**Python topics**

1. Language Fundamentals
2. Operators
3. Input and Output Statements
4. Flow Control
5. Strings
6. List Data Structure
7. Tuple Data Structure
8. Set Data structure
9. Dictionary Data structure
10. Functions
11. Modules
12. Packages

Advantages

1) concise code

2) Rich libraries(Batteries)

3)Object Oriented programming language

4) procedural programming language

5) Modular programming language

->day1 python shell is available.

6) Open source

7) How to install python?

Install python 3 because python is not going to support after 2020. And highly recommended to chose path.

8) Python 2 and Python 3 is totally independent.

9)REPL Tool : Python IDLE

Read, Evaluate ,print and loop again

10)python virtual machine have inbuilt capabilities of compiler then being a programmer no need to do compile.

10) Editplus,notepad++,python shell,editors

11)IDES

Pycharm,eclipse,spyder

12) It’s a general language(use for every purpose like web, desktop application, Internet of things(IOT),machine learning, artificial intelligence, data science so on) high level programming language(human readable code).

13)Python has been developed by Guido Van Rossam. came in 1989 when he was in NRI netherland .Official date is 20/02/1991.Popular show on BBC that is Monty Python Flying circus.

-> >a,b=10,20 is allowed in python but not allowed in java(statically typed programming).but python is dynamically typed programming language.

->in python type,; is no need to write.

->Features

1) Functional programming from C

2) OOP from C++

3) Scripting language(interpreted language) from perl and shell script

4) Modular programming language from Modula-3

Most Syntax borrowed from C and ABC language.

->Where we can use python?

1) Desktop(standalone) application(calculator)->internet not required

2) web application->Django,Flask frameworks to develop web application

3) database application

4)for networking applications

5)games

6) data analysis

7)machine learning

8)AI

9)IOT applications

* Features of Python

1. Simple and easy to learn
2. Only 30 keywords but in java 53
3. Free ware and open source(customize source code like Jython is java version for work with java)
4. High level programming language(human readable)
5. Platform independent(write once and run anywhere(WORA))
6. Portability(moving one platform to another without changing)
7. Dynamically Typed
8. Both procedural and object oriented
9. Interpreted(python interpreter will take care of compilation)
10. Extensible(Other language program in python)
11. Embedded(python program can use in any other language)
12. Extensive library

* Limititations

Performance is not upto the mark

Frequently not used for mobile applications

* Myth : Python is not suitable for large scale enterprise applications.
* Flavours of Python

1. CPython
2. JPython or Jython
3. IronPython
4. Pypy(JIT compiler inside python virtual machine)
5. RubyPython
6. AnacondaPython(Handing BigData)
7. Stackless(Python for concurrency)

* **Python Versions**

Python 1 : 1994

Python 2 : 2000

Python 3 : 2008

Any newer version doesn’t support for old version programs.

Python 3.6.3 : 2016 doesn’t support Python 2

Eg: print “Hello” is compatible but not in Python 3

Long datatype compatible in p2 but not in p3

**Topic**

**1) Python Features**

**2) Identifiers**

3) **Reserved Keywords**

4) Data Types

* **Identifiers** :

Name in python program is called identifier.variable,class,method name

Rules :

1. Alphabet symbols(both upper case and lower case),

digits(0,9),

underscore(\_)

1. Not starts with digits
2. Case sensitive
3. Keyword not used as identifier.
4. No length limit in python for identifier but not recommended.

**Note**:

Identifier start with underscore(\_) then it’s called private.

Identifier start with two underscore(\_\_) then it’s called strongly private.

Identifier start with two underscore(\_\_) and ends with two underscore(\_\_) then it’s called language specific identifier defined by python.

* **Reserved Keywords**

True,False,None

and,or,not,is

if,else,elif

while,for,break,continue,in,yield

try,except,finally,raise,assert, async, await

import,from,as,class,def,pass,global,nonlocal,lamda,del,with

Note :

Only alphabet symbol except first 3.

>>>import keyword

>>>keyword.kwlist

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

* **Data Types**

This is dynamic datatype programming language.in python everything is an object either variable or method.

Inbuilt databyte

Int

Float

Complex----------🡪Fundamental Datatype

Bool

Str

Bytes

Bytearray

List

Range

Tuple

Set

Frozenset

Dict

NoneType

* Inbuilt Method

Print()

Type()

Id()

* **Int**

Integral value

1)Decimal form(a=1111)

2)binary(a=0b1111 or 0B1111)

3)octal(a=0o556 or 0O556)

4)hexadecimal(a=0x1111 or 0X234 or 0xBeef) 0x,0X,0-9,a-f,A-F

>It will only display decimal value.

>Size or range such type of terminology is not available in python because everything is behave like objects.

>Base conversion : bin(),oct(),hex()

Long datatype is not there available in Python3 and available in Python2. So we have represent long value using int only.

* **Float datatype**

**>**only allowed decimal value apart from that octal,hexadecimal or binary is not allowed.

**>**exponential form allowed in float datatype.

* **Complex**

>x=a+bj

a=real part(int and float,any part of integral value is allowed)

b=imaginary part(int and float,only decimal value is allowed)

j^2=-1

>real=x.real,imaginary=x.imag

>use for mathematics based application

* **Bool datatype**

**>**internally True has been defined to 1 and False defined to 0

>b=True+True=2

* **Str datatype**

**>**b=’vikash’ or “vikash”

**>**char datatype is not available in python.

>if we want to write string in multi line then initialize triple single quote or triple double quote.

>eg:s=’’’durga sir “python” is very helpful’’’

>slice operator : to get substring : s[begin:end], s[begin:end:step],it will display from begin to end-1 and begin will be lower that end.

> if we are using ‘ or ‘’ as a symbol then string will be closed by non matching quotes or \ or ‘’’ or ”””.

>we can display data using index or slice operator.

>s[begin:end:step]

Step => +ve or –ve

+ve => forward direction

-ve => backward direction

If +ve forward direction from begin to end-1

If –ve backward direction from begin to end+1

Forward direction => begin=0,end=length,step=1(default value)

Backward direction => begin=-1,end=-(length+1),step=1(default)

Begin must be small from end

>len(s)

> \* repetition operator

e.g:s[2:8],s[-1],s[100],s[1:],s[:4],s[:],s[-4:-1],s\*10

>method for removing space from string

lstrip() => Remove space at beginning

rstrip() => Remove space at end

strip() => Remove space at beginning and end of the string

>find() =>s.find(substring,pos+1,n)

Index() =>

rfind() =>

rindex() =>

>counting substrings in the given string

s.count(substring)

s.count(sub,begin,end)

>replacing string

s.replace(old,new)

>splitting string

s.split(’- ’,3) => forward direction

s.rsplit(‘, ’,3) =>backward direction consider 3 commas then consider one string

>join string

S=’-’.join(l)

>changing case of string

upper()

lower()

swapcase()

title() => first letter will caps of every word

capitalize() => first letter will be in capital case

>checking starting and ending part of string

s.startswith(“learning”)

s.endswith(“easy”)

>check alphanumeric

s.isalpha()

s.isdigit()

>sorting

Sorted(s)

* **Typecasting Function or Type coercion**

int(12.3)-12

int(10+20j) : TypeError

int(True) : 0

int(“10”) : 10

float(10) : 10

float(10+20j) : NP

float(True) : 1.0

float(false) : 0.0

float(“10”) : 10.0

float(“10.5”) :10.5

float(“ten”) : value error

float(“0b1111”):valueerror

complex(x) : x+0j

complex(x,y) : x+yj

complex(“10”,”20”)->value error we can take only real as string

bool(0)->zero means false

bool(10)->non zero means True

bool(0.0)->false same rule as int

bool(0+1j)->True same as above rule

bool(“”)->False->empty as false, other cases True

str(1)->’1’

str(1+2j)->’1+2j’

* **Immutable Vs Fundamental Data types**

>all fundamental datatypes are immutable.

>Before creating the objects it will check that this value is available then it will not create separate object but it will point to same object. Now if I will change the contents then all the reference will be affected. Hence all fundamental datatypes are immutable.

>Memory utilization, performance will improved

>int => 0-256 | bool => always | str => always | float and complex => always create new objects

>at the time of python interpreter starts it will create object from 0 to 256 cooz this is commonly used.

>

>x=257

Y=257

X is y=False-🡪This is applicable for 0-256 for int datatype.

>x=10.0,y=10.0 x is y==false because reusing concept is not available.

>x=10+20j,y=10+20j x is y==false because reusing concept is not available.

* **Bytes and bytearray datatype**
* Used for binary data like images, video files and audio files.
* It represents a group of byte numbers just like an array.
* In the range 0 to 256.
* It’s immutable.
* Bytearray and bytes datatype both are same except this is mutable.
* **List datatype**
* Same as java but here null not allowed at the place of null None is allowed.
* Values should be enclosed with [].
* \* operator applicable for repeatation
* It’s mutable.
* Slice operator is allowed.
* List=[x\*x for x in range(1,11) ] or [x\*x for x in l1 if x%2==0]-list comprehension
* **Tuple datatype**
* Represented by ().
* Same as list except it’s immutable.
* For single element we have to take comma otherwise type will be int.t=(10,)
* **Range datatype**
* Represent a sequence of values.
* Immutable
* Only for int value.
* Range(10),range(10,30),range(10,100,5)
* **Set datatype**
* Don’t want duplicate and IO not preserved.
* Represented by {}.
* mutable
* Indexing, slicing not allowed
* **Frozenset datatype**
* Same as set but it’s immutable.
* S=frozenset(s)
* **Dict datatype**
* Represent key-value pair.
* {key:value}
* Mutable
* \* operator not allowed
* None datatype
* Method doesn’t return any value then it will return None.
* **Constant**
* No such terminology is there ,developer have to care.
* **Operatos**
* **Arithmatic**
* **All are same as java but two opeartors are different.**
* **+,-,\*,/,%,//,\*\***
* **//=>floor division operator**
* **\*\* => exponent operator or power operator**
* **If using + with string then all variable must be string.**
* **Relational**
* **>,>=,<,<=**
* **Chaining of relational operators allowed.10<20<30**
* **Equality Operator**
* **== ,!=**
* **== will do comparision between values but is operator will do comparision between address.**
* **Chaining allowed.**
* **Logical Operator**
* **Boolean Type**

**and => if both arguments are True then only True**

**or => if atleast one argument is True then True**

**not =>**

* **for non-boolean types:**

**0 means False**

**Non-zero means True**

**Empty string means False**

**E.g:x and y**

**if x evaluates to false then result is x otherwise returns y**

**x or y**

**if x evaluates to True then return x otherwise returns y**

**not x**

* **Bitwise operator**
* **& : if both bits are 1 then only one otherwise 0**
* **| : if atleast one bit is 1 then 1 otherwise 0**
* **^ : x-or : if both arguments are different then 1 otherwise 0**
* **~ : bitwise complement: 1 replaced by 0 and 0 replaced by 1**

**Store into 32 bit or 64 bit.First bit consider as signed bit**

**0 means +ve no and assigned directly into memory**

**1 means –ve no and assigned in 2’s complement**

**4 => 0000000000000000000000…100**

**~ 4=> 1111111111111111111111…011**

**1’s complement => 0000000000000000000…100**

**2’s complement => 0000000000000000000…101**

* **<< : bitwise left shift :**

**10<<2**

**00000000..1010**

**000000..101000 =>40**

**Right hand side vacant cells fill with 0’s.**

* **>> : bitwise right shift :**

**10>>2**

**00000000..1010**

**0000000000..10 => 2**

**Left hand side vacant cells fill with signed bit.0 for +ve and 1 for -ve**

* **Applicable for only for int and Boolean**
* **Assignment Operatos**
* **=,+=,-=,\*=,%=,//=,\*\*=,&=,|=,^=,>>=,<<=**
* **Ternary : x=30 if 10<20 else 40,x=30 if 10<20 else 40 if 10<20 else 50**
* **Identity Operators**
* **Used for address comparision.**
* **Is :both variable are equal then return True**
* **Is not : both variable are different then True**
* **Membership Operatos**
* **In : value is present then True hence False**
* **Not in : value is not present then True hence False**
* **Operator Precendence**
* **Module**
* **Import math or import math as m | from math import sqrt,pi | from math import \***
* **Read dynamic data from keyboard**
* **Read single data**

**Raw\_input(“aaaaa”)->always typecast into str | python 2**

**Input(“aaaaaa”) => not typecasting same as raw\_input() of python 2| python 2 & 3**

* **Read multiple data**

**A,b=[Int(x) For x in Input(“Enter 2 numbers:”).split(‘:’)]**

**Eval() => take string and convert into desired result**

**X=eval(input(“enter the list”)) =>convert into desired data**

**A,b,c=[ eval(x) For x in Input(“Enter 2 numbers:”).split(‘:’)]]**

* **Command line arguments**

**Argv is available in sys module**

* **Output statement**

**Print()**

**Print(str+str)**

**Print(str,str)**

**Print(a,b,c,”vikash”)**

**Print(a,b,c,sep=’:’) with sep attribute**

**Print(“hello”,end=’ ’) with end attribute**

**Print(“a value is %i” %a)#formatted string**

**Print(“hello {0} your salary is {1} and your Gf {3} is waiting”.format(name,sal,gf)) #replacement**

* **Flow control**
* **3types**

1. **Conditional/selection**

**If | if-else | if-elif-else**

1. **Iterative**

**For-each loop : execute body for every item in the given sequence.**

**While : execute body as long as some condition is true.**

1. **Transefer**

**Break : condition true then terminate execution**

**Continue : skip the current execution go for next iteration**

**Pass : want to provide empty block then use pass**

**\*\*\*\*For-else : if for will execute without break then else will execute**

**\*\*\*\*while-else :**

**\*\*\*\*del statement:want to delete an object permanently then object will be eligible for gc.e.g:del x,del x,y => delete variable and object both for single object. But if multiple variable is point to same object then only delete reference.**

**\*\*\*s1=None =>don’t want object but need variable**

* **Annonymous function**
* **Instant use**
* **Lambda keyword**
* **Filter(function,list)**
* **Map(function,list)**
* **Reduce(function,list)**
* **Nested function**
* **Only used inside the function**
* **Function decorator**
* **Use to modify our existing functionality without changing .**
* **Defined @decor\_fun on top of existing functionality**
* **Modules**
* **Variables**

**Instance variable**

**Django Application**

**>**python –version

>pip –version

> pip install virtualenvwrapper-win

> mkvirtualenv test

>pip install django

> django-admin –version

> django-admin startproject djangodemo

> python manage.py runserver

<http://127.0.0.1:8000/>

**>** python manage.py startapp calc

>create urls.py in app folder

>create one views.py in app folder

>mention our app url in project urls.py

> python manage.py collectstatic

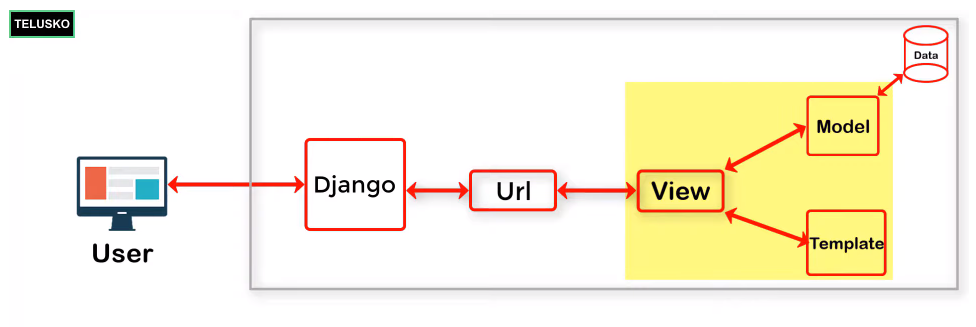
> pip install psycopg2(postgresql connector)

>pip install pillow(to use images)

> python manage.py makemigrations(migrate model to DB)

> python manage.py sqlmigrate travello 0001(to create table into DB)

> python manage.py migrate

****