PYTHON TUTORIAL FOR BEGINNERS

Source: www.youtube.com/@RishabhMishraOfficial

Chapter - 17

Tuple in Python

- What is Tuple
- Create Tuples
- Access Tuples: Indexing & Slicing
- Tuple Operations
- Tuple Iteration
- Tuple Methods
- Tuple Functions
- Unpack Tuples
- Modify Tuple

Tuple in Python

A **tuple** is a collection of items in Python that is **ordered**, **unchangeable** (**immutable**) and allow **duplicate** values.

Tuples are used to store multiple items in a single variable.

Note: Ordered – Tuple items have a defined order, but that order will not change.

Example:

```
fruits = ("apple", "orange", "cherry", "apple")
print(fruits)
# Output: ('apple', 'orange', 'cherry', 'apple')
```

Create Tuple in Python

There are several ways to create a **tuple** in Python:

```
1. Using Parentheses ()
colors = ("red", "green", "blue")
numbers = (1, 2, 3, 4, 5)
mixed = (1, "hello", 3.14, True)
nested = (1, [2, 3], (4, 5, 6))
```

```
2. Without Parentheses (Comma-Separated)
also_numbers = 1, 2, 3, 4, 5

3. Using the tuple() Constructor
new_tuple = tuple(("apple", "banana", "cherry"))# use doble
brackets
list_items = ["x", "y", "z"]    # Creating a tuple from a list
tuple_items = tuple(list_items)  # ('x', 'y', 'z')

4. Single-Item Tuple
tuplesingle = ("only",)
```

Accessing Tuple Elements - Indexing

You can **access elements** in a tuple by referring to their **index**. Python uses **zero-based** indexing, meaning the first element has an index of **0**.

Syntax: `tuple_name[index]`

Example:

```
fruits = ("apple", "orange", "cherry", "apple", "mango")
```

Index	0	1	2	3	4
	-5	-4	-3	-2	-1

```
# Access first element
print(fruits[0])   # Output: apple

# Access third element
print(fruits[2])   # Output: cherry

# Access last element using negative index
print(fruits[-1])  # Output: mango
```

Tuple Slicing

Slicing allows you to **access a range of elements** in a tuple. You can specify the **start and stop indices**, and Python returns a new tuple containing the specified elements.

Syntax: `tuple_name[start:stop:step]`

Example:

```
numbers = (10, 20, 30, 40, 50, 60)
```

Index	0	1	2	3	4	5
	-6	-5	-4	-3	-2	-1

```
# Slice from index 1 to 3
print(numbers[1:4])  # Output: (20, 30, 40)

# Slice from start to index 2
print(numbers[:3])  # Output: (10, 20, 30)

# Slice all alternate elements
print(numbers[0::2])  # Output: (10, 30, 50)

# Slice with negative indices
print(numbers[-4:-1])  # Output: (30, 40, 50)

# Reverse list
print(numbers[::-1])  # Output: (60,50,40,30,20,10)
```

Tuple Operations

1. Concatenation

```
# You can join two or more tuples using the + operator.
tuple1 = (1, 2, 3)
tuple2 = (4, 5)
combined = tuple1 + tuple2
print(combined)  # Output: (1, 2, 3, 4, 5)
```

2. Repetition

```
# You can repeat a tuple multiple times using the * operator.
tuple3 = ("hello",) * 3
print(tuple3)  # Output: ('hello', 'hello', 'hello')
```

3. Checking for an Item

```
# Use the in keyword to check if an item exists in a tuple.
numbers = (10, 20, 30, 40)
print(20 in numbers)  # Output: True
```

Iterating Over Tuple

Iterating allows you to traverse each element in a tuple, using loops.

```
#Example: fruits = ("apple", "mango", "cherry")

# Using for loop
for fruit in fruits:
    print(fruit)

# Using while loop

i = 0
while i < len(fruits):
    print(fruits[i])
    index += 1
```

Tuple Methods

Python provides two **built-in methods** to use on tuples.

Methods	Description
count()	Returns the number of times a specified value occurs in a tuple.
index()	Searches the tuple for a specified value and returns the position of where it was found.

```
# count
colors = ("red", "green", "blue", "green")
print(colors.count("green"))  # Output: 2

# index
colors = ("red", "green", "blue", "green")
print(colors.index("blue"))  # Output: 2
```

Tuple Functions

Python provides several **built-in functions** to use on tuples.

Methods	Description
len()	Returns the number of items in a tuple.
sorted()	Returns a new sorted **list** from the items in the tuple
sum()	Sums up all the numeric items in the tuple.
min(), max()	Return the smallest and largest items in the tuple, respectively.

```
numbers = (2, 3, 1, 4)

print(len(numbers))  # Output: 4

sorted_num = sorted(numbers)
print(sorted_num)  # Output: [1,2,3,4]

print(sum(numbers))  # Output: 10
print(min(numbers))  # Output: 1
print(max(numbers))  # Output: 4
```

Packing and Unpacking Tuples

a. **Packing** is the process of putting multiple values into a **single** tuple.

```
a = "Madhav"
b = 21
c = "Engineer"
pack_tuple = a,b,c   # Packing values into a tuple
print(pack_tuple)
```

b. **Unpacking** is extracting the values from a tuple into **separate** variables.

```
name, age, profession = person  # Unpacking a tuple
print(name)        # Output: Madhav
print(age)         # Output: 21
print(profession)  # Output: Engineer
```

Modifying Tuple - Immutable

Once a tuple is created, you **cannot** modify its elements. This means you **cannot** add, remove, or change items.

```
# Creating a tuple
numbers = (1, 2, 3)

# Attempting to change an item
numbers[0] = 10
# This will raise an error
```

Modifying Tuple

But there is a **trick**. You can **convert the tuple into a list, change the list, and convert the list back into a tuple.**

```
my_tuple = ("apple", "mango", "cherry")

# type cast tuple to list
y = list(my_tuple)
y.append("orange")

# type cast back list to tuple
my_tuple = tuple(y)
print(my_tuple)
```

Tuple Use Case - Examples

Storing Fixed Data (Immutable Data)

Example: Storing geographic coordinates (latitude, longitude) or RGB color values, where the values shouldn't be changed after assignment.

```
coordinates = (40.7128, -74.0060)  # Latitude and longitude for NYC
rgb_color = (255, 0, 0)  # RGB value for red
```

Using Tuples as Keys in Dictionaries

Since tuples are immutable and hashable, they can be used as keys in dictionaries, unlike lists.

```
location_data = {
    (40.7128, -74.0060): "New York City",
    (34.0522, -118.2437): "Los Angeles"
   }

print(location_data[(40.7128, -74.0060)]) # Output: New York City
```

Python Tutorial Playlist: **Click Here**

https://www.youtube.com/playlist?list=PLdOKnrf8EcP384Ilxra4UlK9BDJGwawg9