**PYTHON**

PYTHON developed by GUIDO VAN RUSSOM in 1989 while he worked at national research institute I Netherlands. And official date of launched in Feb-20th- 1991.

Python name is came from a popular fun show in BBC called Monty Python Flying Circus show (1969 to 1974)

Python borrows some features some languages ,they are

1. Functional programming from C
2. OOPS from C++
3. Scripting languages from Perl and Shell script
4. Modular programming features from Modula-3
5. Most of the SYNTAX from C and ABC language

**WHY WE NEED TO LEARN PYHTON**

1. Python is a simple and easy programming language.
2. It is a general purpose programming language means we can use in AI, MACHINE LEARNING, ROBOTICS, WEB DEVELOPMENT, IOT devices etc.,
3. High level language means human understandable language.
4. We can write code easily with less code

PYTHON is a *dynamically typed programming language* means we are not required to declare the datatype of variable, python automatically declares datatype while entering data is know as dynamically typed

**WHERE WE CAN USE PYTHON**

1. In Desktop Applications (Standalone applications)
2. To develop Web Applications ---> Django, Flask etc,
3. In Database Applications
4. In Data Analysis
5. Data Science
6. Games
7. Machine Learning Applications
8. Artificial Applications ( AI )
9. For IOT applications

**WHICH COMPANIES USE PYTHON ?**

1. Google
2. YouTube
3. NSE
4. NASA
5. Dropbox etc.,

**FEATURES OF PYTHON:**

1. Simple and easy to learn
2. It is Open Source and Free ware
3. High Level Programming Language
4. **Platform Independent** ------> write code once and run anywhere
5. **Portability**  ------> moving python program from one machine to different machines without any change anything in code (ex: our sim can port to idea to Jio with the same number)
6. **Dynamically Typed Programming Language** -----> we are not required to declare type of the variables
7. Both POP (Procedure Oriented Programming) and OOP (Object Oriented Programming)
8. **Interpreted Language** ------> do not need any compilation, internally it compiles and if any error occurs then it will arise syntax error. It executes line by line
9. **Extensible** ------> we can use other language code in python. We can use some other language functionality in python
10. Embedded ------> we can use python code in other languages also.
11. Extensive library ------> we prefer to buy rice from market rather than to produce in fields

**LIMITATIONS OF PYTHON**

1. Not up to in Performance
2. Not in all mobile applications

**IDENTIFIES IN PYTHON**

1. Any name in python by default ad identifies names may be variable names, methos names, class names

**Rules for identifies:**

1. Allowed symbols are a to z, A to Z and 0 to 9, \_
2. Identifies should not starts with digits
3. It is a Case Sensitive
4. We could not use Reserved words/Keywords
5. No length limit

If any identifier starts with

* \_ name ===> then it is private.
* \_ \_ name ===> if strongly private
* \_ \_ name \_ \_ ====> language defined special name

**RESERVED WORDS /KEYWORDS**

The words that are reserved to represent some type functionality or meaning for the word. Ex: VIP chairs in front row in meetings for a particular person

In python there are 33 reserved words are there:

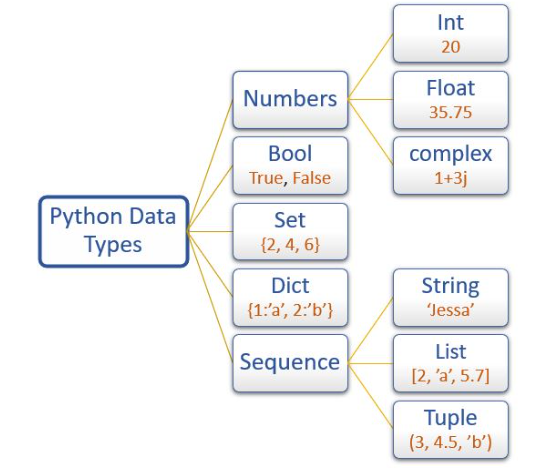
* True, False, None
* and, or, not, is
* if, elif, else
* while, break, continue, pass, yield, return, for, in
* try, except, finally, raise, assert
* import, from, as, class, def, global, nonlocal, lambda, del, with

In IDLE we can see all keywords by typing below code

import keyword

print(keywords.kwlist)

**DATATYPES IN PYTHON**

1. int (integers) ----->4, 2, 210, 521, 52, whole numbers
2. float -----> 5.5, 5.655, 5879.2
3. complex ------> 10+2i, 5+35j
4. Bool -------> True or false statements
5. Strings -------> characters
6. List -------> list of values ()
7. Tuple ------->list of values []
8. Set ------->list of values without duplicates {}
9. Dictionary ------> key value pair
10. Bytes -------> group of bytes (0101), immutable
11. Bytearray -------->similar to byte, mutable
12. Range -------> range of value
13. Frozen set --------> same as set but it is immutable we can’t modify
14. None -------->

**Python provides some inbuilt functions**

1. print()
2. type()
3. id() used to know the address of object

**DETAILED EXPLANATION**

**1 ) int**

Number without decimal value.

We can represent int values in different ways

* decimal form base-10 ex :- a=9656 by default python all numbers as decimal form
* Binary form base -2 ex:- 0, 1 a=0b01011 in this format we can represent binary format
* Octal form base-8 ex:- 0 to7 a=0o1241
* Hexa decimal base-16 ex:- 0 to 9 and a to f , A to F a=0xFace74

Here we can take b, o, x in small and bigger case

Output is only in Decimal form

**2)float**

We can represent float values in binary form and one extra feature is we can represent in exponential form

Ex:- f=1.5e3

**3)complex**

**Ex:-** a+bj a= real part , b= imaginary part we use only j other than j there will be error

In real part we can use any form (binary, decimal, hexa, octa) but in imaginary part we can use only decimal and float.

**4)bool**

True or False statements

**5)str**

String means any sequence of character enclosed in “ ”, ‘ ‘ for single line. and for multiline “”” “””, ‘’’ ‘’’

Ex:- a=”example”

Print(a)

b=”””welcome to

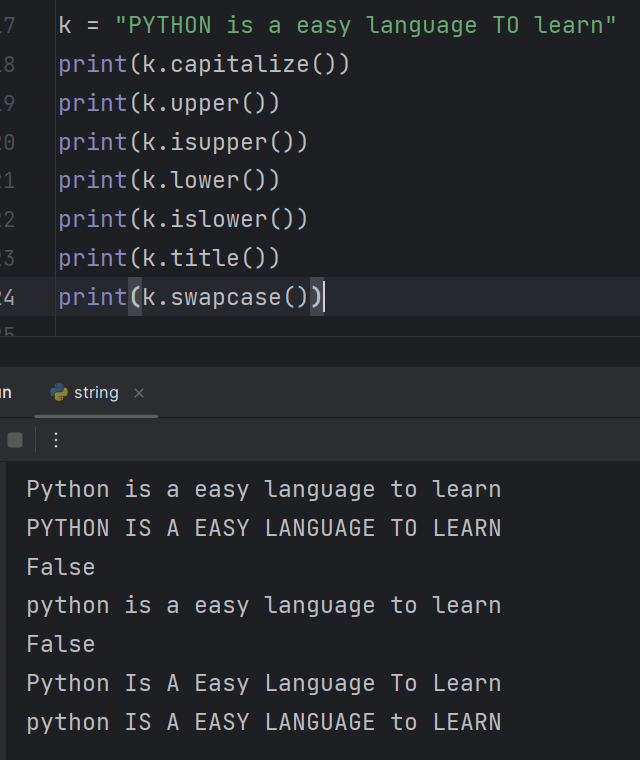
tweak talent”””

print(b)

**Built in** methods for strings are

capitalize, center, count, endswith, startswith, find, index, rfind, rindex, isalnum, isalpha, isdigit, isspace, islower, istitle, ljust, rjust, lower, upper, strip, lstrip, rstrip, min, max, replace, split, swapcase, title, zfill

capitalize:- only the first letter is uppercase and remaining total string is in lowercase

Upper :- it changes the complete string into uppercase

lower :- it changes the complete string into lowercase

Swapcase:-it changes as lowercase to uppercase and uppercase to lowercase

Examples for all methods:

text = "Hello, World 123"

isupper() - Check if all characters are uppercase

print(text.isupper()) # False

islower() - Check if all characters are lowercase

print(text.islower()) # False

swapcase() - Swap the case of all characters

print(text.swapcase()) # hELLO, wORLD 123

lower() - Convert all characters to lowercase

print(text.lower()) # hello, world 123

isspace() - Check if all characters are whitespace

print(text.isspace()) # False

istitle() - Check if the string is titlecased

print(text.istitle()) # True (because it starts with a capital letter)

isnumeric() - Check if all characters are numeric

print(text.isnumeric()) # False

isalpha() - Check if all characters are alphabetic

print(text.isalpha()) # False (because of the comma and space)

isdigit() - Check if all characters are digits

print(text.isdigit()) # False (because of the letters and comma)

center() - Center the text within a specified width

centered\_text = text.center(20, "\*")

print(centered\_text) # \*\*Hello, World 123\*\*\*

capitalize() - Capitalize the first character of the string

print(text.capitalize()) # Hello, world 123

endswith() - Check if the string ends with a specified substring

print(text.endswith("123")) # True

startswith() - Check if the string starts with a specified substring

print(text.startswith("Hello")) # True

min() - Find the minimum character (based on ASCII value)

print(min(text)) # ' ', the space character

max() - Find the maximum character (based on ASCII value)

print(max(text)) # 'r', the 'r' in "World"

Strip: it is used to remove spaces at first and last of the string

Syntax:- s=” python language is easy to learn”

S1=s.strip()

lstrip: this is used to remove spaces at left side/beginning

syntax:- S1=s.lstrip()

rstrip: this is used to remove spaces at right side /end

syntax:- S1=s.rstrip()

find: is used to find the index of a particular character or word, if the specified substring is not there then the output gives -1

rfind: is used to find the index of word/character in reverse direction but gives a positive index only. If the word is not there then gives -1 as output.

Syntax :- s=”python language is easy to learn”

s.find(“substring”, start, end)

s.rfind(“substring”, start, end)

index: same as find but if substring is not available then the output gives value Error

rindex: same as rfind output is like index method

syntax: s.index(“substring”, start, end)

s.rindex(“substring”, start, end)

count: is used to count a particular character or a substring is available in main string

syntax:- s.count(“substring”, start, end)

replace:- is used to replace a particular character or a substring

syntax: s.replace(“old substring”, ”new string”)

though the string is immutable we are to change the string because replace method always create new object

split :- is used to split w.r.t some character

syntax: l=s.split(“separator”, maxsplit)

maxsplit is for how many splits do you want

split is related to strings method and sep operator is related to print function

rsplit: is used to split from right side of the string

syntax: l=s.rsplit(“separator”, maxsplit)

join :- it is used to join the string with a separator

syntax:l1=” ”.join(l)

Zfill :- The **zfill()** method in Python is used to pad a numeric string with zeros (0) on the left side to a specified width.

# Original numeric string

numeric\_string = "42"

# Pad with zeros to make it 5 characters wide

padded\_string = numeric\_string.zfill(5)

print(padded\_string) # Output: "00042"

try reversed of a string, with different types

These 5 datatypes are known as fundamental datatypes and these are immutable.

In python object reusability is possible

Ex: - x=10, y=10, z=10

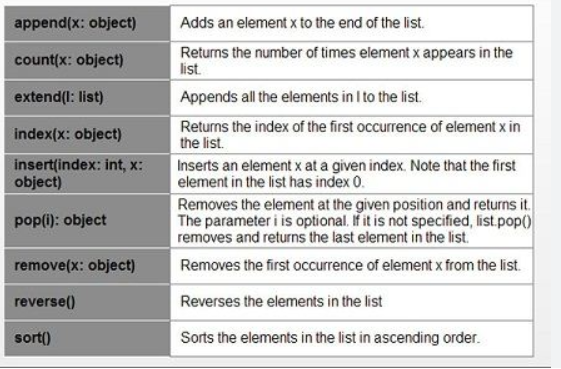
Here 1 object is created with 10 and x, y, z are pointed to one abject only, to check that print id of x, y

**6)List**

* list elements can be represented in [ ]
* List is mutable
* Duplicates are allowed
* Allows both homogeneous and heterogeneous datatypes
* Slicing and indexing are possible

Some of the basic operations for List are

+, -, len, min, max, sum, membership, iterations, repetition

Built in methods for List are:

**7)Tuple**

* It performs Basic operations
* Built in method for Tuple are

Count, index

**8) Set**

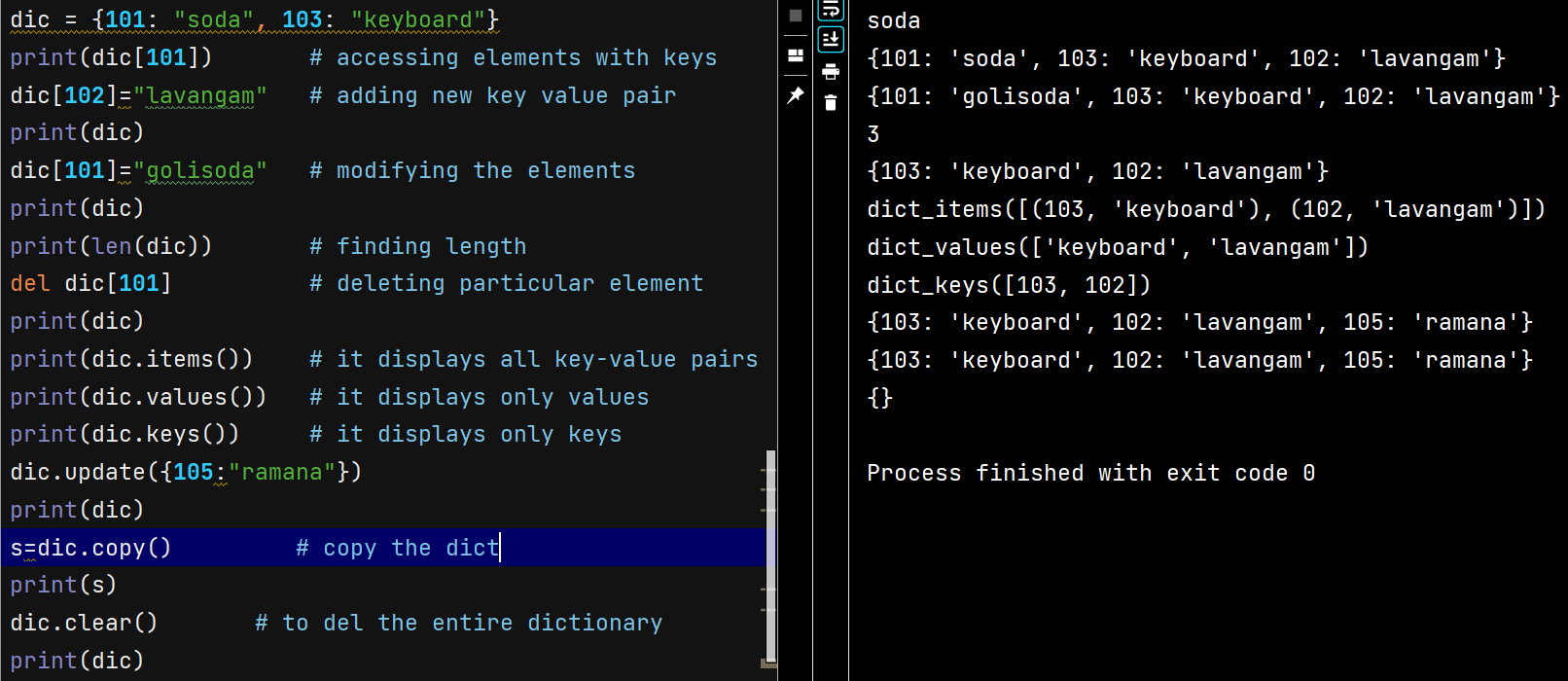
**Built in methods** for set are

add, remove, discard, pop, clear, len, membership, issubset, issuperset, union, intersection, difference, copy, symmetric\_difference, update, intersection\_update, difference\_update, symmetric\_difference\_update

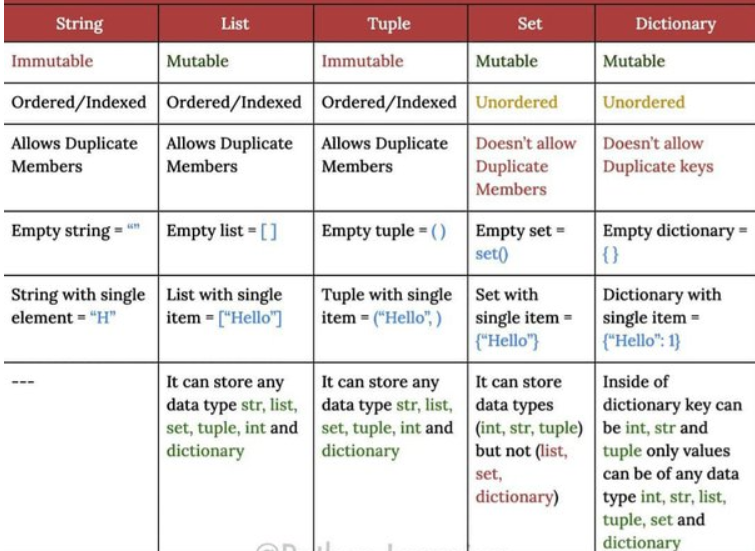
1. **Dictionary**

**Built in methods**  for dictionary are

Clear, copy, items, values, keys, update

****

**Collection :** A group of objects stored as a single entity.



* **Difference between String, list, Tuple, Set, Dictionary**
* ****These are the list built in methods methods

**ESCAPE CHARACTERS**

**\n**  for new line

**\t**  for tab space

**\r** cursor will go to first position

**\b**  back space mans it removes the space

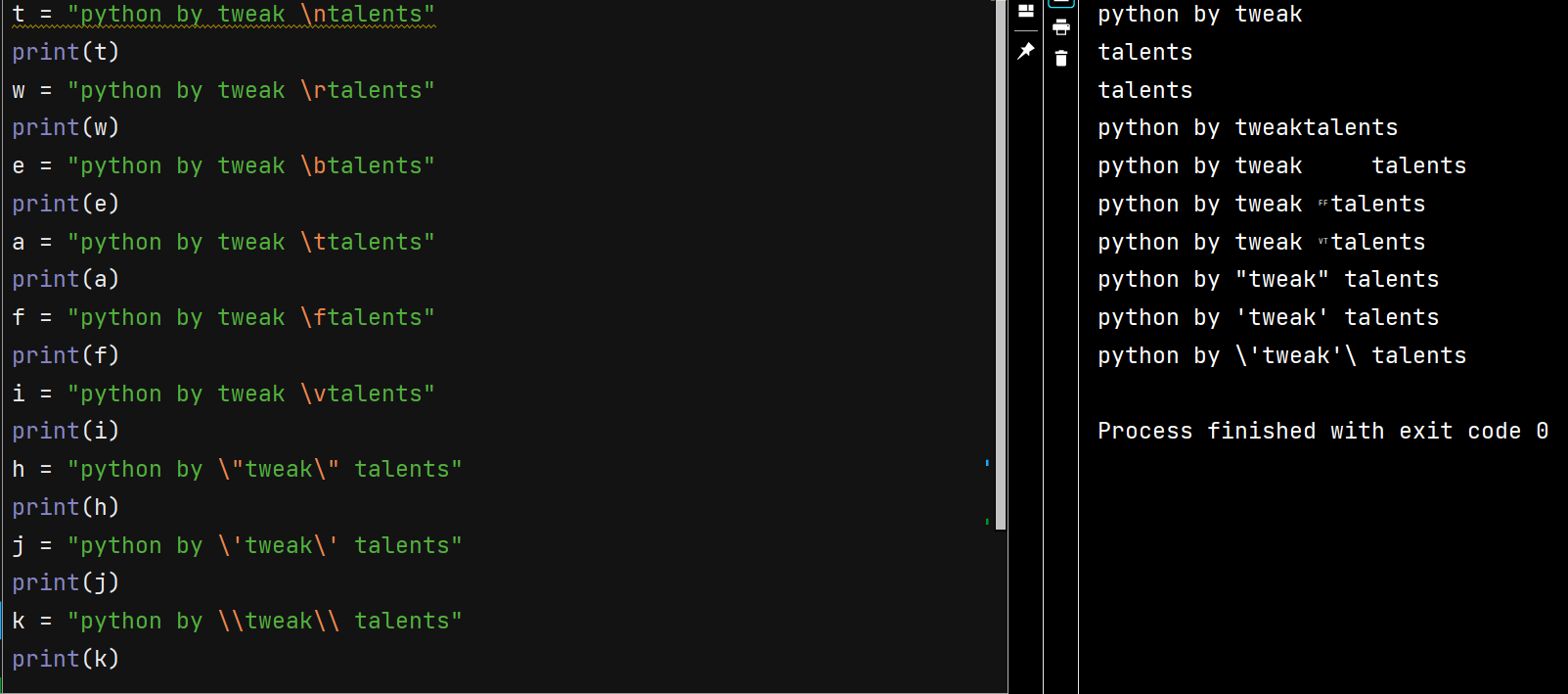
**\f** go to bottom of the page

**\’**

**\”**

**\\**

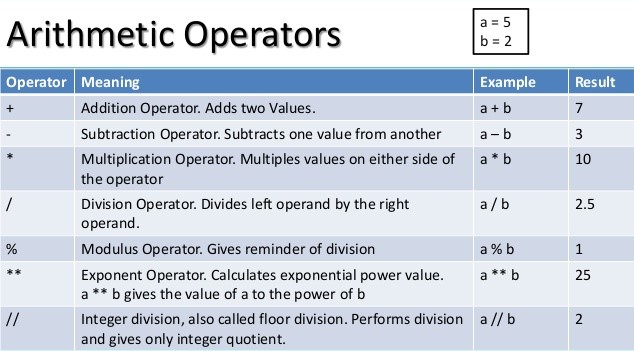
**\v** vertical tab



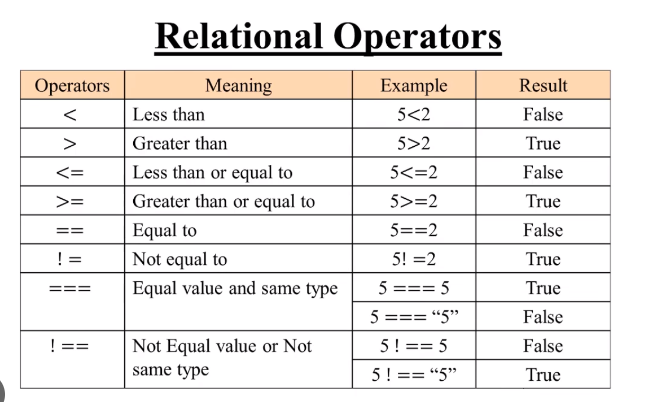
**OPERATORS**

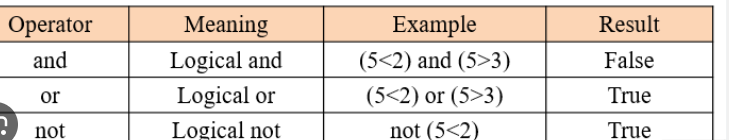
Operators are special symbols are keywords used to perform various operations on values.

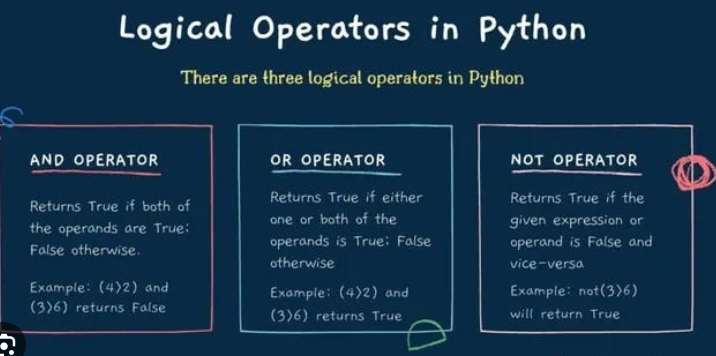
1. Arithmetic operator
2. Relational operator
3. Logical operator
4. Assignment operator
5. Bitwise operator
6. Special operator

**1)**

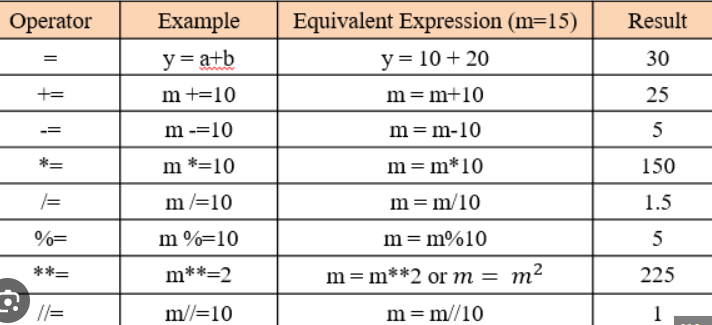
**2)**

****

**3)Logical operators**

****

**4)Assignment operator**

****

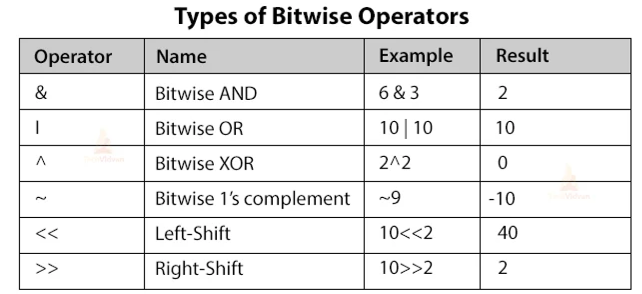
**5)Bitwise operator**

This is applicable for only integers and Boolean datatypes only

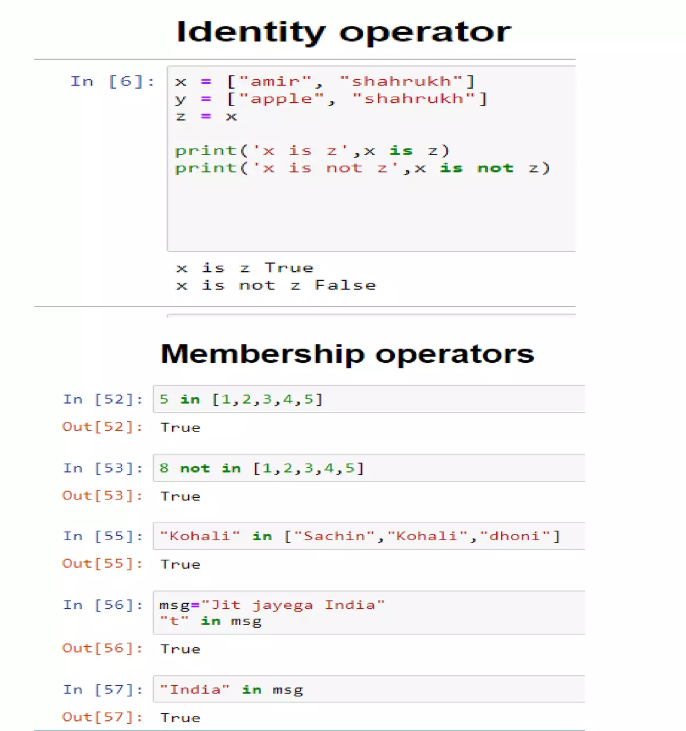
& if both bits are 1 then result is 1 otherwise 0

| if atleast 1 bit is 1 then the result is 1 otherwise 0

^ if both bits are different then result is 1 otherwise 0

****

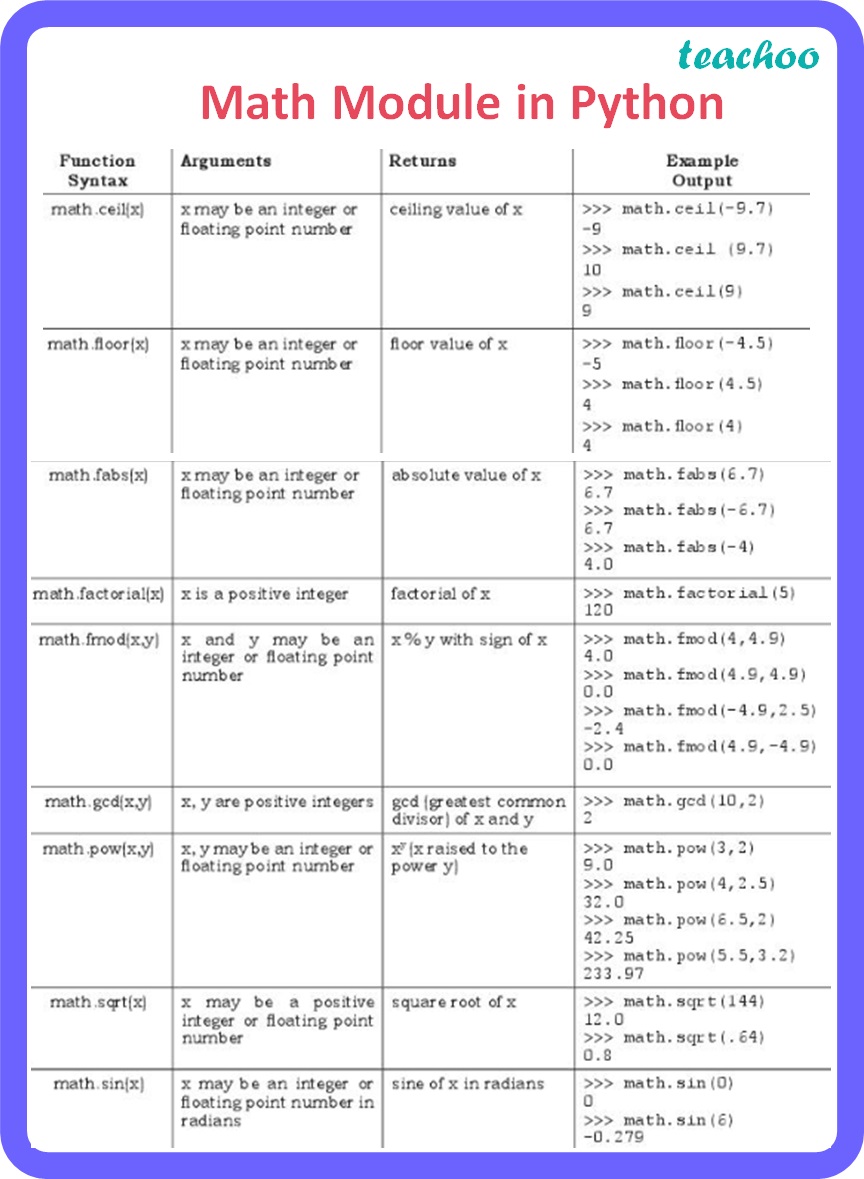
**6)Special operator**

****Python offers some special operators like identity and membership operator

**Modules**

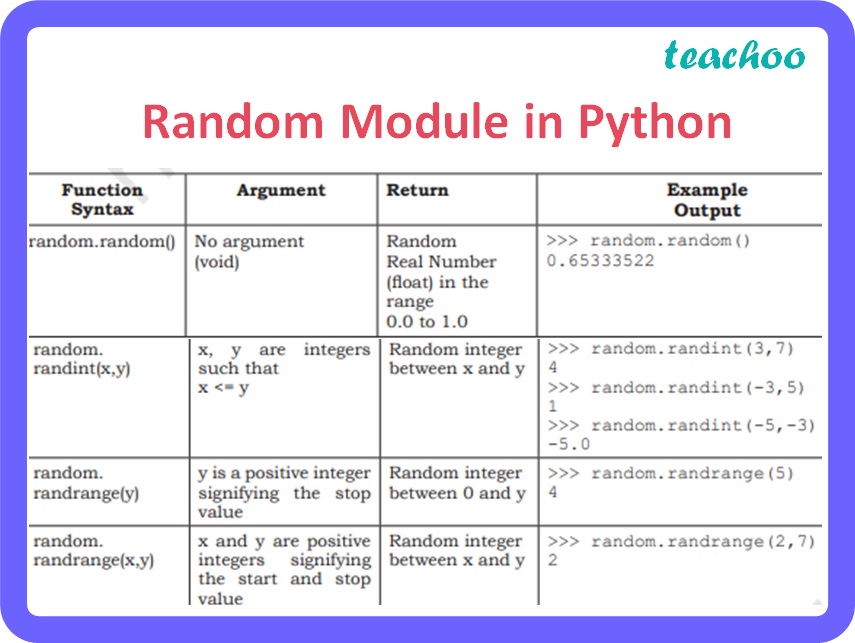
A group of functions, classes, variables know as modules. Every python file (.py) act as module. We can use them by importing them. Examples are math module, random module etc.,

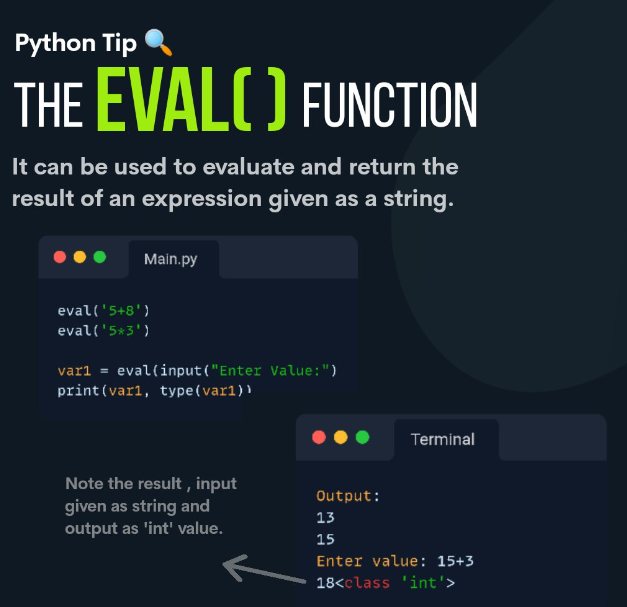
Import math

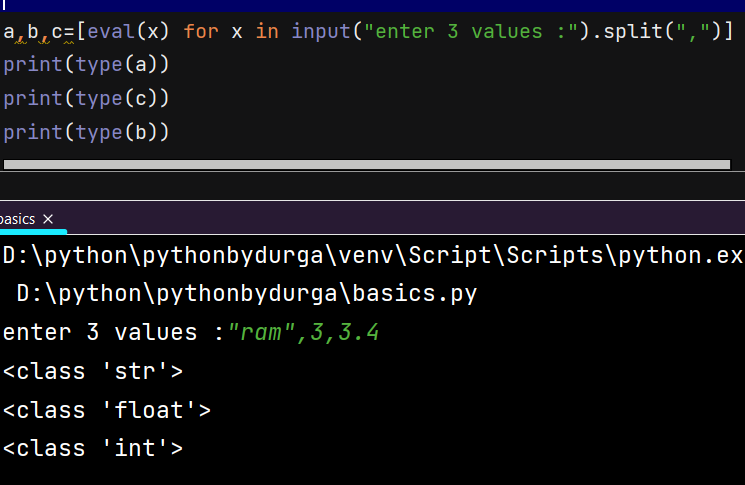
****

Practice all in your IDLE then you will get a clarity on each function

**Random module**

****

**Python eval() function**



**FLOW CONTROL**

At runtime in which order of the statements or code is executed is know as flow control statements. There are 4 types

1. Sequential statements
2. Conditional statements/selection statements
3. Iterative statements/Loops
4. Jumping statements/Transfer statements

1)sequential statements:

In this type the code executes line by line

2)Conditional statements: (if, if elif else, if else, nested if else)

It executes w.r.t condition, any one condition is executed

name = input("enter the name:")  
if name == "tweak":  
 print("Good morning")  
elif name == "Hyderabad":  
 print("welcome to Hyderabad, Have a great day")

else:  
 print("welcome to tweak talents")

here when u enter a name any one condition is executed.

Else is always optional

Practice yourself with more examples then you will be perfect in this.

3)Iterative Statements: (for, while, nested for)

A group of statements are executed

Ex: - for loop is used when you know iterations in advance

s="sunny Leone"   
count=0   
for i in s:  
 print(i)  
 count+=1  
print("number of characters:",count)

output is

s

u

n

n

y

l

e

o

n

e

number of characters: 11

While is used when you don’t know the iterations in advance

Ex: -

x=1  
while x<10:  
 print(x)

output:

1

1

1

1

1

1

1

1

Infinity…..

Here we don’t know when the condition is true. when you add (x+=1) then it will stop

4)Jumping Statements: (pass, continue, break)

**Break**

To exist from the loop based on some condition.

**Continue**

It skips only current iteration of the loop and continue the next iteration

**Pass**

for any future need to write code In a particular place then we just give pass statement. Pass can’t do anything.

p=["plreddy"**,1,2,3**]  
r=["plreddy"**,1,2,3**]  
print(p is r)  
print(p==r)

**False ,** Because list is mutable so in future we can change the list so that every list has individual object

**True**

“ Is ” operator is used for reference comparison

“ == ” operator is used for content comparison