

Que: Introduction to Python and its Features (simple, high-level, interpreted language).

Ans:

Python is a machine learning, interpreted, high level programming language with dynamic semantic. In python variable do not need declaration, allowing access to many external libraries, highly portable, Runs almost anywhere (windows, Linux, iOS), Clean and clear syntax.

Uses: data science, data analysis, machine learning, web development.

Features of python:

Easy to learn and use

Interpreted language

Dynamically typed

object oriented

Extensive standard library

open source

Que: History and evolution of Python.

Ans:

Python was designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed to clean and clear code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Compare to java, c, c++ it used fewer code to express the concept

Que: Advantages of using Python over other programming languages.

Ans:

Simple to learn and use, user-friendly built-in data type, less coding, access to external library, high performance (smart memory management).

Que: Understanding Python's PEP 8 guidelines.

Ans:

PEP (Python Enhancement Proposal) 8 is Python's official style guide, providing guidelines for writing clean and consistent code, covering topics like naming conventions, code layout, indentation and whitespace usage.

Que: Indentation, comments, and naming conventions in Python.

Ans:

Indentation:

Use 4 spaces per indentation level (avoid using tabs).

comment:

4 spaces between the " # " symbol and the comment text

```
#, """ """
```

Maximum Line Length:

Limit all lines to a maximum of 79 characters

Naming Conventions:

Python has specific conventions for naming variables, functions, classes, and modules.

Classes:

Use CamelCase (first letter capitalized, no underscores).

Variable name:

use snake_case, Must Begin with a Letter or Underscore. Contain Letters, Numbers, and Underscores, Case-Sensitive, cannot be reserved keywords.

Que: Understanding data types: integers, floats, strings, lists, tuples, dictionaries, sets.

Ans:

Data type which is represent different type of values.

Int(integer): It represent whole number

Float: It represents number with fractional part.

String: It is a collection of character. It represents text data.

List: It is a collection of similar and dis-similar data elements.it is ordered, mutable.

Tuple: It is a collection of similar and dis-similar data elements.it is ordered, immutable.

Dictionary: it is a most power concept in python. It contains key and values in a pair.

Sets: set is an unordered collection of unique elements. It is a built-in data type that allows you to store multiple items in a single variable

Que: Python variables and memory allocation.

Ans:

In python variables are dynamic. It means their type is determined at runtime. Variables do not need to declare.

memory allocation:

stack memory:

it is automatically managed by the system. When function is called, its local variable and control information are pushed on stack. memory allocation and deallocation happen automatically.

Stores:

Function calls, Local variables, Control flow

fast access:

Memory allocation and deallocation are automatic, once a function finishes execution, its stack frame is removed.

heap memory:

it is manually managed. it stores object, arguments and large data structures. memory allocation and deallocation manually.

stores:

Objects, Large data structures

slow access:

Requires garbage collection to free up unused memory. (manually allocation and deallocation memory)

Que: Python operators: arithmetic, comparison, logical, bitwise.

Ans:

To perform specific operation, we need to use symbols that symbols called operators.

Arithmetic operator:

Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modules
//	Floor division
**	Exponent

Comparison operator:

Operator	Name
==	Equal
!=	Not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Logical operator:

Operator	Name
And	And operator
Or	Or operator
Not	Not operator

Bitwise operator:

It is used to compare binary numbers.

Operator	Name	Description
&	AND	Return 1 if both operand bits are 1
	OR	Return 1 if at least one operand bit is 1
^	XOR	Returns 1 if one of the operand bits is 1 but not both
~	NOT	Reverse all bits
<<	Left shift	Operand shift left
>>	Right shift	Operand shift right

Que: Introduction to conditional statements: if, else, elif.

Ans:

With use of conditional statement we can perform a specific action according to the condition.

If statement:

In this, condition is compared with value which is given by user or already defined, if condition is true then it will execute the block inside if statement.

If condition:

Block of code

If else:

In this, condition is compared with value which is given by user or already defined, if condition is true then it will execute the block inside if statement. Or if condition is false than it will execute code inside else statement.

If condition:

Block of code

else:

Block of code

elif:

In this, when we have more than one condition then we used elif condition. It is also called as ladder if else condition

If condition:

Block of code

elif condition:

Block of code

elif condition:

Block of code

else:

Block of code

Que: Nested if-else conditions.

Ans:

In this, when we have nested conditions it means condition inside a condition then we use it.

If condition:

 If condition:

 Block of code

 else:

 Block of code

else:

 If condition:

 Block of code

 else:

 Block of code

Que: Introduction to for and while loops.

Ans:

Loop is mainly use for repetition of specific code or statement.

For loop:

For loop is a sequence control loop it means we can decide how many time loop will execute.

In for loop we use range() function. By using of range function we can control loop where to start and where to end and also in how many steps.

By default for loop starts from 0 and end last-1.

Range(start,end,step)

Ex.

```
for i in range(1,6):
```

```
    print("hello")
```

it will print hello five times

While loop:

While loop is a entry control loop, it means we don't know how many times it will run but it terminates when specific given condition is false.

While condition:

Block of code

Que: How loops work in Python.

Ans:

loop is mainly use for repetition of specific code of specific statement until given condition is satisfied. nested loop allows to use loop inside a loop.

while loop:

in this, code will execute until give condition is satisfied. when the condition becomes false the code after loop will execute.

initialization

while condition:

block of code

for loop:

for loop is a sequence control loop it means we can decide how many time loop will execute.

In for loop we use range() function. By using of range function we can control loop where to start and where to end and also in how many steps.

By default for loop starts from 0 and end last-1.

Range(start,end,step)

Ex.

```
for i in range(1,6):
```

```
    print("hello")
```

it will print hello five times

Que: Using loops with collections (lists, tuples, etc.).

Ans:

we can use loop with collections. In this using loop we can extract elements inside list and tuple.

Use of loop in list:

List is a collection of similar and dis-similar data elements. It is mutable means we can change its value.

Ex.

```
l1 = ["raj", 45, "patel"]
```

```
for i in l1:
```

```
    print(i)
```

use of loop in tuple:

tuple is a collection of similar and dis-similar data elements. It is immutable means we cant change its vlaue

Ex.

```
t1 = ("raj", 45, "patel")
```

```
for i in t1:
```

```
    print(i)
```

Que. Understanding how generators work in Python.

Ans.

Generator defined just like normal function. But it use yield instead of return.

It generate result when its needed.it is useful when we want to work with large number of data it does not store values in memory.

We can print yield values using next().

Que. Difference between yield and return.

Ans.

Yield is used in generator. It store more than one result value and it does not occupies memory. And return value one by one using next() function.

Return store only one value and it occupies a memory.

Que. Understanding iterators and creating custom iterators.

Ans.

With use of iterator, we can travel through all elements of iterable. And access them one by one.

Using iter() function we can initialize the iterator.

Using next() function we can access data elements.

For creating custom iterator:

First take one list or any iterable and then use iter function in new variable. And then print using next() function one by one.

Ex.

```
l1 = [1,2,3,4,5]
data = iter(l1)
print(next(data))
print(next(data))
print(next(data))
print(next(data))
```

Que. Defining and calling functions in Python.

Ans.

To define a function we use def keyword and then function name and parentheses.

To call function write function name and then parentheses.

Que. Function arguments (positional, keyword, default).

Ans.

Positional arguments:

In this, arguments are assigned to parameter in ordered.

Keyword arguments:

In this, arguments are pass as a parameter name.

Default argument:

When no argument pass in function, the default value applied to function.

Que. Scope of variables in Python.

Ans.

Scope of variable depend on the area of where variable can access or modify.

Global scope:

When the variable is define top of program or outside any block or function its called global variable. We can access anywhere in program.

Local scope:

When the variable is define inside a block of code its called local scope variable. We can only access only inside block.

Enclosed:

When we have a function define inside of another function, the outer most function is accessible inside nested block.

Que. Built-in methods for strings, lists, etc.

Ans.

string methods:

- upper()
- lower()
- capitalize()
- strip()
- title()
- replace()
- find()
- split()
- join()
- isalpha()
- isdigit()
- isalnum()
- startswith()
- endswith()
- zfill()

list methods

- len()
- append()
- insert()
- extend()
- pop()
- remove()
- index()
- reverse()
- sort()

Que: Understanding the role of break, continue, and pass in Python loops.

Ans:

break, continue and pass are types of control statements. It is mainly use for alter the flow of loop.

break:

in this, when the given condition is satisfied, the loop while terminated even if the loop condition is true.

```
for i in range(1,6):
```

```
    if i==4:
```

```
        break
```

continue:

in this, when the given condition is satisfied, the loop will skip that part and continue the loop.

```
for i in range(1,6):
```

```
    if i==2:
```

```
        continue
```

pass:

in this, it is null operator in python, it does nothing when executed. It is used as placeholder where syntax required.

```
for i in range(1,6):
```

```
    if i==2:
```

```
        pass
```

Que: Understanding how to access and manipulate strings.

Ans:

string is collection of characters. it represents text data. in python string which is represented by double quotes " " and single quote ' '. string which is immutable - we can't change it's value.

for access string:

```
str = "python"
```

```
print(str)
```

manipulate string:

```
len()
```

```
lower()
```

```
upper()
```

```
title()
```

```
capitalize()
```

```
strip()
```

```
find()
```

```
replace()
```

```
startswith()
```

```
endswith()
```

```
split()
```

```
join()
```

```
isalpha()
```

```
isdigit()
```

```
isalnum()
```

```
zfill()
```

Que: Basic operations: concatenation, repetition, string methods (upper(), lower(), etc.).

Ans:

concatenation: for concatenation of string use “ + “ .

repetition : for repetition of string use “ * “.

string methods:

len(): Returns the length of a string

lower(): Converts all characters in the string to lowercase.

upper(): Converts all characters in the string to uppercase.

capitalize(): Converts the first character of the string to uppercase and the rest to lowercase.

title(): Converts the first character of each word in the string to uppercase.

strip(): Removes any leading (spaces at the beginning) and trailing (spaces at the end) whitespace characters from the string.

replace(): Replaces a substring within the string with another substring.

string.replace(old, new, count)

find(): Searches for a substring and returns the index (position) of the first occurrence. If the substring is not found, it returns -1.

startswith(): Checks if the string starts with the specified substring.

endswith(): Checks if the string ends with the specified substring.

split(): Splits the string into a list of substrings based on a specified delimiter (default is whitespace).

join(): Joins a list of strings into a single string, with a specified separator between each element.

isalpha(): Returns True if all characters in the string are alphabetic

isdigit(): Returns True if all characters in the string are digits

isalnum(): Returns True if all characters in the string are alphanumeric

zfill(): fill the string with zero until specific length satisfied.

Que: String slicing.

Ans:

in this, we can extract a specific portion from string. It allows you to access specific part of a string using index value.

syntax:

`string[start:end:step]`

start= the index where slicing begins and index character included in result value.

stop= the index where slicing ends and index character not included in result value.

step= it decide how many character to skip.

```
str = "pythonprogramming"
```

```
print(str[0:4])
```

```
print = pyth
```

Que. Using map(), reduce(), and filter() functions for processing data.

Ans.

Map:

in map, its apply condition on all iterable and return a result with all elements of iterable.

Filter:

In filter, its apply condition on all iterable and return filtered result according to condition.

Reduce:

In reduce, its apply condition on all iterable and return single statement result.