

WELCOME

P Y T H O N
P R E S E N T A T I O N

BCA IBM

BY

ANSH SHUKLA

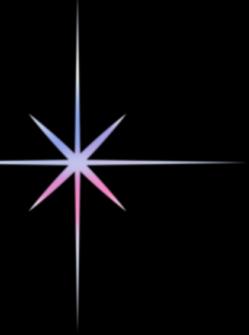
ALFAIZ

MO SHAHNAWAJ

MOHD ADIL

AASHISH YADAV

CONTAINER DATA TYPE



TYPES

LIST

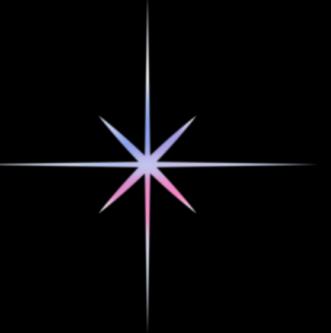
TUPLE

SET

DICTIONARY



LIST



DEFINITION

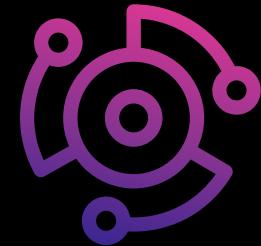
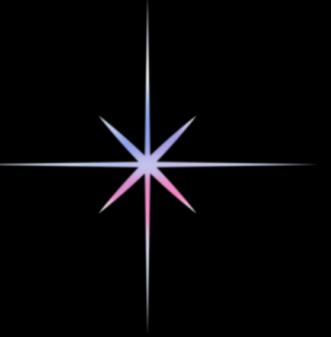
In Python, a list is a built-in data type that is used to store an ordered collection of items. Lists are mutable, meaning they can be changed after their creation, and they can contain elements of different data types, such as integers, strings, or even other lists. Lists are defined by placing a sequence of elements within square brackets, separated by commas, like this: `my_list = [1, 2, 3, 'apple', 'banana']`.

SYNTAX

```
my_list = [12, 20 ,30, “gaddafi”]
```



TUPLE



DEFINATION

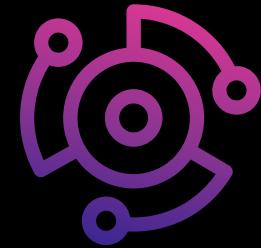
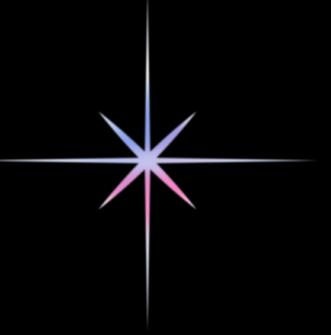
A tuple in Python is a collection of ordered elements that is immutable, meaning its contents cannot be changed after it is created. Tuples are defined by enclosing the elements in parentheses, separated by commas. They are useful for grouping related data and can contain elements of different data types.

SYNTAX

```
my_tuple = (5, 20 ,30)
```



SET



DEFINATION

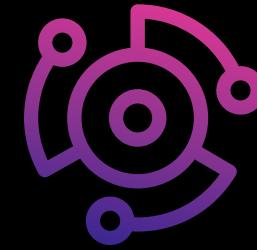
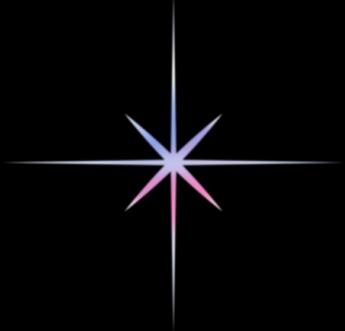
In Python, a set is a collection of unordered and unique elements. Sets are defined by enclosing the elements in curly braces, separated by commas. They are useful for storing distinct items and support operations like union, intersection, and difference. Sets are mutable, allowing you to add or remove elements after their creation.

SYNTAX

```
my_set = {5, 20 ,30,40}
```



DICTIONARY



DEFINATION

In Python, a dictionary is a collection of key-value pairs, where each key is unique. Dictionaries are defined by enclosing the key-value pairs in curly braces, with each pair separated by a comma and each key and value separated by a colon. They are useful for storing data that requires a label or identifier for each item, and they support operations like adding, updating, and removing key-value pairs. Dictionaries are mutable, allowing you to modify their contents after creation.

SYNTAX

```
my_dict = {  
    "name": "gaddafi",  
    "age": 40,  
    "course": "BCA ibm"  
}
```



THANK YOU!

