PYTHON

Python is a high-level, interpreted, and versatile programming language known for its simplicity and readability. It was created by **Guido van Rossum** in 1991 and is widely used in web development, data analysis, artificial intelligence, machine learning, automation, and more.

Key Features of Python:

1. **Simple and Easy to Learn**: Its syntax is clean and mirrors natural language, making it beginner-friendly.
2. **Interpreted Language**: Python executes code line by line, allowing for easy debugging and interactive coding.
3. **Cross-Platform**: It works seamlessly on various operating systems like Windows, macOS, and Linux.
4. **Extensive Libraries**: Python has a rich set of libraries like NumPy, Pandas, TensorFlow, etc., for various tasks.
5. **Dynamic Typing**: You don't need to declare variable types explicitly.
6. **Community Support**: Python has a vast and active community, ensuring quick help and robust resources.

Applications of Python:

1. **Web Development**: Frameworks like Django and Flask.
2. **Data Science**: Libraries like Pandas and Matplotlib.
3. **Machine Learning and AI**: Using TensorFlow, Keras, etc.
4. **Game Development**: Libraries like Pygame.
5. **Automation**: Scripting repetitive tasks.
6. **IoT Projects**: Working with microcontrollers using MicroPython.

Criteria	List	Tuple	Set
Mutability	Mutable: Elements can be modified after creation	Immutable: Elements cannot be modified after creation	Mutable: Elements can be modified after creation
Order	Ordered: Elements have a specific order	Ordered: Elements have a specific order	Unordered: Elements have no specific order
Duplicates	Duplicates allowed	Duplicates allowed	Duplicates not allowed
Indexing	Access elements by index	Access elements by index	No indexing

List Vs Set Vs Dictionary Vs Tuple

Lists	Sets	Dictionaries	Tuples
List = [10, 12, 15]	Set = {1, 23, 34} Print(set) -> {1, 23,24} Set = {1, 1} print(set) -> {1}	Dict = {"Ram": 26, "mary": 24}	Words = ("spam", "egss") Or Words = "spam", "eggs"
Access: print(list[0])	Print(set). Set elements can't be indexed.	print(dict["ram"])	Print(words[0])
Can contains duplicate elements	Can't contain duplicate elements. Faster compared to Lists	Can't contain duplicate keys, but can contain duplicate values	Can contains duplicate elements. Faster compared to Lists
List[0] = 100	set.add(7)	Dict["Ram"] = 27	Words[0] = "care" ->TypeError
Mutable	Mutable	Mutable	Immutable - Values can't be changed once assigned
List = []	Set = set()	Dict = {}	Words = ()
Slicing can be done print(list[1:2]) -> [12]	Slicing: Not done.	Slicing: Not done	Slicing can also be done on tuples
Usage: Use lists if you have a collection of data that doesn't need random access. Use lists when you need a simple, iterable collection that is modified frequently.	Usage: - Membership testing and the elimination of duplicate entries. - when you need uniqueness for the elements.	Usage: - When you need a logical association b/w key:value pair. - when you need fast lookup for your data, based on a custom key. - when your data is being constantly modified.	Usage: Use tuples when your data cannot change. A tuple is used in comibnation with a dictionary, for example, a tuple might represent a key, because its immutable.

Python offers numerous advantages that make it a highly preferred programming language. Its simplicity and readability, along with an intuitive syntax, make it easy to learn and use, even for beginners. Python supports multiple programming paradigms, including object-oriented, procedural, and functional programming, making it versatile for various applications such as web development, data analysis, artificial intelligence, and automation.

Its extensive standard library and third-party libraries simplify complex tasks, eliminating the need to write code from scratch. Being cross-platform, Python ensures that code written on one operating system can run seamlessly on another. The dynamic typing feature allows developers to write flexible and concise code without specifying variable types.

Furthermore, Python integrates well with other programming languages like C, Java, and .NET, and its strong community support ensures access to comprehensive documentation, tutorials, and quick

problem resolution. These features, combined with Python's suitability for rapid prototyping and automation, make it a powerful and efficient tool for developers worldwide.