Quick Guide to Python Commands and Syntax

Input Statements:

```
# Prompt for Decimal Number
x = float(input('Enter the value for variable, x '))

# Prompt for Integer
y = int(input('Enter the integer value for variable, y '))

# Prompt for String
month = input('Enter the month you were born ')

# Prompt for List of Numerical Values
List = eval(input('Enter list values in square brackets separated by commas '))
```

Output Statements:

```
# Decimal Numbers
x = 75.176
print('The variable x = {0:.2f} \n'.format(x))
```

```
The variable x = 75.18
```

```
# Integers
y = 750
print('The value of y is: {0} \n'.format(y))
```

```
The value of y is:  750
```

```
# Strings
course = 'Calculus II'
print('The name of this course is: {0} \n'.format(course))
```

```
The name of this course is:  Calculus II
```

```
# Mixture
print('x = {0:0.1f}, y = {1}, course is: {2} \n'.format(x,y,course))
```

```
x = 75.2, y = 750, course is Calculus II
```

Arithmetic, Comparison, and Logical Operators:

Arithmetic Operators		Comparison Operators		Logical Operators
Addition	+	Equal	==	and
Subtraction	–	Not Equal	!=	or
Multiplication	*	Less Than	<	not
Division	/	Greater Than	>	
Power	**	Less Than or Equal To	<=	
Modulo	%	Greater Than or Equal To	>=	

Conditional Statements:

```
if Condition1:
    # Python Commands to execute if Condition1 is TRUE
elif Condition2:
    # Python Commands to execute if Condition1 is FALSE
    # and Condition2 is TRUE
elif Condition3:
    # Python Commands to execute if Condition1 is FALSE
    # and Condition2 is FALSE
    # and Condition3 is TRUE
else:
    # Python Commands to execute if Conditions 1-3 are FALSE
```

For Loops:

```
Assume variable N is defined (an integer)
for k in range(N):
    # Python Commands to execute a total of N times
    # Counter index, k, starts at 0 and increments to N-1

Assume List is a list of numerical values
for k in List:
    # Counter index, k, takes on each value in List starting
    # with the first and ending with the last entry in List
```

While Loops:

```
while Condition:
    # Python Commands will execute again and again as long as
    # Condition is true
```

Functions:

Basic Syntax:

*Rectangle.py - C:\Users\Kathy Ossman\Desktop...

File Edit Format Run Options Window Help

```
def Rectangle(Width,Height):
    Area = Width*Height
    Perimeter = 2*Width + 2*Height
    return Area, Perimeter
```

Ln: 7 Col: 0

Calling a Function or Functions from another module:
Create another module called Main.py (see below) that imports any necessary functions then calls the function(s). Run Main.py and respond to prompts.

Main.py - C:/Users/Kathy Ossman/Desktop/Main.py (3.7.0)

File Edit Format Run Options Window Help

```
# Main Function that calls Rectangle Function

Width = float(input('Enter Width: '))
Height = float(input('Enter Height: '))

import Rectangle #Filename Here

[Area,Perimeter] = Rectangle.Rectangle(Width,Height) #FileName.FunctionName

print('Area = {0:.2f}, Perimeter = {1:.2f} \n'.format(Area,Perimeter))
```

Ln: 14 Col: 0

Lists:

In Python, the first entry in a list or array is indexed as 0. This is different from MATLAB where the first entry in an array is indexed as 1!

Creating 1-d Lists and 2-d Lists:

```
List_1d = [2, 7, 6, 42, 73]

List_2d = [[1, 5, 7], [2, 4, 6], [10, 14, 18]]
```

Suppose L1 is a 1-d List of numbers with at least 5 values

```
L1[0]           # 1st entry in the list, L1

L1[3]           # 4th entry in the list, L1

L1[1:4]         # Pulls out 2nd 3rd, and 4th entries from List, L1

L1[2] = 5       # Replaces the 3rd entry of list, L1, with a 5

N = len(L1)     # N = the number of entries in list, L1

L1.append(12)   # Puts a 12 at the end of the list, L1

del L1[1]       # Deletes 2nd entry in the list, L1
```

Suppose L2 is a 2-d List of numbers with 5 rows and 3 columns

```
L2[1][2]        # Entry in row 2, column 3 of list, L2
```

<u>Reading and Writing to Files</u>:

Open a file to read (r), or write (w), or read and write (w+):

```
fid = open('FileName.txt','r')

fid = open('FileName.txt','w')

fid = open('FileName.txt','w+')
```

Read from a file :

```
List = fid.readlines()

     OR

for line in fid:
```

Write to a file:

```
fid.write(<string to write>)  #Can only write strings!
```

Close the file:

```
fid.close()
```