1. **Lists** are written with **square** brackets.
2. **Tuples** are written with **round** brackets.
3. **Sets** are written with **curly** brackets.

**Note**: Lists, Tuples and Sets uses the double **round-brackets for Constructor**.

In **Dictionary** keywords are not string literals. use of equals rather than

Colon for the assignment. eg : brand="Ford"

Dictionary is a collection which is unordered, changeable and indexed.

Dictionaries are written with curly brackets, and they have keys and values.

**help()**

**quit**

You can find the address of the variable by using:

num = 5

id (num)

184654879 #this is the address

**Boolean:**

True - 1

False - 0

**Data types:**

* None (null)
* numeric
* List
* Tuple
* Set
* String
* Range
* Dictionary

The **elif** keyword is python’s way of saying "if the previous conditions were not

true, then try this condition".

**Python while Loops:**

With the **while loop** we can execute a set of statements as long as a condition is true.

With the **break** statement we can stop the loop even if the while condition is true.

With the **continue** statement we can stop the current iteration, and continue with the next:

**The range() Function**

To loop through a set of code a specified number of times, we can use the range() function,

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

The range() function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter: range(2, 6), which means values from 2 to 6 (but not including 6).

The range() function defaults to increment the sequence by 1, however it is possible to specify the increment value by adding a third parameter: range(2, 30, **3**).

The else keyword in a for loop specifies a block of code to be executed when the loop is finished.

Nested Loops

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop"

Decimal (base 10) **->** 0-9

Binary (base 2) **->** 0-1

Octal (base 8) **->** 0-7

HexaDecimal (base 16) **->** 0-9 a-f

# **Python Lambda**

Syntax

lambda *arguments*: *expression*

**e.g.:**

x = lambda a: a + 10  
print(x (5))

Output = 15

x = lambda a, b: a \* b  
print(x (5, 6))

Output = 30

**Import Math Funtions**

Type: **import math or import math as m**

x = math.sqrt(25) #To find square root

Output: 0.5

**Floor**: Even if value is 2.9 you will get 2.

**Ceil**: Even if value is 2.1 you will get 3.

E.g.: print(math.floor(2.9))

2

print(math.ceil(2.1))

3

If you want specific functions to be imported from math type:

**from math import sqrt, pow**

(then you don’t have to specific math in command) e.g. pow(4,5)

argv – Argument values(Multiple values)

* If you give input from cmd then :

Import from sys

import sys

x= int(sys.argv[1])

y= int(sys.argv[2])

z=x+y

print(z)

Now, in cmd:

python Mycode.py 5 10