1. What is Python and why is it popular?

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language that is widely used for web development, scientific computing, data analysis, artificial intelligence, and many other applications. The Python Software Foundation is the organization behind Python. It was created in the late 1980s by Guido van Rossum and was first released in 1991.

## Why Python Programming is so popular:

Python is a simple language and is easy to understand. This is one of the reasons developers like working on it. It is easy for a new developer to learn Python. It is easily readable as compared to other languages. Python has many useful libraries that make the development work easy. Python has a lot of active communities of programmers across the world. People can share their coding problems, and other programmers provide solutions.

Python is one of the most widely used programming languages in Data Science.

Data Science involves the application of various Mathematical and Statistical methods along with Algorithms to find patterns and useful insights in large amounts of structured and unstructured data.

Python has been popular in Machine Learning, and most professionals in this field use Python. Machine Learning (ML) is a branch of AI that works towards making systems perform certain tasks and take decisions without the supervision of human beings. ML is a developing field and has already been used in popular sites like Netflix for movie recommendations. Virtual assistants, chatbots, social media apps, etc use ML.

Python provides strong library support and makes web development easier as compared to other languages.

Python has a lot of tools, packages, and modules to support the automation of applications quickly.

1. What are the differences between Python 2 and Python 3?

Python 3 has changes from Python 2 in below listed features.

Print function:

In python 2 print is a statement so it is written as print <output content>. But in python 3 it is a function, so it is written as print(<output content>) with the parentheses and the output inside the parentheses.

Integer Division:

In python 2, all the numbers given without decimal points are treated as integers, so sometimes the division of two integers in python 2 might lead to unexpected results. For example, if you try to execute 5/4 in python 2, the result will be 1 instead of 1.25. This is because here both 5 and 4 are treated as integers and so the result will not contain any decimal points. This behavior is fixed in python 3

Unicode Strings:

.

By default, python 2 stores strings as ASCII and requires you to mark the string specifically with a “u” if you wish to store them as Unicode. Whereas, python 3 stores strings as Unicode by default. Unicode strings are more versatile compared to ASCII strings as they can store letters from foreign languages as well as emojis and the standard Roman letters and numerals

### Error Handling

In the except clause of try-except block, ‘as’ keyword is added in python 3.x, which was not there previously in python 2. Moreover, the ‘as’ keyword is made mandatory in python 3. If not specified in python 3 it throws an error. Take a look at the below sample code of python 2.x and 3.x to understand it better.

### Xrange

The xrange() function of python 2 is removed in python 3 Instead, the behavior and properties of xrange() are incorporated into the [range() function in python](https://www.scaler.com/topics/range-function-in-python/) 3

### Raising Exception:

The syntax for raising an exception is modified in python 3

Python 3: raise IOError(“error message”)

Python 2: raise IOError, “error message”

### List Comprehension Loop Variables:

In python 2, if we give the name of the variable that is iterated over in for loop list comprehension and also used the same name for a global variable, then the values of the global variable will get updated because of the variable with the same name that is iterated in the for loop list comprehension which should not be the case. This issue is fixed in python 3. The variable used to iterate in the for loop is treated as a local variable inside the for loop and the global variable with the same name won’t be affected.

1. What is the difference between a tuple and a list in Python?

* List is mutable. Tuple is immutable
* List iteration is slower and is time consuming. Tuple iteration is faster.
* List is useful for insertion and deletion operations. Tuple is useful for read only operations like accessing elements.
* List consumes more memory. Tuples consumes less memory.
* List provides many in-built methods. Tuples have less in-built methods.
* List operations are more error prone. Tuples operations are safe.

1. How do you create a dictionary in Python?

Dictionary items are ordered, changeable, and do not allow duplicates.

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

dcEmployee= {

"Name": "Kumararaja Sethuraja",

"Id": "I22060",

"City": Chennai

}

1. What is a function in Python and how do you define one?

* Function blocks begin with the keyword def followed by the function name and parentheses ( ( ) ).
* Any input parameters or arguments should be placed within these parentheses. We can also define parameters inside these parentheses.
* The first statement of a function can be an optional statement - the documentation string of the function or docstring.
* The code block within every function starts with a colon (:) and is indented.
* The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

Syntax:

def functionname( parameters ):

"function\_docstring"

function\_suite

return [expression]

Ex:

def printme( str ):

"This prints a passed string into this function"

print str

Return

1. What is object-oriented programming (OOP) and how does it relate to Python?

Object-oriented programming is a method of structuring a program by bundling related properties and behavior into individual objects. These objects are related to entities. Object-oriented programming language helps us in writing reusable code. It is a popular and widely used method of solving problems by creating objects.

Python is an object oriented programming language. Everything in python is an object. Using python we can create classes and objects, for example, that functions are first-class objects. Functions, classes, strings, and even types are objects in Python: like any object, they have a type, they can be passed as function arguments, and they may have methods and properties. Therefore, Python is an object oriented programming language.

1. How do you handle exceptions in Python?

The cause of an exception is often external to the program itself. For example, an incorrect input, a malfunctioning IO device etc. Because the program abruptly terminates on encountering an exception, it may cause damage to system resources, such as files. Hence, the exceptions should be properly handled so that an abrupt termination of the program is prevented.

Python uses try and except keywords to handle exceptions. Both keywords are followed by indented blocks.

Syntax:

try :

#statements in try block

except :

#executed when error in try block

Ex:

try:

a=5

b='0'

print(a/b)

except:

print('Some error occurred.')

print("Out of try except blocks.")

1. How do you read and write files in Python?

In Python, the [IO](https://docs.python.org/3/library/io.html#module-io) module provides methods of three types of IO operations; raw binary files, buffered binary files, and text files. The canonical way to create a file object is by using the open() function.

Any file operations can be performed in the following three steps:

* Open the file to get the file object using the built-in [open()](https://www.tutorialsteacher.com/python/open-method) function. There are different access modes, which you can specify while opening a file using the [open() function](https://www.tutorialsteacher.com/python/open-method).
* Perform read, write, append operations using the file object retrieved from the open() function.
* Close and dispose of the file object.

Ex:

>>> f = open('C:\myfile.txt') # opening a file

>>> lines = f.read() # reading a file

>>> lines

'This is the first line. \nThis is the second line.\nThis is the third line.'

>>> f.close() # closing file object

1. How do you install and use external packages in Python?

The Python package manager (pip) allows for the installation of modules and packages.

Open a terminal and use the pip command to install a module across the entire system.

PIP is a package manager for Python modules or packages. PIP comes inbuilt in

Python versions 3.4 and beyond.

The newest version of a module and any dependencies will be installed using the command

“pip install hashlib” from the Python Package Index

1. How do you use the "if" statement in Python to perform conditional execution?

In Python, there are three forms of the if...else statement.

* if statement
* if...else statement
* if...elif...else statement

Syntax:

if condition:

# block of code if condition is True

else:

# block of code if condition is False

Ex:

number = 10

if number > 0:

print('Positive number')

else:

print('Negative number')