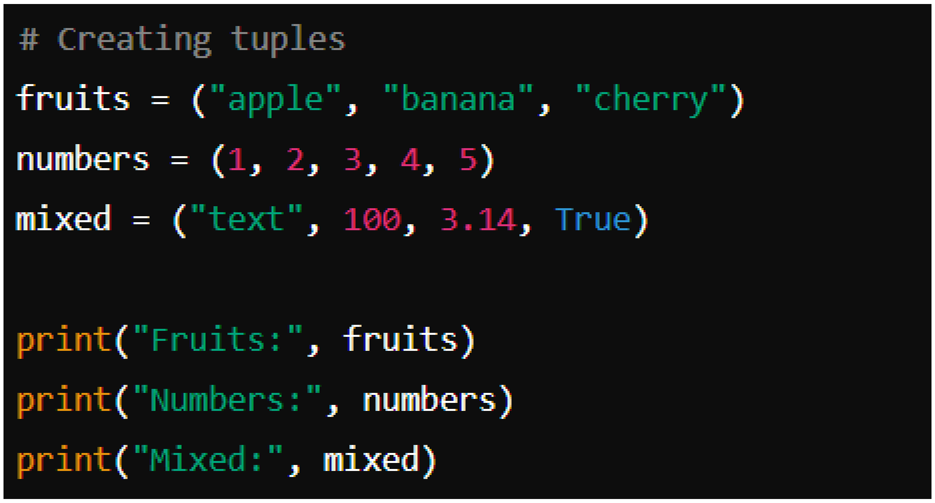
Task 6 - Tuples

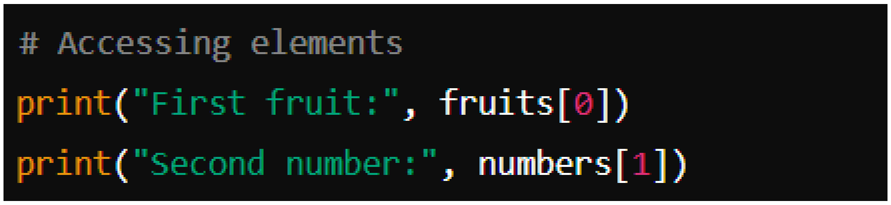
Definition: A tuple is an ordered collection of elements that is immutable, meaning once it is created, its elements cannot be changed. Tuples can hold multiple data types and allow duplicate elements.

Creating Tuples: Tuples are created by placing elements inside parentheses `()` and separating them with commas.



```python
# Creating tuples
fruits = ("apple", "banana", "cherry")
numbers = (1, 2, 3, 4, 5)
mixed = ("text", 100, 3.14, True)

print("Fruits:", fruits)
print("Numbers:", numbers)
print("Mixed:", mixed)
```



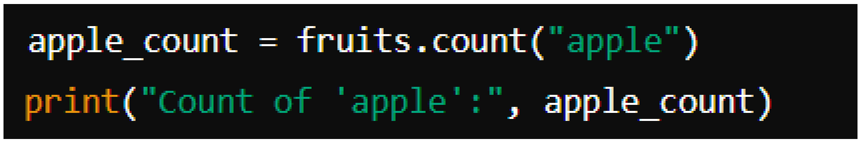
```python
# Accessing elements
print("First fruit:", fruits[0])
print("Second number:", numbers[1])
```

Immutability of Tuples

Once a tuple is created, its elements cannot be modified, added, or removed. This immutability makes tuples useful for ensuring data integrity.

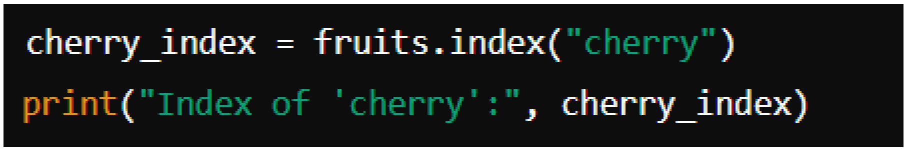
Tuple Methods and Functions

1. `count()`



```python
 apple_count = fruits.count("apple")
print("Count of 'apple':", apple_count)
```

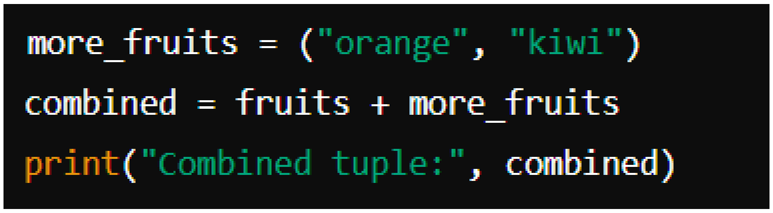
2. `index()`



```
cherry_index = fruits.index("cherry")
print("Index of 'cherry':", cherry_index)
```

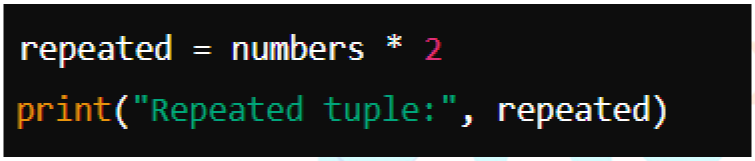
Additional Tuple Operations

1. Concatenation



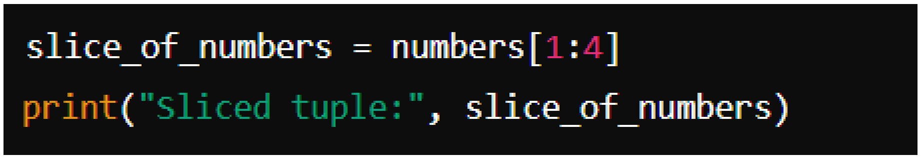
```
more_fruits = ("orange", "kiwi")
combined = fruits + more_fruits
print("Combined tuple:", combined)
```

2. Repetition



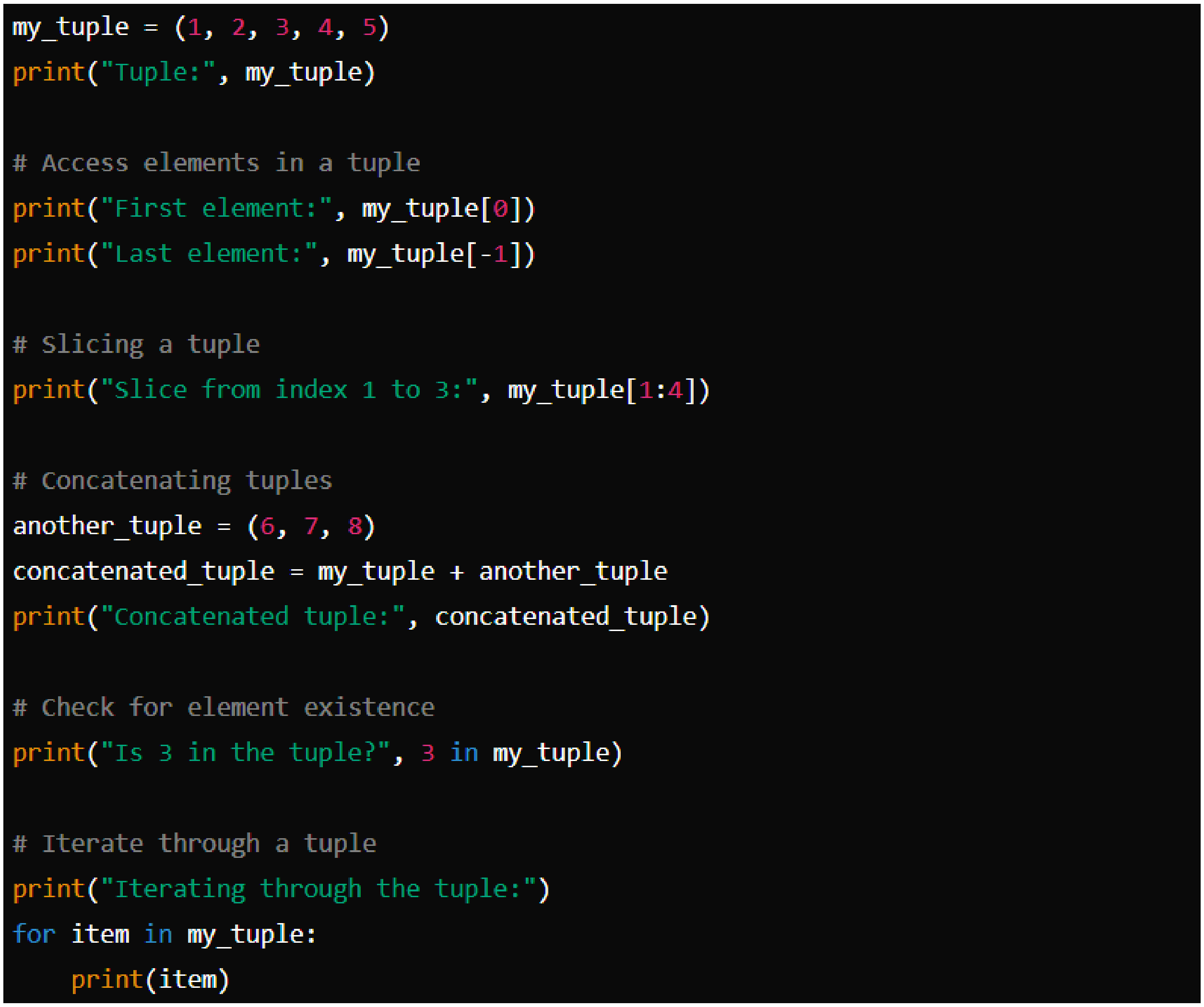
```
repeated = numbers * 2
print("Repeated tuple:", repeated)
```

3. Slicing



```
slice_of_numbers = numbers[1:4]
print("Sliced tuple:", slice_of_numbers)
```

Task 6: Write a program to demonstrate working with tuples in python



```python
my_tuple = (1, 2, 3, 4, 5)
print("Tuple:", my_tuple)

# Access elements in a tuple
print("First element:", my_tuple[0])
print("Last element:", my_tuple[-1])

# Slicing a tuple
print("Slice from index 1 to 3:", my_tuple[1:4])

# Concatenating tuples
another_tuple = (6, 7, 8)
concatenated_tuple = my_tuple + another_tuple
print("Concatenated tuple:", concatenated_tuple)

# Check for element existence
print("Is 3 in the tuple?", 3 in my_tuple)

# Iterate through a tuple
print("Iterating through the tuple:")
for item in my_tuple:
    print(item)
```