PYTHON

Why python??

1. Interpreted vs. compiler

* 2. User productive -?(Versatile, Easy to Use and Fast to Develop.

)

Coding very simple, more equivalent to normal English

Example: other programming languages (Excluding braces)

//java: line of code-3

class hello

{

public static void main(String []ra)

{

system.out.println("Hello World");

}

}

//c++: line of code -4

#include<iostream>

using namespace std;

main()

{

cout<<"Hello WOrld";

}

//python: line of code 1

print(“Hello world”)

3. Why python?

Many facilities –web development, ML, AI, DS, interactive with DB, desktop application

Under Development: Cross platform mobile app development (currently in testing stage )

1. Presence of third-party modules  
   2) Extensive support libraries (NumPy for numerical calculations, Pandas for data analytics etc)  
   3) Open source and community development  
   4) Easy to learn  
   5) User-friendly data structures  
   6) High-level language –more equal to english  
   7) Dynamically typed language(No need to mention data type based on value assigned, it takes data type)  
   8) Object-oriented language  
   9) Portable and Interactive  
   10) Portable across Operating systems

Few packages come with py. But many packages are available.

math,os,sys,json,csv,itertools, (Come with py)

few which don’t come with python

Tensorflow-neural network

,pandas-DS calculation-series,dataframe

,Numpy-numerical py –numerical calculation- numpy arrays

Django –web framework

Matplotlib-data visualization –graph –pie chart bar graph

HTML-skeleton

CSS- Appearance skin

JS-Brain/blood

**Applications :**  
1) GUI based desktop applications(Games, Scientific Applications)  
2) Web frameworks and applications  
3) Enterprise and Business applications  
4) Operating Systems  
5) Language Development  
6) Prototyping

Disavn

1.slow processing

2. no native mobile app support

3.not easy to start py and move to other oop language

Cpp->py eaay

Py to cpp difficult

* 4. **Run-time Errors**

HOW TO RUN PROGRAMS in python’

1. File suffix - (.py)

2.where to run?

Two places- 1.ide/ interpret file/IDLE

1. Interactive python shell

All variable are objevts only

1.data types in py other language

Numeric-int ,float int float

String – string char,string

bool bool

Data structures-

list,dict,set,tuple arrays,map,set,

dict in other languages: map,associate arrays;

py features

all are objects

dynamically typed

semicolon inference

in other lang – indendation for clarity and readability

but in py it is a syntax ??why –no braces for blocks

list = [, ] square bracket mutable

set = {, } curly brace not at all subscriptable

tuple= ( ,) tuple immutable

dict=dictionary key-value pair not stored in order

{“key”:value,}

How to write comments in py

* 1. # hash
  2. Triple quotes –dirty /inefficient way –multi line 🡪 actualyy called as docstrings

‘’’ this is comment‘’’

# comment –cmd:

Below are iterables:

>>> x=[] list()

>>> y={} set()

>>> z=() tuple()

dict()

>>> type(x),type(y),type(z)

(<class 'list'>, <class 'dict'>, <class 'tuple'>)

Extend vs append

Pop(index) vs remove(value) vs clear()

Previous output in py interactive shell will be stored in “\_”

Typecasting str(),int(),float(),list(),dict(),set(),tuple()

If –no bracket but condition followed by colon and indentation for statements

If x>10:

Print(“Greater than 10”)

Else if is not there but elif is the else if condition followed by colon and indentation for statements

else: followed by colon and indentation for statements

looping :

while condition:

statements

Cmd to cloes?

exit

py close:quit() or exit()

Code testing

Microsoft Windows [Version 10.0.18363.1082]

(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\acer>d:

D:\>cd python

D:\python>python

Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:

This Python interpreter is in a conda environment, but the environment has

not been activated. Libraries may fail to load. To activate this environment

please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.

>>> print("Hello world!!")

Hello world!!

>>>

D:\python>python

Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:

This Python interpreter is in a conda environment, but the environment has

not been activated. Libraries may fail to load. To activate this environment

please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.

>>> x=[]

>>> y={}

>>> z=()

>>> type(x),type(y),type(z)

(<class 'list'>, <class 'dict'>, <class 'tuple'>)

>>>

KeyboardInterrupt

>>> x=[1,2,3,4]

>>> x

[1, 2, 3, 4]

>>> y={1,1,2,2,3,4,5,3}

>>> y

{1, 2, 3, 4, 5}

>>> type(y)

<class 'set'>

>>> z={}

>>> type(z)

<class 'dict'>

>>> z={1:"venky",2:"ashwin",3:"vaikunth",4:5}

>>> z

{1: 'venky', 2: 'ashwin', 3: 'vaikunth', 4: 5}

>>> x[0]

1

>>> x[1]

2

>>> y[0]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'set' object is not subscriptable

>>> y

{1, 2, 3, 4, 5}

>>> z[]1

File "<stdin>", line 1

z[]1

^

SyntaxError: invalid syntax

>>> zz={"venky":1}

>>> zz

{'venky': 1}

>>> zz["venky"]

1

>>> zz[1]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

KeyError: 1

>>> x

[1, 2, 3, 4]

>>> dir(x)

['\_\_add\_\_', '\_\_class\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_delitem\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getitem\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_iadd\_\_', '\_\_imul\_\_', '\_\_init\_\_', '\_\_init\_subclass\_\_', '\_\_iter\_\_', '\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_mul\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_reversed\_\_', '\_\_rmul\_\_', '\_\_setattr\_\_', '\_\_setitem\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']

>>> x.append(5)

>>> x

[1, 2, 3, 4, 5]

>>> x.append(1)

>>> x

[1, 2, 3, 4, 5, 1]

>>> x.count(2)

1

>>> x.count(1)

2

>>> yy=[123,124,125]

>>> x.extend(yy)

>>> x

[1, 2, 3, 4, 5, 1, 123, 124, 125]

>>> x.append(yy)

>>> x

[1, 2, 3, 4, 5, 1, 123, 124, 125, [123, 124, 125]]

>>> x.pop()

[123, 124, 125]

>>> x.pop(0)

1

>>> x

[2, 3, 4, 5, 1, 123, 124, 125]

>>> x.remove(123)

>>> x

[2, 3, 4, 5, 1, 124, 125]

>>> x.reverse()

>>> x

[125, 124, 1, 5, 4, 3, 2]

>>> x.sort()

>>> x

[1, 2, 3, 4, 5, 124, 125]

>>> x.clear ()

>>> x

[]

>>> x=()

>>> type(x)

<class 'tuple'>

>>> x=(10)

>>> type(x)

<class 'int'>

>>> x=(10,)

>>> type(x)

<class 'tuple'>

>>> x=(1,2,24,456,)

>>> x

(1, 2, 24, 456)

>>> type(x)

<class 'tuple'>

>>> x=([143],"djhasfsdhk",{1:"he"})

>>> x

([143], 'djhasfsdhk', {1: 'he'})

>>> len(x)

3

>>>