**Tuples**

A tuple is an immutable (unchangeable) sequence type in Python that can hold a collection of elements. They are similar to lists, but unlike lists, tuples cannot be modified after they are created.

**Creating Tuples**

# Creating an empty tuple

empty\_tuple = ()

# Creating a tuple with multiple elements

fruit\_tuple = ('apple', 'banana', 'cherry')

# Creating a tuple without parentheses

number\_tuple = 1, 2, 3, 4, 5

# Creating a tuple with one element (note the comma)

single\_element\_tuple = ('apple',)

# Accessing elements in a tuple

print(fruit\_tuple[0]) # Output: 'apple'

print(fruit\_tuple[1]) # Output: 'banana'

print(fruit\_tuple[-1]) # Output: 'cherry'

**Tuple Methods**

While tuples are immutable, they have a few useful methods:

1. **count()**: Returns the number of times a specified value appears in the tuple.

numbers = (1, 2, 3, 2, 2, 4, 5)

x= numbers.count(2)

print(x)

# Output: 3

1. **index()**: Returns the index of the first occurrence of the specified value.

numbers = (1, 2, 3, 2, 2, 4, 5)

x= numbers.index(3)

print(x)

# Output:2

**Min and Max Method**

numbers = (8, 2, 5, 1, 9, 7)

# Finding the smallest element in the tuple

smallest = min(numbers)

print(f"The smallest element is {smallest}") # Output: The smallest element is 1

# Finding the largest element in the tuple

largest = max(numbers)

print(f"The largest element is {largest}") # Output: The largest element is 9

**sum()**

Calculates the sum of all elements in a tuple (assuming they are numeric).

python

numbers = (8, 2, 5, 1, 9, 7)

total\_sum = sum(numbers)

print(f"The sum of all elements is {total\_sum}") # Output: The sum of all elements is 32

**len()**

Returns the number of elements in a tuple.

numbers = (8, 2, 5, 1, 9, 7)

length = len(numbers)

print(f"The number of elements is {length}")

# Output: The number of elements is 6

**sorted()**

Returns a sorted list of the elements in the tuple.

numbers = (8, 2, 5, 1, 9, 7)

sorted\_numbers = sorted(numbers)

print(f"The sorted elements are {sorted\_numbers}")

# Output: The sorted elements are [1, 2, 5, 7, 8, 9]

**any() and all()**

* any(): Returns True if any element of the tuple is True. If the tuple is empty, it returns False.
* all(): Returns True if all elements of the tuple are True. If the tuple is empty, it returns True.

values = (1, 0, 3, 0)

print(any(values)) # Output: True (since 1 and 3 are True)

print(all(values)) # Output: False (since 0 is False)

**Tuple Operations**

Here are a few common tuple operations:

**Concatenation**

You can concatenate tuples using the + operator:

tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

combined\_tuple = tuple1 + tuple2 # Output: (1, 2, 3, 4, 5, 6)

**Repetition**

You can repeat a tuple multiple times using the \* operator:

repeated\_tuple = (1, 2, 3) \* 3 # Output: (1, 2, 3, 1, 2, 3, 1, 2, 3)

**Slicing**

You can slice tuples to create a new tuple:

sliced\_tuple = combined\_tuple[1:4] # Output: (2, 3, 4)