
Features of python

- ① Simple : Python is a simple programming language. When we read a python program, we feel like reading English sentences. It means more clarity and understanding program will become easy. Hence developing and understanding program will become easy.
- ② Easy to Learn : Python uses very few keywords. Its programs use very simple structure. Most of the language constructs which are also available in python.
- ③ Open source : There is no need to pay for python ~~which~~ software. We can easily download from python website.

Signature

its source code can be read, modified and can be used in programs.

④ Dynamically typed : we need not declare anything. An assignment statement binds a name to an object and the object can be of any type. If a name is assigned to an object of one type, it can be later be assigned to other type.

⑤ Platform independent : When a python

program is compiled, it generates byte code. byte code represents a fixed set of instructions that run on all operating system.

⑥ Portable : Python program will give the

same result since they are platform independent. Once a program is written it can run on any OS. Hence portable.

⑦ Huge Library : It has a very huge

library which can be accessed. Programmers can develop program very easily using modules available in python library.

App 2

Python objects

An object is simply a

collection of data (variables) and methods

(functions) that act on grosse data

e.g.: Rain → object



Dictionaries ③

A dictionaries represents a group of elements arranged in the form of key value pairs. The first element is considered as 'key' and the immediate element is taken as its 'value'. The key and its value are separated by a colon (:). All key - value pairs in a dictionary are inserted in the curly braces - {}.

Eg:- `dict = { 'Name': 'Shahzot', 'Id': 200, 'Salary': 9080.50 }`

Here the name of dictionary is 'dict'. 'Name' is the key and Shahzot is the value.
Similarly

'id' → key

200 → value

'Salary' = key

'9080.50' = value.

→ When the key is provided we can get back its 'value'.
Signature

IQ: Searching a value in the dictionary.

Sr. No. _____

Date: _____

dict['Name']
→ 'Shahzot'.

To know how many key value pairs are there in the dictionary we can use len() function

Eg: dict = {'Name': 'Shahzot', 'Id': 200, 'Salary': 9080.50}

n = len(dict)
print('No. of key-value pairs = ', n)

We can use any datatype for values. A value can be a number, string, list, tuple or another dictionary. But key should be unique, duplicate keys are not allowed. If same key is used again the value will be overwritten and the latest value is assigned.

→ We can use number, string, tuples as key since they are immutable.

→ We cannot use list or dictionary as a key.

Program is with laptop! Signature _____

~~W~~ Built-in attributes related to file object

Date: _____

St. No. _____

following are file objects most used attributes.

① closed - bool indicating the current state of the file object

② encoding - the encoding that the file uses

(access mode)

③ mode - the mode for the file in which it was opened

④ name - if the file object was created using open(), the name of the file, otherwise, some string the indicates the source of the file object.

⑤ ~~file~~ pythom, __pythom

In python the file system contains files and directories. To handle these files and directories python supports "os" module

The os module provides us the methods that are involved in the file operation. like renaming, deleting, get current directory, changing directory etc.

⑥ remove (Delete the specified directory)

Date: _____

St. No. _____

os.remove("directory-name")
a new file name)

os.rename("current file name", "New name")

os.remove("file") (used to remove the specified file)

os.rmdir()

os.remove("file-name")

os.mkdir("directory-name")

os.rmdir()

os.chdir("New-directory")

os.getcwd() (returns the current working directory)

os.chdir()

os.getcwd()

os.getcwd()

Signature _____

Signature _____

⑥ Exception handling in Python

Date: _____

Sl. No.: _____

Date: _____

Sl. No.: _____

→ when the errors can be handled, they are called exceptions.
To handle exceptions, the programmer should prefer the following steps.

Step ①: The programmer should observe the statements in the programs where there may be a possibility of exceptions. Such statement blocks should be written inside a try block.

In try block if some exception arises inside it, the program will not be terminated. When Python understands that there is a exception, it jumps into an 'except' block.

Step ②: Write the 'except' block where it should display the exception details like user. This helps the user to understand that there is some error in the program. The programmer should also display a message regarding what kind of errors can be done to avoid error.

The statements written inside an except block are called 'handlers'.

Step ③: Lastly the programmer should perform clean up action like closing the files and terminating any other processes which are running. This code is written in the finally block.

All 3 steps are exception handling

⑦ Assertions

The assert keyword is used when debugging code. It let you test if a condition in your code returns true or false, if not, the program will have an assertion error. Within our can write a message if the code returns false. Eg.

```
x = "hello"
```

```
#if condition evaluates False, Assertion  
error is raised:  
assert x == "good by", "x should be  
Hello"  
Signature _____
```

⑧ Raising Exceptions

Date : _____

Sr. No. _____

As a python developer you can choose to throw an exception if a condition occurs. To throw (or raise) an exception, use the raise keyword. We can define what kind of error to raise, and text to print to the user.

Eg:-

Raise a TypeError if x is not an integer.

x = "hello"

if not type(x) and:

raise TypeError("only integers are allowed.")

Q ~~Module related to files~~

CSV → Allows access value files

file cmp → Comparison of dictionaries
and files

file input → interface with multiple lines
of text file.

getopt → It provides command line
argument passing or manipulation
of type of file.

tempfile → To create and generates a
temporary files

Signature _____

The `zipfile` module → offers a number of

Date : _____

Sr. No.

high level operations on files and
collections of files.

`zipfile` → It provides tools and utilities
to read & write Zip Archives
files.

built-in types

Date: _____

Sr. No. _____

Date: _____

Sr. No. _____

A datatype represents the type of data stored into a variable or memory. The data types which are already available in Python language are called Built-in datatypes.

Some Built-in Data types are 5 types

- None type ✓
- Numeric types ✓
- Sequences [Already done]
- Sets []
- Mappings

Complex data types

: It is written in the form of $a+bi$: a represent real part
 $b \rightarrow$ imaginary
 $j \rightarrow$ square roots value.

Mapping types : It represents a group

of elements in the form of key, value pair . We can store a value associated with the key.

But compare us dictionary.

[Write about dictionary].

Numeric types

: Represent numbers.

- int
- float
- complex

Unit datatype

Signature _____

It represents an integer number, with out any decimal or fraction part.

float datatype : It represents floating point number.

we can define a function using the keyword def followed by function name . After the function name, we should write parentheses () which may contain parameter.

Signature _____

⑩ Different operators [Standard type operators]

→ An operator is a symbol that performs an operation. An operator acts on some variables called operands.

* If an operator acts on single variable it is called unary operator

* If " " two variable " " binary operator

* If " " three variable " "
ternary operator.

→ The operators depending upon their nature.

There are 8 operators mainly used.

1. Arithmetic Operators

→ These operators are used to perform basic arithmetic operator like addition, subtraction, division etc.

Eg:- Addition operator (+) $a+b$

Subtraction operator (-) $a-b$

Multiplication operator (*) ~~a*b~~

Division operator (/) a/b

Modulus operator ($\% \%$) $a \% b$
Exponent operator ($**$) $a ** b$

Date: _____ Sr. No. _____

Integer division ($/$) a / b

When this operator is used before a variable its value is unmodified
Date: _____ Sr. No. _____

print(n)

$\Rightarrow -10$

④ Relational operator

These operators are useful to extract the right side value into a left side variable. They can also be used to perform simple arithmetic operations like addition, subtraction etc. and then store the result into a variable.

Eg:- ① Assignment operator $z = x + y$

② Addition assignment operator $z += x$

③ Subtraction assignment operator $z -= x$

④ Multiplication assignment operator $z *= x$

⑤ Division assignment operator $z /= x$

⑥ Modulus assignment operator $z \% = x$

⑦ Exponentiation assignment operator $z **= y$

⑧ Floor division assignment operator $z // = y$

9. Unary Minus operator

\rightarrow The unary minus operator is denoted by the symbol minus (-)

$\text{False} = 0$

$\text{True} = \text{any other number}$

Signature _____

Signature _____

⑨ Logical operators : logic operators are useful to construct compound conditions. A compounded condition is a combination of more than one simple condition.

\rightarrow Each of the condition are evaluated to true or false

$\text{False} = 0$

Signature _____

Signature _____

② OR ($X \text{ or } Y$)

Date: _____

Sr. No. _____

- ③ Not $\rightarrow \text{not } X$, $\text{not } Y$

④ Not in : vice versa of in
Ex: $\text{Not in : } \text{vice versa of in}$

Date: _____

Sr. No. _____

⑤ Boolean Operators

\rightarrow There are two 'bool' type literals

- ① True ② False

The result provided by Boolean operators will be again True or False.

⑥ Identity Operator

and $(x \text{ and } y)$
or $(x \text{ or } y)$
not $(\text{not } x)$

\rightarrow These operators compare the memory location of two objects. Hence it is possible to know whether the two objects are same or not. The id() function is used to get the memory location:

Ex: $a = 25$

$b = 25$

$\text{id}(a)$

167095

$\text{id}(b)$

167095

There are two types of operator

- { ① id in
② not in

Two operator ① is ② is not

is useful to compare

Same \downarrow
Same \downarrow
Not same \downarrow

is useful to compare

Same \downarrow
Same \downarrow
Not same \downarrow

This operator return True if an element is found in the specified sequence. Else return False.

Signature: _____

→ Built-in functions

The Python built-in functions are defined as the functions whose functionality is predefined in Python. Some of the built-in functions are discussed below

abs :- Returns the absolute value of a number

Signature _____

Date : _____

Sr. No. _____

dict) Returns the binary version of a
number

dict) :- Returns a dictionary (array)

float) :- Returns float value

id) :- Returns the id of the object

list) :- Allowing users input

list) :- Returns List

int) :- Returns integer values

print) :- Print the output

Sequenzen am Python

→ A sequence represents a group of elements. Or idem. There are 6 types of sequences in Python.

- Str
- bytes
- bytearray
- List
- tuple
- Range

Data type

→ str represents string data.
Data: _____
Sr. No. _____

Type A string is represented by a 'get' of char. Strings are enclosed in single quotes or double quotes.

e.g.: "welcome" { storing
str = 'welcome'

print(str)

print(str[0])

print(str[3:-1])

print(str[3:-1])

print(str[-1])

print(str*2)

decoding accessed

bytes Data type

The bytes Data type represents a group of byte numbers. A byte number is any positive integer.

from 0 to 255 (inclusve).

Bytes array can store numbers in the range from 0 to 255 it cannot store negative numbers.

e.g. elements = [10, 20, 0, 40, 15] { store X = bytes(elements) }

print(X[0]) → Accessed

byte array Data type

Date: _____
Sr. No. _____

same as bytes Data type but can be modified.

e.g.: element = [10, 20, 0, 40, 15]
X = bytes(element)
print(X[0])

X[0] = 69

X[1] = 40

list Data type

A list represents a group of elements. The main difference between bytes and list is that list can store different types of elements.

The list Data type represents a sequence of brackets [] stored.

e.g.

list = [10, -20, 15.5, 'Shahzad']

print(list)

print(list[0])

print(list[1:3])

print(list[-2:3])

print(list*2)

Accessed

tuple as discussed previously.

tuple Data type : The tuple Data type represents a sequence of numbers.

The numbers in the range are not

modifiable.

Sr. No. —

modifiable.

Date:

$y = \text{change}(x)$

tuple Data Type

A tuple is similar to a list. A tuple contains a group of elements which can be of different types. The elements in the tuple are separated by commas and enclosed in parentheses ().

If it is not possible to modify the tuple elements. tuple is a read-only list.

Signature _____

Date: _____

Sr. No. _____

The slicing operations which can be done on lists are also valid in tuple.

For example program are as follows.

Some basic operation's

```
print(tpl)
print(tpl[0])
print(tpl[1:3])
print(tpl[-2])
print(tpl[2])
```

```
print tpl[0]=aa X
```

⑫ Context management & Exception handling

→ A context management usually takes care of setting up some resource e.g. opening a connection, and automatically handles the clean up when we are done with it.

→ The exception will be converted into a string i.e., the and passed into the variable string. The string variable will be printed out in the shell at the end.

⑬ 5 Standard exception of Python

Exception

- AssertionError
- cause of error
- Raised when an assert statement fails

AttributeError

③ Attribute Error

Date : _____

Raised when
attribute assignment
or reference fails

④ EOFError

Date : _____

Raised when the
input function
hits end of file
condition

⑤ FloatingpointError

Date : _____

Raised when a
floating point
operation fails

⑥ ImportError

Date : _____

Raised when the
imported module
is not found.

⑦ Using getopt module

Date : _____

⑧ Standard file & command line arguments

Date : _____

getopt module extends the
processing of the input by parameter
validation. It allows both short, long
options including a value assignment.

To use it, it is required to:

remove the first element from the list
of command - line arguments

⑨ Using argparse module

Date : _____

It provides a lot of options such
as positional arguments, default value
for arguments, help message, specifying
data type of argument etc.

This module provides access to
some variables used or maintained

Signature _____

Standard files

They are two types of files that can be handled in python, they are

- ① Normal text files
- ② Binary files

① Each line is terminated with a special character called as EOL
(End of line)

② Written in binary language 0's and 1's.

⑤

Python module for serialization and persistence?

Date : _____

Sr. No. : _____

It is possible to store the state of a Python object in the form of bytes in a stream, dumping to a file, or a memory stream and restore it to its original state. This process is called as serialization and deserialization process.

→ The term state persistence means it continues to exist even after the application has ended. Thus data stored in a non-volatile storage medium such as a disk file.