 = not True or 2 == 3

# Make me true!
b4 =  not False or 3*4 ==12

# Make me true!
b5 = 1 != 2 and 1 == 1
```

```
b1 = not 2>3
b2 = not True
b3 = not False
b4 = not 3>2
b5 = not 2==3
print b1, b2, b3, b4, b5
```

Conditional Statements

26. We're going to go back to creating a function. Inside of that we'll use an "if" statement to return something. In the code to the right replace the "condition" with a comparison that will be evaluated as True.

```
def cont1():
    if condition:
        return "Great Success!"
def cont2():
    if condition:
        return "Great Success! Very Nice!"
print cont1()
print cont2()
```

27. Ammend the cont1 function so that the if condition evalutes if a text variable is equal to "The Dude Abides". If the condition is true, return the string "Far out Man". If not, return the string "Fine! You human paraquat!"

```
def cont1():
    if condition:
        return "Far out man"
    else:
        return "You human paraquat!"

print cont1()
```

28. Ammend cont1 function so that it takes and integer as input and evalutes if it is less than or greather 5 or zero

```
def cont1(number):
    if condition:
        return 'greater than 5'
    elif condition:
        return 'less than 5'
    else:
        return 'number is zero'

print cont1(6)
```

29. Nowe we'll put all this together to create a pig late translator!

Pig Latin Translator

30. Start with a print statment welcoming the user to pig Latin translator
31. create a variable that asks for input from the user
    i. check that the word input by the user has a length greater than 1
    ii. if it doesn't, print a statment alerting the user
32. also, check that the word is a word (not just using length) you can use word.isalpha() to check. If word was equal to 10, this would return false. Add this to the conditional statement using an "and"
33. After getting the user input, but before entering the "if" statment, convert the word to all lower case
34. access the first character of the user input word and set it equal to a variable using the bracket notation
35. create a 2nd variable that is everything but the first letter of the word. You can do this using the brackets, and colon ":" notation.
36. For Pig Latin, you convert the word differently if the first letter is a vowel or consonant.
    i. if the first letter is a vowel the conversion is the word + 'ay'
    ii. if the the first letter is a consonant you have the word minues the first letter + the first letter + 'ay'
37. This should get you close, see if you can complete the Translator.

```
2  print "Your statement here"
3  word = raw_input('some text here asking for input')
4  if condition:
5      print word + ' is a real word'
6  else:
7      print 'sorry, not a word'
```

```
5  first_letter = word[0]
```

```
5  first_letter = word[0]
6  other_letters = word[1:]
```

```
8  if words is greater than zero and a word:
9      if first_letter is a vowel:
10         pig_word =
11         print pig_word
12     else:
13         pig_word =
14         print pig_word
15 else:
16     print 'Try again, you didn\'t type a real word'
```