What is list ?

A list is an ordered, mutable (changeable) collection that allows duplicates. It is defined using square brackets [ ].

It maintain an intersection order

It allows the duplicates valves

It can be modified

It is used to store multiple data types

SYNTAX:

```
list1_districts= ["hyd","nzb","kcg","sec","wgl","nrml","kmr"]

list2_fruits=["apple","mango","orange","pineapple","grapes"]

list3_INT=[10,20,30,40,50]

print(f"the list of the districts is :{list1_districts}")

print(f"the list of the fruits is :{list2_fruits}")

print(f"the list of the INTEGERS is :{list3_INT}")
```

OUTPUT

the list of the districts is :['hyd', 'nzb', 'kcg', 'sec', 'wgl', 'nrml', 'kmr']

the list of the fruits is :['apple', 'mango', 'orange', 'pineapple', 'grapes']

the list of the INTEGERS is :[10, 20, 30, 40, 50]

What is tuple ?

A tuple is an ordered, immutable (unchangeable) collection that allows duplicates. It is defined using parentheses ( ).

It maintain an intersection order

It allows the duplicate values

It is immutable

The elements can be accepted by the index

It is ordered collection of the data /elements

SYNTAX

```
# tuple

numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9,)

colours= ("red", "blue", "green","yellow","orange","green")

print(f"the list of the numbers :{numbers},the list of the colours is:{colours}")
```

OUTPUT

the list of the numbers :(1, 2, 3, 4, 5, 6, 7, 8, 9),the list of the colours is:('red', 'blue', 'green', 'yellow', 'orange', 'green')

what is sets?

The set is an unordered collection of data /elements .it is defined using the curly brackets

{ }

It doesn't follow any specific order

It doesn’t allow duplicates

It mutable we can add or remove the data/elements

The set is immutable when it is frozen set.

SYNTAX

```
# SETS

set_of_numbers= {1, 2, 3, 4, 5}

set_of_characters = {"a", "b", "c", "d"}

set_of_car_companies={"maruthisuzuki","toyota","kia","hundai","honda","citron","M
G","hundai","maruthisuzuki"}



print(f"the set of numbers:{set_of_numbers}, the set of
charactors:{set_of_characters}, the set of car compinies
:{set_of_car_companies}")
```

OUTPUT

the set of numbers:{1, 2, 3, 4, 5}, the set of charactors:{'c', 'd', 'a', 'b'}, the set of car compinies :{'maruthisuzuki', 'MG', 'kia', 'honda', 'hundai', 'citron', 'toyota'}

What is dictionary?

A dictionary is an unordered, mutable collection of key-value pairs. It is defined using curly brackets {} with keys and values.

It stores the key value pairs

The keys must be unique

It is mutable it can be modified

SYNTAX

```
#dictonary

personal_details = {"name": "Alice", "age": 25, "city": "New York",
"work":"software engineer","salary":250000}

car_details= {"brand": "Toyota", "model": "Camry", "year": 2022,"colour":"mat
black"}



print(f"employee personal detailes :{personal_details}")

print(f"full details of car :{car_details}")

print(car_details["colour"])
```

OUTPUT

employee personal detailes :{'name': 'Alice', 'age': 25, 'city': 'New York', 'work': 'software engineer', 'salary': 250000}

full details of car :{'brand': 'Toyota', 'model': 'Camry', 'year': 2022, 'colour': 'mat black'}

mat black

THE DIFFERENCES BETWEEN THE LIST AND TUPLE

## Tuple vs List (Differences) in Python

1. Mutability
    - A tuple is immutable, meaning its elements cannot be changed after creation.
    - A list is mutable, meaning its elements can be modified (add, remove, update).
2. Syntax
    - A tuple is created using parentheses ( ), e.g., tuple1 = (1, 2, 3).
    - A list is created using square brackets [ ], e.g., list1 = [1, 2, 3].
3. Order
    - Both tuples and lists maintain the insertion order of elements.
4. Duplicates
    - Both tuples and lists allow duplicate values.

5. Use Cases

- Use tuples when data should not change (e.g., storing coordinates, database records, or fixed configurations).
- Use lists when data needs modification (e.g., to-do lists, shopping carts, or dynamic collections).

The differences between the list and sets

## ist vs Set (Differences) in Python

1. Mutability
    - Both lists and sets are mutable, meaning elements can be added or removed.
2. Order
    - A list maintains the insertion order of elements.
    - A set is unordered, meaning the order of elements is not guaranteed.
3. Duplicates
    - A list allows duplicate values.
    - A set does not allow duplicate values—only unique elements are stored.

4. Use Cases

- Use lists when order matters and duplicates are allowed (e.g., maintaining a sequence of elements).
- Use sets when uniqueness is required and fast searching is needed (e.g., storing unique usernames).

CONVERTING TUPLE TO LIST

SYNTAX

```
# CONVERTING TUPLE TO LIST

tuple = (100, 200, 300, 400)

list = list(tuple)



print(list
```

Output

[100, 200, 300, 400]

Updating the list

```
#modifiying the the list and adding the element

list.append(600)

list[2] = 150 , 250



print(list)
```

Output

[100, 200, (150, 250), 400, 600]

CONVERTING BACK TO TUPLE

```
#CONVERTING BACK TO TUple

list3= [1,2,3,4,5,6,7]

tuple4 = tuple(list3)



print(tuple4)
```

Output

(1, 2, 3, 4, 5, 6, 7)

Delete the Created Tuple

```
del tuple4
```

Output

Traceback (most recent call last):

  File "C:\Users\saini\PycharmProjects\helloword\tuple.py", line 9, in <module>

    print(tuple4)

          ^^^^^^

NameError: name 'tuple4' is not defined. Did you mean: 'tuple'?