**📘 Tuple Operations – Demonstration & Explanation**

**✅ 1. Create a Set of Tuples**

set\_data = {(1, "hello"), (2, 4), (9, 0)}

print(f"set of tuples: {set\_data}")

✅ Sets can contain immutable types like tuples.  
❌ Duplicate tuples are automatically removed.

**✅ 2. Create and Print a Tuple**

tuple\_data = tuple(x for x in range(1, 20))

print(f"Original tuple: {tuple\_data}")

tuple() converts an iterable (like a generator or list) into an immutable tuple.

**✅ 3. Access Tuple Elements**

print(f"First: {tuple\_data[0]}")

print(f"Middle: {tuple\_data[len(tuple\_data)//2-1]}")

print(f"Last: {tuple\_data[-1]}")

Indexing lets you access elements by position.

* tuple[0]: First element
* tuple[-1]: Last element
* tuple[len(tuple)//2]: Middle

**✅ 4. Slice Tuples**

print(tuple\_data[0:3]) # First 3 elements

print(tuple\_data[-3:]) # Last 3 elements

print(tuple\_data[2:6]) # Elements 2 to 5

Slicing returns a new tuple: tuple[start:stop]  
Negative indices count from the end.

**✅ 5. Concatenate Tuples**

tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

tuple3 = tuple1 + tuple2

print(tuple3)

+ operator joins tuples into a new one.

**✅ 6. Nested Tuple (Matrix Access)**

matrix = (

(1, 2, 3),

(4, 5, 6),

(7, 8, 9)

)

print(matrix[1][2]) # 6

print(matrix[0][1]) # 2

Tuples can contain other tuples (nested structure), like a 2D matrix.

**✅ 7. Count and Index Methods**

my\_tuple = (1, 2, 2, 3, 3, 3, 4, 4)

print(my\_tuple.count(2)) # Count of 2

print(my\_tuple.index(2)) # First index of 2

.count(x): Count occurrences of value  
.index(x): Return first index of value

**✅ 8. Unpack Tuple into Variables**

my\_tuple = (x for x in range(1, 6)) # Generator

a, b, c, d, e = my\_tuple

print(a, b, c, d, e)

Tuple unpacking assigns values to multiple variables.  
⚠️ Must be a fixed-size iterable, not a generator.

✅ **Fix**:

my\_tuple = tuple(x for x in range(1, 6))

**✅ 9. Convert List to Tuple**

lst = [1, 2, 3, 4, 5]

tuple\_list = tuple(lst)

print(tuple\_list)

Converts a mutable list to an immutable tuple.

**✅ 10. Tuple of Tuples (Nested Tuples)**

tupledata = ((1, 2, 3), ("hello", "hey", "bye"), (True, False, 0))

for item in tupledata:

print(item)

Tuples can contain other tuples — this is a **tuple of tuples**.

**✅ 11. Join Tuple of Characters into a String**

tuple\_chars = ("h", "e", "l", "l", "o")

joined = ''.join(tuple\_chars)

print(joined)

.join() concatenates elements into a single string.

**✅ 12. Dictionary with Tuple Keys**

dictionary\_data = { (1, 2): 3, (3, 2): 4 }

print(dictionary\_data)

Tuples are hashable and can be used as dictionary keys.

**✅ 13. Iterate Over a Nested Tuple**

tupledata = ((1, 2), (3, 2), (5, 6))

for subtuple in tupledata:

for item in subtuple:

print(item, end=" ")

print()

Nested iteration for nested tuple structures.

**✅ 14. Remove Duplicates by Converting Tuple to Set**

tupledata = (1, 2, 2, 3, 4, 5, 6, 6, 6)

setdata = set(tupledata)

print(setdata)

Sets automatically remove duplicate elements.

**✅ 15. Tuple Aggregation Functions**

def functiontupleMinMax(tupledata, flag):

if flag == "Min":

return min(tupledata)

if flag == "Max":

return max(tupledata)

if flag == "Sum":

return sum(tupledata)

print(functiontupleMinMax(tupledata, "Min"))

print(functiontupleMinMax(tupledata, "Max"))

print(functiontupleMinMax(tupledata, "Sum"))

Use built-in functions:

* min(), max(), and sum()
* These work on numeric tuples

**🧾 Summary: Common Tuple Functions**

| **Function** | **Description** |
| --- | --- |
| len(tup) | Length of tuple |
| min(tup) | Minimum element |
| max(tup) | Maximum element |
| sum(tup) | Sum of all elements |
| tup.count(x) | Count occurrences of x |
| tup.index(x) | Index of first occurrence of x |
| tuple(iterable) | Converts an iterable to a tuple |