

TUPLES

⑨ Tuples

- Tuples are used to store multiple items in a single variable.
- Tuples are used to store collection of data.
- Tuples are collection which is ordered and unchangeable. (Immutable).
- Tuples are written with round brackets.
- Tuple uses parenthesis, whereas list use square brackets.
- Sequence of values store in a tuples can be any type, and indexed by integer.
- Tuples are created by placing sequence of values separated by commas.
- Note: Creation of Python without use of the parenthesis is known as Tuple packing.

① Create Tuples

```
thisTuple = ("Apple", "Banana", "cherry")  
print(thisTuple)
```

(i) tuple Items:

- tuple items are ordered, unchangeable and allow duplicate values.
- tuple item are indexed, first item has indexed [0], second [1], etc.

(ii) ordered:

- tuples are ordered, it means that the items have a defined order, and that order will not change.

(iii) Unchangeable:

- tuples are unchangeable, meaning we cannot change, add or remove items after tuple has been created.

(iv) Allow Duplicates

- tuple are indexed, tuples can have items with the same value.
thisTuple = ("Apple", "Banana", "cherry", "Apple")

(v) tuple length:

- to determine how many items a tuple has, use len() function.

`print(len(thislist))`,

(vi) creating tuple with one item:

- To create a tuple with one item, you have to add comma after the item.
- otherwise python will not recognize it as tuple.

`thistuple = ("Apple",) ← tuple`

`thistuple = ("Apple") ← Not tuple.`

`print(type(thistuple))`.

(vii) Tuple Item - Data Types

- tuple item can be any data types (String, int, Boolean).

`tuple1 = ("Apple", "Banana", "cherry")`

`tuple2 = (1, 5, 7, 9, 3)`

`tuple3 = (True, false, false)`.

- tuple contain different data types.

`tuple1 = ("ABC", 34, True, 40, "male")`.

(viii) Tuple constructor.

- It is also possible to use the `tuple()` constructor to make a tuple.
- using `tuple()` method, to make tuple.

`thistuple = tuple(("Apple", "Banana", "cherry"))`

`print(thistuple)`.

① Access Tuple Items:

- you can access tuple items by referring to the index number, inside square bracket

`thistuple = ("Apple", "Banana", "cherry")`

`print(tuple[1])`.

(i) Negative Indexing:

- -1 refers to the last item, -2 for second last item etc.

`print(thistuple[-1])`

(i) Range of Indexes:

- You can specify a range of indexes by specifying where to start and where to end the range.
- Return value will be a new tuple with the specified items.

```
print(thistuple[2:5])
```

Note: search start at (2) and end at index (5).

- by leaving out the start value, range will start at the first item.
`thistuple = ("A", "B", "C", "D", "E", "F", "G")`
`print(thistuple[:4])`

- by leaving out the end value, range will go at the end of list.

```
print(thistuple[2:])
```

(ii) Range of Negative Indexes:

- specify Negative Indexes, if you want to start the search from tuples at the end of the tuples.

```
print(thistuple[-4:-1])
```

② Update Tuple:

- tuple are unchangeable (immutable), means that you cannot change add, or remove items once the tuple is created.
But there are some workarounds.

(i) change tuple values:

- once tuple is created, you cannot change its value. Tuples are unchangeable as it also called.
- convert tuple into a list, change the list and convert the list back into a tuple

Example:

```
x = ("Apple", "Banana", "cherry")
```

```
y = list(x)
```

```
y[1] = 'kiwi'
```

```
x = tuple(y)
```

```
print(x)
```

- once tuple is created, you cannot add item to it.
- Tuples are unchangeable, so you cannot remove item from it, but you can some workaround,

```
x = ("Apple", "Banana", "cherry")
```

```
y = list(thistuple)
```

```
y.remove("Apple")
```

```
thistuple = tuple(y)
```

- or you can delete the tuple completely.
- The del keyword delete tuple completely.

```
del thistuple
```

```
print(thistuple)
```

(3) Unpack Tuples:

(i) packing tuples:

- When we create tuple, we assign normally value to it called as packing.

```
Fruits = ("Apple", "Banana", "cherry")
```

(ii) Unpacking tuples:

- We also allow to extract the values into variables called as unpacking.

```
Fruits = ("Apple", "Banana", "cherry")
```

```
(green, yellow, red) = Fruits
```

```
print(green)
```

```
print(yellow)
```

```
print(red)
```

④ Join - Tuples :

- To join two or more tuples you can use the + operators.

- tuple1 = ("a", "b", "c")

tuple2 = (1, 2, 3)

tuple3 = tuple1 + tuple2.

print(tuple3).

(ii) Multiply Tuples :

- If you want to multiply the content of a tuple a given number of times

- use * operator.

fruits = ("apple", "Banana", "cherry")

mytuple = fruits * 2.

print(mytuple).

tuple Methods :

method	description.
(i) count()	Return the numbers of times a specified value occurs in tuple.
(ii) index()	searches the tuples for a specified value and returns the position of where it was found.