



TUPLE METHODS

Python has only TWO built-in methods for tuples: (Because they are immutable) ☐

1. count()

Returns the number of times a specified value occurs in a tuple.

```
nums = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = nums.count(5)
print(x)
```

Output:
2

2. index()

Searches the tuple for a specified value and returns the position of where it was found.

```
nums = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = nums.index(8)
print(x)
```

Output:
3

Common Mistake! ☐

Trying to use `.append()` or `.remove()` on a tuple will give an `AttributeError`. Remember, Tuples are IMMUTABLE!

MINI EXERCISE

Q: Create a tuple with 5 numbers and find the sum of all elements.

```
t = (10, 20, 30, 40, 50)
print(sum(t))
```

Output:
150

Quick Task!

Try to convert a tuple to a list, change an item, and convert it back to a tuple. It's a common workaround!



UNIT 7: TUPLES IN PYTHON

Hey students! Today we are starting one of the most interesting data types in Python - TUPLES! ☐

What is a Tuple? ☐

A Tuple is a collection which is ordered and IMMUTABLE (cannot be changed). It is similar to a list, but the main difference is that we use round brackets () instead of square brackets [].

Key Features:

- Ordered: Items have a defined order.
- Immutable: Unchangeable after creation.
- Allows Duplicates: Same value can appear multiple times.

Pro Tip!

Once a tuple is created, you cannot add, remove, or change its items. It's like a 'read-only' list!

Example of Creating a Tuple:

```
my_tuple = ('Python', 'Java', 'C++')  
print(my_tuple)
```

Output:
('Python', 'Java', 'C++')

What is the difference between a List and a Tuple?

Interview Q:



TUPLE OPERATIONS

Since tuples are immutable, we can't do everything we do with lists, but we can still perform many operations! □

1. Accessing Items

We use index numbers (starting from 0) to access items.

```
fruits = ('apple', 'banana', 'cherry')  
print(fruits[1]) # Positive indexing  
print(fruits[-1]) # Negative indexing
```

Output:

banana

cherry

2. Slicing

Extract a range of items using [start:stop].

```
nums = (10, 20, 30, 40, 50)  
print(nums[1:4])
```

Output:

(20, 30, 40)

3. Joining Tuples

Use the + operator to join two or more tuples.

```
t1 = (1, 2)  
t2 = (3, 4)  
t3 = t1 + t2  
print(t3)
```

Output:

(1, 2, 3, 4)

Real-world:

Tuples are used for data that shouldn't change, like GPS coordinates (lat, long) or RGB color codes.



PACKING & UNPACKING

This is a super cool feature in Python! ☐

Tuple Packing:

When we create a tuple, we normally assign values to it. This is called 'packing'.

```
fruits = ('apple', 'banana', 'cherry') # Packing
```

Output:
Done!

Tuple Unpacking:

We can extract the values back into variables. This is 'unpacking'.

```
(green, yellow, red) = fruits  
print(green)  
print(yellow)  
print(red)
```

Output:
apple
banana
cherry

Using Asterisk (*):

If the number of variables is less than the number of values, use * to collect remaining values into a list.

```
colors = ('red', 'blue', 'green', 'yellow')  
(r, b, *others) = colors  
print(others)
```

Output:
['green', 'yellow']

How do you swap two variables using a tuple in one line?

Interview Q:

Answer: a, b = b, a (Python internally uses tuple packing/unpacking!)