**Q1. What is the difference between list and tuples in Python?**

LIST TUPLES

Lists are mutable i.e they can be edited. Tuples are immutable (tuples are lists which can’t be edited).

Lists are slower than tuples. Tuples are faster than list.

Syntax: list\_1 = [10, ‘Chelsea’, 20] Syntax: tup\_1 = (10, ‘Chelsea’ , 20)

**Q2. What are the key features of Python?**

Python is an interpreted language. That means that, unlike languages like C and its variants, Python does not need to be compiled before it is run. Other interpreted languages include PHP and Ruby.

Python is dynamically typed, this means that you don’t need to state the types of variables when you declare them or anything like that. You can do things like x=111 and then x="I'm a string" without error

Python is well suited to object orientated programming in that it allows the definition of classes along with composition and inheritance. Python does not have access specifiers (like C++’s public, private).

In Python, functions are first-class objects. This means that they can be assigned to variables, returned from other functions and passed into functions. Classes are also first class objects

Writing Python code is quick but running it is often slower than compiled languages. Fortunately，Python allows the inclusion of C-based extensions so bottlenecks can be optimized away and often are. The numpy package is a good example of this, it’s really quite quick because a lot of the number-crunching it does isn’t actually done by Python

Python finds use in many spheres – web applications, automation, scientific modeling, big data applications and many more. It’s also often used as “glue” code to get other languages and components to play nice.

**Q3. What type of language is python? Programming or scripting?**

Ans: Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language. To know more about Scripting, you can refer to the Python Scripting Tutorial.

**Q4.Python an interpreted language. Explain.**

Ans: An interpreted language is any programming language which is not in machine-level code before runtime. Therefore, Python is an interpreted language.

**Q5.What are the benefits of using Python?**

Ans: The benefits of using python are-

Easy to use– Python is a high-level programming language that is easy to use, read, write and learn.

Interpreted language– Since python is interpreted language, it executes the code line by line and stops if an error occurs in any line.

Dynamically typed– the developer does not assign data types to variables at the time of coding. It automatically gets assigned during execution.

Free and open-source– Python is free to use and distribute. It is open source.

Extensive support for libraries– Python has vast libraries that contain almost any function needed. It also further provides the facility to import other packages using Python Package Manager(pip).

Portable– Python programs can run on any platform without requiring any change.

The data structures used in python are user friendly.

It provides more functionality with less coding.

**Q6.What are decorators in Python?**

Ans: Decorators are used to add some design patterns to a function without changing its structure. Decorators generally are defined before the function they are enhancing. To apply a decorator we first define the decorator function. Then we write the function it is applied to and simply add the decorator function above the function it has to be applied to. For this, we use the @ symbol before the decorator.

**Q7.What are Dict and List comprehensions?**

Ans: Dictionary and list comprehensions are just another concise way to define dictionaries and lists.

Example of list comprehension is-

x=[i for i in range(5)]

The above code creates a list as below-

[0,1,2,3,4]

Example of dictionary comprehension is-

x=[i : i+2 for i in range(5)]

The above code creates a list as below-

[0: 2, 1: 3, 2: 4, 3: 5, 4: 6]

**Q8.What are the common built-in data types in Python?**

Ans: The common built-in data types in python are-

Numbers– They include integers, floating-point numbers, and complex numbers. eg. 1, 7.9,3+4i

List– An ordered sequence of items is called a list. The elements of a list may belong to different data types. Eg. [5,’market’,2.4]

Tuple– It is also an ordered sequence of elements. Unlike lists , tuples are immutable, which means they can’t be changed. Eg. (3,’tool’,1)

String– A sequence of characters is called a string. They are declared within single or double-quotes. Eg. “Sana”, ‘She is going to the market’, etc.

Set– Sets are a collection of unique items that are not in order. Eg. {7,6,8}

Dictionary– A dictionary stores values in key and value pairs where each value can be accessed through its key. The order of items is not important. Eg. {1:’apple’,2:’mango}

Boolean– There are 2 boolean values- True and False.

**Q9.What is slicing in Python?**

Ans: Slicing is used to access parts of sequences like lists, tuples, and strings. The syntax of slicing is-[start:end:step]. The step can be omitted as well. When we write [start:end] this returns all the elements of the sequence from the start (inclusive) till the end-1 element. If the start or end element is negative i, it means the ith element from the end. The step indicates the jump or how many elements have to be skipped. Eg. if there is a list- [1,2,3,4,5,6,7,8]. Then [-1:2:2] will return elements starting from the last element till the third element by printing every second element.i.e. [8,6,4].

**Q10.What are Keywords in Python?**

Ans: Keywords in python are reserved words that have special meaning.They are generally used to define type of variables. Keywords cannot be used for variable or function names. There are following 33 keywords in python-

And, Or, Not, If, Elif, Else, For, While, Break, As, Def, Lambda, Pass, Return,

True, False, Try, With, Assert, Class, Continue, Del, Except, Finally, From, Global

Import, In, Is, None, Nonlocal, Raise, Yield

**Q11.What are Literals in Python and explain about different Literals**

Ans: A literal in python source code represents a fixed value for primitive data types. There are 5 types of literals in python-

String literals– A string literal is created by assigning some text enclosed in single or double quotes to a variable. To create multiline literals, assign the multiline text enclosed in triple quotes. Eg.name=”Tanya”

A character literal– It is created by assigning a single character enclosed in double quotes. Eg. a=’t’

Numeric literals include numeric values that can be either integer, floating point value, or a complex number. Eg. a=50

Boolean literals– These can be 2 values- either True or False.

Literal Collections– These are of 4 types-

a) List collections-Eg. a=[1,2,3,’Amit’]

b) Tuple literals- Eg. a=(5,6,7,8)

c) Dictionary literals- Eg. dict={1: ’apple’, 2: ’mango, 3: ’banana`’}

d) Set literals- Eg. {“Tanya”, “Rohit”, “Mohan”}

6. Special literal- Python has 1 special literal None which is used to return a null variable.

**Q12.What are the new features added in Python 3.9.0.0 version?**

Ans: The new features in Python 3.9.0.0 version are-

New Dictionary functions Merge(|) and Update(|=)

New String Methods to Remove Prefixes and Suffixes

Type Hinting Generics in Standard Collections

New Parser based on PEG rather than LL1

New modules like zoneinfo and graphlib

Improved Modules like ast, asyncio, etc.

Optimizations such as optimized idiom for assignment, signal handling, optimized python built ins, etc.

Deprecated functions and commands such as deprecated parser and symbol modules, deprecated functions, etc.

Removal of erroneous methods, functions, etc.

**Q13. How is memory managed in Python?**

Ans: Memory is managed in Python in the following ways:

Memory management in python is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have access to this private heap. The python interpreter takes care of this instead.

The allocation of heap space for Python objects is done by Python’s memory manager. The core API gives access to some tools for the programmer to code.

Python also has an inbuilt garbage collector, which recycles all the unused memory and so that it can be made available to the heap space.

**Q14. What is namespace in Python?**

Ans: A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.

**Q15.What are local variables and global variables in Python?**

Global Variables:

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

Local Variables:

Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.

a=2

def add():

b=3

c=a+b

print(c)

add()

Output: 5

When you try to access the local variable outside the function add(), it will throw an error.

**Q16. Is python case sensitive?**

Ans: Yes. Python is a case sensitive language.

**Q17.What is type conversion in Python?**

Ans: Type conversion refers to the conversion of one data type into another.

int() – converts any data type into integer type

float() – converts any data type into float type

ord() – converts characters into integer

hex() – converts integers to hexadecimal

oct() – converts integer to octal

tuple() – This function is used to convert to a tuple.

set() – This function returns the type after converting to set.

list() – This function is used to convert any data type to a list type.

dict() – This function is used to convert a tuple of order (key, value) into a dictionary.

str() – Used to convert integer into a string.

complex(real,imag) – This function converts real numbers to complex(real,imag) number.

**Q18. What are functions in Python?**

Ans: A function is a block of code which is executed only when it is called. To define a Python function, the def keyword is used.

def Newfunc():

print("Hi, Welcome to Edyoda")

Newfunc(); #calling the function

Output: Hi, Welcome to Edyoda

**Q19.What is a lambda function?**

Ans: An anonymous function is known as a lambda function. This function can have any number of parameters but, can have just one statement.

a = lambda x,y : x+y

print(a(5, 6))

**Q20. What does this mean: \*args, \*\*kwargs? And why would we use it?**

**Ans:** We use \*args when we aren’t sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. \*\*kwargs is used when we don’t know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments. The identifiers args and kwargs are a convention, you could also use \*bob and \*\*billy but that would not be wise.