1.Introduction to python and its Features(simple, highlevel, interpreted language).

Python is an object oriented, high-level-general purpose, interpreted programming language, created by Guido van rossum in 1991. Python has a clear syntax with extensive libraries support which makes it perfect for development, python used in web-dev, data-science, machine learning,AI, automation, desktop apps.

FEATURES →

Simple python is easy to read and write because it have a simple clear and straightforward syntax
High-leven : python code syntax is closely resembles to english which makes it easy to understand
Interpreted : python uses interpreter which executes python code line by line on runtime and display output

2.History and evolution of python.

Python was created by Guido van rossum in 1991 who intended to create a language which is easy to read and write and provide more easy syntax then other programming languages.

Evolution :

1919 : python 0.9.0 released publically first time.
1994 : python 1.0 released which added support for exceptions and other features.
2000 : python 2.0 released which introduced **PEP8** official python style guide, garbage collection and automatic memory management
2008 : python 3.0 released, it removed support of many features from python 2.X and introduced a new standard library, new data types. And a new language syntax

3.Advantages of using python python over other programming languages.

- Cross platform support (windows, linux, macos)
- Simple and minimalist syntax
- Huge libraries support
- Large community support : for learning resource and open source contributions
- Versatile application

4.installing python and setting up the development environment.

For window go to https://www.python.com and download the latest installer exe for python, install the exe, while installing it ask check to add path environment variables (check it), then install, after open any terminal application and type python –version to ensure python installed. download a code editor like : vscode from https://code.visualstudio.com install it go to extensions tab and download python extension from microsoft,
Create a new file with .py extension, write a hello world program and run int.

5.write and execute your first python program.

Install latest python open a code editor create a file with.py extension wite print(“hello world”) in it open terminal locate to directory where your file saved, type python filename.py and hit enter it will execute the fiel and display output.

6.Understanding pep 8 guideline.
pep 8 stands for python enhancement proposal, pep8 is official style guide for python that provides the best practice to write python programs.
Pep 8 includes →
Code layout, use of indentation, max-line length, naming-conventions, use of white space, and case syntax for variable, function and classes

7.indentation, comments and naming-conventions in python.
Indentation : according to pep 8 use 4 space for indentation to indicate each level of depth in block
Comments : use a # symbol before writing any single line comment, wrap text in “””” “””” for multiline comment
Naming-conventions : for naming variables and functions use snake_case without any space or first character should start with a alphabet character, variable name not include any special Characters, use meaning full names for naming variables and function that describes there Use, for classes use Pascal Case,

8.Write rateable and maintainable code.

- To write readable and maintainable code,
- always follow the naming conventions for variables function and classes,
- use comments for sections of code,
- use complex logics and repeated code as functions,
- use object oriented features if applicable for better code reusability.

9.Understanding data types: integer, float, string, list, tuples, set, dictionaries.
Int: int stands for integer, contain only whole numbers without fractional portion, can be negative or positive

Float: float stand for floating point, contain decimal portion, can be positive or negative

Str: stands for string, strings are sequences of characters enclosed in double or single quotation marks**.** Strings are immutables

List : a type of collection which is represented by [ ], list is ordered,indexed and mutable, and contains multiple elements of same or different data types and allows duplicate values.

Tuple: a type of collection which is ordered, indexed but immutable, allowing multiple elements of similar and dissimilar data types. Represented by ( )
Dictionaries: a type of collection which stores data in a unique key : value paire, it represented by {key : value } , dictionaries are unordered, and mutable and allow similar and dissimilar data type values

10.Python variables and memory allocation.
Python variables are dynamically typed,and we don't need to specify the type before creating a variable. Python is dynamically typed, that is why it has memory management and a garbage collection that allocates delicate memory on runtime automatically.

11.Python operators: arithmetic, comparison, logical, bitwise
Arithmetic: →

+	addition
-	subtraction
*	multiplication
/	division
//	floor division
%	modulus

Comparison operators →

==	Equal
!=	NotEqual
<	less than
>	greater than
<=	less than Equal
>=	greater than Equal

logical operators →

and	logical and operator
or	logical or operator
not	logical not operator

bitwise →

&	bitwise and
|	bitwise or
~	bitwise not
^	bitwise xor
>>	right shift
<<	left shift

12.Introduction to conditional statements if else , elif
Conditional statements are used to execute different blocks of code on a specified condition evaluates as true or false.
If : used to execute a block of code when a condition happens to be true
else : when if statements condition is evaluated as false than statements written in else block are executes
elif : to check another condition when if condition is false

13.Nested if else condition.
When we have to check multiple condition inside a single if else block we nest another if else block inside it which is called nested if else.
When first if block condition evaluates as true than it check for another condition inside if block.

14.Introduction to for and while loop.
For loop is a sequence control loop it iterates over a sequence or an iterable and executes a block of code for each item in sequence.

```
for i in range(1,11):
        print(i)
```

While loop is entry control loop, it executes a block of code while a certain condition is true.

```
X = 0
while x < 10:
    print(x)
    X += 1
```

15.How loop works in python
Loop is used for repeatedly executing a block of code while a certain condition is true.
Loop has an initializer a condition and update for each iteration
If the condition is true the block of code inside the loop executes and initializer value get updated and it continues till condition inside loop not false.

```
   x = 0
   while x < 10 :
      print(x)
      x += 1
```

16.Using loop with collections (list, tuple, etc).

```
 l1 = [1,3,4,5,7]
   t = (2,3,5,7,8)

 mydict = {
      "name" : "kp",
      "age"  : 23
   }
```

loop with list

```
    for item in l1:
        print(item)
```

loop with tuple

```
    for item in t:
        print(item)
```

loop with dictionary

```
    for k in mydict.keys():
        print(mydict[k])
```

17.Understanding how generators work in Python.

18.Difference between yield and return.

19.Understanding iterators and creating custom iterators

20.Defining and calling functions in Python.

Functions are blocks of code that are executes on calling and used for reusing the complex logic and repeated code blocks.
To define a function use def keyword in python after write function name followed parameters in () parenthesis
To call a function type function name followed by parameters in parenthesis()

```
def greetings(name) :
        return "hello " + name

greetings("kp")
```

21.Function arguments (position, keyword and default).

positional arguments : positional arguments assigned to parameters in the ordered they are passes, first argument will assigned to first parameters

keyword : when function's passed argument and parameter has same name, the order of passing not matter as long as name is same

default : in function definition a parameter can be assigned a default value if no argument provided

22.Scope of variables in python
Scope refers to the area or section of program where a variable can be accessible and modified

Scopes in python:
Global : when variable defined on top of program or outside any block or function it has a global scope and can be accessed anywhere throughout program

local : when variable defined inside a block of code it has a local scope and it only accessible inside that block

enclosed : when have a function or a block inside of another function or block than variables defined inside outer most block are accessible in inner block or functions

23) Built-in methods for strings, lists, etc

strings methods →

len()	: get string length
strip()	: remove leading and trailing space
count()	: count for a substring or character in strings
upper()	: switch to uppercase
lower()	: switch to lowercase
title()	: switch to title case
capitalize()	: capitalize each word
replace()	: replace a substring with another string
index()	: to find index value of a substring or character
find()	: to find index value of a substring or character
split()	: to split string at a separator
join()	: to join a string to each element
isalpha()	: to find if a string made of alphabets
isalnum()	: to find if string contain alphabets and numerics only
isdigit()	: to find if a string contain digits only
startswith()	: find if string starts with specific string
zfill()	: fill the starting of string with zero for a specific character width
endswith()	: find if string ends with specific string
center()	: center a string with specific character width

list methods →

len()	: to get list length
append()	: append an element end of list
insert()	: insert an element at specific index
extend()	: extend a list with another list or tuple
pop()	: pop out last element or a specific index element from list
remove()	: remove a specific element from list
index()	: to find the index of specific element

reverse( ): reverse the list
sort() : sort elements of list
clear() : clear all the element of list
count() : count the occurance of a element in list

24) Understanding the role of break, continue, and pass in Python loops
break continue and pass are control statements used to alter the flow of loop
Break: break statement terminates the loop when a specific condition is true regardless of loop’s condition
Continue: continue statement skip the current iteration of a loop when a certain condition is true and continue over next iteration
Pass: pass uses as a placeholder and do nothing, in a loop when a statement is required we use pass

25) Understanding how to access and manipulate strings.
strings written in double quotations and are immutable in python,
to access we store a string into variable : string = "some text"

to access string character use indexing =
string[0] : s
string[1] : o
or a range--
string[0:4] : some
string += "hello" : string concatenation

string manipulation using methods :

use len(string)	to get length of string
stirng.upper()	to uppercase string
string.lower()	to lowercase string
string.title()	to title case string
string.capitalize()	to capitalize string
string.replace(old,new)	to replace a substring in string
string.split(',')	to split string at a separator
f"string{name}"	to format a string using format method

escape sequence character for text formatting --> \n \b \\ \v
etc..

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

concatenation --> concat a string with another using + operator

```
string = "kapil"
string = "hello " + string
string += " how are you"
```

using f"" string format (f-string)

```
string = f"hello {string} how are you"
```

repetition string --> repeating a string using * operator

```
repeated = string * 3
print("hello " * 3)
```

string method -->

stirng.upper()	to uppercase string
string.lower()	to lowercase string
string.title()	to title case string
string.capitalize()	to capitalize string
string.replace(old,new)	to replace a substring in string
string.split(',')	to split string at a separator
string.center(20,'-')	to center a string on specific length with a symbol or character
string.zfill()	to fill the leading part of string with zeros for specific length

27) String slicing.

string slicing is a method to create a substring out of original string in python be specifying a range and step in [ ] breakets

substring = string [star t: end : step]   step part is optional by default it takes 1 step

```
string = "python"
```

substring = string[0:5]	: specify start and end
substring = string[:5]	: specify only end
substring = string[2:]	: specify only end
substring = string[-5:-2]	: negative indexing

29) Using map(), reduce(), and filter() functions for processing data.
map(), reduce(), and filter() are built in functions that allow to apply a specific function on items of iterables
map(function, iterable)
map apply a function on all the items of a iterable and return a new list

filter(function, iterable)
filter apply a function on all the items of a iterable and and filter out the results based on condition inside function