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

|  |  |
| --- | --- |
| **LIST vs TUPLES** | |
| **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**](https://www.edureka.co/blog/python-class/) 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](https://www.edureka.co/blog/python-numpy-tutorial/" \t "_blank) 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. Learn more about Big Data and its applications from the[Data Engineering Training](https://www.edureka.co/microsoft-azure-data-engineering-certification-course).

### ****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](https://youtu.be/9F6zAuYtuFw).

### ****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 is pep 8?****

***Ans:***PEP stands for **Python Enhancement Proposal.**It is a set of rules that specify how to format Python code for maximum readability.

**Q6.What are the benefits of using Python?**

***Ans:***The benefits of using python are-

* 1. **Easy to use**– Python is a high-level programming language that is easy to use, read, write and learn.
  2. **Interpreted language**– Since python is interpreted language, it executes the code line by line and stops if an error occurs in any line.
  3. **Dynamically typed**– the developer does not assign data types to variables at the time of coding. It automatically gets assigned during execution.
  4. **Free and open-source**– Python is free to use and distribute. It is open source.
  5. **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).
  6. **Portable**– Python programs can run on any platform without requiring any change.
  7. The data structures used in python are user friendly.
  8. It provides more functionality with less coding.

**Find out our Python Training in Top Cities/Countries**

|  |  |  |
| --- | --- | --- |
| **India** | **USA** | **Other Cities/Countries** |
| [Bangalore](https://www.edureka.co/python-programming-certification-training-bangalore) | [New York](https://www.edureka.co/python-programming-certification-training-new-york-city) | [UK](https://www.edureka.co/python-programming-certification-training-uk) |
| [Hyderabad](https://www.edureka.co/python-programming-certification-training-hyderabad) | [Chicago](https://www.edureka.co/python-programming-certification-training-chicago) | [London](https://www.edureka.co/python-programming-certification-training-london) |
| [Delhi](https://www.edureka.co/python-programming-certification-training-delhi) | [Atlanta](https://www.edureka.co/python-programming-certification-training-atlanta) | [Canada](https://www.edureka.co/python-programming-certification-training-canada) |
| [Chennai](https://www.edureka.co/python-programming-certification-training-chennai) | [Houston](https://www.edureka.co/python-programming-certification-training-houston) | [Toronto](https://www.edureka.co/python-programming-certification-training-toronto) |
| [Mumbai](https://www.edureka.co/python-programming-certification-training-mumbai) | [Los Angeles](https://www.edureka.co/python-programming-certification-training-los-angeles) | [Australia](https://www.edureka.co/python-programming-certification-training-australia) |
| [Pune](https://www.edureka.co/python-programming-certification-training-pune) | [Boston](https://www.edureka.co/python-programming-certification-training-boston) | [UAE](https://www.edureka.co/python-programming-certification-training-uae) |
| [Kolkata](https://www.edureka.co/python-programming-certification-training-kolkata) | [Miami](https://www.edureka.co/python-programming-certification-training-miami) | [Dubai](https://www.edureka.co/python-programming-certification-training-dubai) |
| [Ahmedabad](https://www.edureka.co/python-programming-certification-training-ahmedabad) | [San Francisco](https://www.edureka.co/python-programming-certification-training-san-francisco) | [Philippines](https://www.edureka.co/python-programming-certification-training-philippines) |

**Q7.What are Python namespaces?**

**Ans:**A namespace in python refers to the name which is assigned to each object in python. The objects are variables and functions. As each object is created, its name along with space(the address of the outer function in which the object is), gets created. The namespaces are maintained in python like a dictionary where the key is the namespace and value is the address of the object. There 4 types of namespace in python-

1. **Built-in namespace**– These namespaces contain all the built-in objects in python and are available whenever python is running.
2. **Global namespace**– These are namespaces for all the objects created at the level of the main program.
3. **Enclosing namespaces**– These namespaces are at the higher level or outer function.
4. **Local namespaces**– These namespaces are at the local or inner function.

**Q8.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.

**Q9.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-

|  |  |
| --- | --- |
| 1 | x=[i for i in range(5)] |

The above code creates a list as below-

|  |  |
| --- | --- |
| 1  2 | 4  [0,1,2,3,4] |

Example of dictionary comprehension is-

|  |  |
| --- | --- |
| 1 | x=[i : i+2 for i in range(5)] |

The above code creates a list as below-

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

**Q10.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**.

**Q11.What is the difference between .py and .pyc files?**

***Ans:*** The .py files are the python source code files. While the .pyc files contain the bytecode of the python files. .pyc files are created when the code is imported from some other source. The interpreter converts the source .py files to .pyc files which helps by saving time. You can get a better understanding with the [Data Engineering Course in Washington](https://www.edureka.co/microsoft-azure-data-engineering-certification-course-washington).

**Q12.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].

**Q13.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

**Q14.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-

1. **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”
2. **A character literal**– It is created by assigning a single character enclosed in double quotes. Eg. a=’t’
3. **Numeric literals** include numeric values that can be either integer, floating point value, or a complex number. Eg. a=50
4. **Boolean literals**– These can be 2 values- either True or False.
5. **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.

**Q15.How to combine dataframes in pandas?**

***Ans:*** The dataframes in python can be combined in the following ways-

1. Concatenating them by stacking the 2 dataframes vertically.
2. Concatenating them by stacking the 2 dataframes horizontally.
3. Combining them on a common column. This is referred to as joining.

The concat() function is used to concatenate two dataframes. Its syntax is- pd.concat([dataframe1, dataframe2]).

Dataframes are joined together on a common column called a key. When we combine all the rows in dataframe it is union and the join used is outer join. While, when we combine the common rows or intersection, the join used is the inner join. Its syntax is- pd.concat([dataframe1, dataframe2], axis=’axis’, join=’type\_of\_join)

**Q16.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.

### ****Q17. How is memory managed in Python?****

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

1. 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.
2. 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.
3. 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.

### ****Q18. What is namespace in Python?****

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

### ****Q19. What is PYTHONPATH?****

***Ans:***It is an environment variable which is used when a module is imported. Whenever a module is imported, PYTHONPATH is also looked up to check for the presence of the imported modules in various directories. The interpreter uses it to determine which module to load.

### ****Q20. What are python modules? Name some commonly used built-in modules in Python?****

***Ans:***Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code.

Some of the commonly used built-in modules are:

* os
* sys
* math
* random
* data time
* JSON