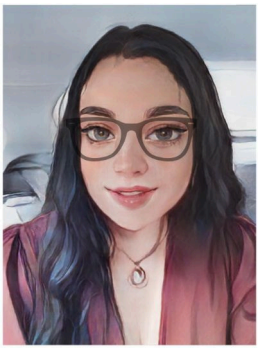

@pythonessdatadiaries

Python Tuple Practice Exercises:

Basic to Advanced (with Solution)📚

🥇 Basic Level

1. Create a tuple with numbers, strings, boolean values and print it.

Code:

```
my_tuple = (10,20.4,"A",'It\'s Python',True,0)
print(my_tuple)
```

Output:

```
(10, 20.4, 'A', "It's Python", True, 0)
```

🔹 *Explanation: Tuple created using ()(Parenthesis) .*
Elements are ordered, immutable.

2. Access the second element from a tuple using Indexing.

Code:

```
my_tuple = (10,20.4,"A",'It\'s Python',True,0)
print(my_tuple[1])
```

Output:

```
20.4
```

🔹 *Explanation: Tuples are 0-indexed. my_tuple[1] = 20.4*

3. Find the length of a tuple.

Code:

```
my_tuple = (10,20.4,"A",'It\'s Python',True,0)
print(len(my_tuple))
```

Output:

```
6
```

🔹 *Explanation: len() function counts number of elements.*

4. Check if an element exists in a tuple.

Code:

```
my_tuple = (10,20.4,"A",'It\'s Python',True,0)

print("a" in my_tuple) #Case-sensitive
print("A" in my_tuple) #Correct
```

Output:

```
False
True
```

🔹 *Explanation: in keyword returns True or False.*

***Strings in Python are case-sensitive.*

5. Find maximum and minimum in a numeric tuple.

Code:

```
my_tuple = (10,45,66,92)

print(max(my_tuple))
print(min(my_tuple))
```

Output:

```
92
10
```

🔹 *Explanation: max() and min() find highest and lowest.*

2 Intermediate Level

6. Count occurrences of an element in a tuple.

Code:

```
my_tuple = (1,2,3,2,2.0,4,3)

print(my_tuple.count(2))
```

Output:

```
3
```

🔹 *Explanation: .count(value) returns how many times value appears.*

** *In Python, 2 and 2.0 are equal, so count(2) counts both.*

7. Find index of an element.

Code:

```
my_tuple = (1,2,3,2,2.0,4,3)

print(my_tuple.index(4))
```

Output:

```
5
```

🔹 *Explanation: .index(value) returns position of first occurrence.*

8. Convert a List into a Tuple.

Code:

```
my_list = [10,14,21,6]
my_tuple = tuple(my_list)

print(my_tuple)
```

Output:

```
(10, 14, 21, 6)
```

🔹 *Explanation: tuple() function converts list → tuple.*

9. Unpack tuple into variables.

Code:

```
my_tuple = (10,20,30)
a,b,c = my_tuple

print(a,b,c)
```

Output:

```
10 20 30
```

🔹 *Explanation: Tuple unpacking distributes values into variables.*

10. Slice a tuple to get a part of it.

Code:

```
#indexing:   0, 1, 2, 3, 4
my_tuple = (10,20,30,40,50)

print(my_tuple[1:3])
```

Output:

```
(20, 30)
```

🔹 *Explanation: my_tuple[1:3] returns elements from index 1 to 2 (leaving last index:3, as per Python Slicing rule)*

🥇 Advanced Level

11. Swap two tuples.

Code:

```
a = (1,2,3) #tuple a defined
b = (3,4,7) # tuple b defined

#swapping
a,b = b,a #tuple a and b are swapped

print(a)
print(b)
```

Output:

```
(3, 4, 7)
(1, 2, 3)
```

🔹 *Explanation: In Python, swap two tuples in one line using multiple assignment, no temp variable needed.*

12. Merge two tuples.

Code:

```
a = (1,2,3) #tuple a defined
b = (3,4,7) # tuple b defined

#merging
merged = a +b #tuple a and b are merged/added

print(merged)
```

Output:

```
(1, 2, 3, 3, 4, 7)
```

🔹 *Explanation: + operator joins tuples.*

13. Multiply tuple elements by an integer.

Code:

```
t = (11,22,33) #tuple t defined

#multiplying by int : 3
print(t * 3)
```

Output:

```
(11, 22, 33, 11, 22, 33, 11, 22, 33)
```

🔹 *Explanation: * repeats the tuple 3 times.*

14. Find common elements between two tuples.

Code:

```
t1 = (1,2,3) #tuple t1 defined
t2 = (4,3,5) #tuple t1 defined

#Using set intersection &
print(tuple(set(t1) & set(t2)))
```

Output:

```
(3,)
```

🔹 *Explanation: Using set intersection to find common elements.*

15. Create nested tuple.

Code:

```
t1 = ((1,2),(3,6),(5,(7,8))) #nested tuple t1 defined

print(t1)

#Use indexing to access Nested Tuple
print(t1[1][1])
```

Output:

```
((1, 2), (3, 6), (5, (7, 8)))
6
```

🔹 *Explanation: Access inner tuple elements using double indexing.*

16. Check if all elements are numbers in a tuple.

Code:

```
t1 = (1,2,'A',10,7.2) #tuple t1 defined

#Use insinstance() to check if all elements are numbers
print(isinstance(t1,int))

#For each element ,use all and for loop to check if elements are numbers
print(all(isinstance(t1,int) for i in t1))
```

Output:

```
False
False
```

🔹 *Explanation: all() and isinstance() check data types.*

🔹 *all( ) checks if all items in the tuple are True,then output is True and isinstance( ) checks if a value belongs to a specific data type like int, str, etc.*

17. Find sum of all elements in a tuple.

Code:

```
t = (2,5,4,7) #tuple t defined
print(sum(t))
```

Output:

```
18
```

🔹 ***Explanation: sum() adds all numbers.***

18. Convert tuple of strings to a single string.

Code:

```
t = ("Learn","Python","with","me") #tuple t defined
print(" ".join(t))
```

Output:

```
Learn Python with me
```

🔹 ***Explanation: ' '.join() joins tuple elements with space.***

19. Sort a tuple of numbers.

Code:

```
t = (10,50,40,20) #tuple t defined
print(tuple(sorted(t)))
```

Output:

```
(10, 20, 40, 50)
```

🔹 ***Explanation: sorted() gives list, then convert back to tuple.***

20. Write a function to return only unique elements from a tuple.

Code:

```
def unique_elements(t):
    return tuple(set(t))

my_tuple = (1,2,2,3,1,4)

print(unique_elements(my_tuple))
```

Output:

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

🔹 ***Explanation: set() removes duplicates, then tuple() restores format.***

🚀 Summary Points:

- Tuples are faster, lighter, and immutable.
- Useful when fixed data is needed (like coordinates, RGB colors).
- Tuple methods are simple: .count() and .index().
- We can slice, merge, and even use set operations with tuples!


HEY PYTHON, CAN I JUST CHANGE ONE LITTLE VALUE IN A TUPLE?
**PYTHON:
Impossible.