Day 9 Final

Tuples in python

Creating a Tuple

In [3]:

```
my_tuple = (1, 2, 3)
print(my_tuple)
print(type(my_tuple))
```

```
(1, 2, 3)
<class 'tuple'>
```

tuple with mixed datatypes

In [5]:

```
my_tuple = (1, "Hello", 3.4)
print(my_tuple)
print(type(tuple))
```

```
(1, 'Hello', 3.4)
<class 'type'>
```

Empty tuple

In [7]:

```

my_tuple = ()
print(my_tuple)
print(type(my_tuple))
```

```
()
<class 'tuple'>
```

nested tuple

In [8]:

```
my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
print(my_tuple)
print(type(my_tuple))
```

```
('mouse', [8, 4, 6], (1, 2, 3))
<class 'tuple'>
```

Create a Python Tuple With one Element

In [10]:

```
var1 = ("Hello")
print(var1)
print(type(var1))
```

```
Hello
<class 'str'>
```

In [11]:

```
var2 = ("Hello",)
print(var2)
print(type(var2))
```

```
('Hello',)
<class 'tuple'>
```

The tuple() Constructor

In [13]:

```
thistuple = tuple(("a", "b", "c"))
print(thistuple)
print(type(thistuple))
```

```
('a', 'b', 'c')
<class 'tuple'>
```

Indexing

In [14]:

```
a = (47,12,63,54,12,78,48,36)
print(a[5])
```

```
78
```

In [15]:

```
b = ("data","python",25,2.36,5+6j)
print(b[4])
```

```
(5+6j)
```

In [16]:

```
c = (12,23,14,15,26,7,8,9,10)
print(c[-1])
```

```
10
```

In [17]:

```
c = (12,23,14,15,26,7,8,9,10)
print(c[-4])
```

```
7
```

Slicing

In [18]:

```
a = (45,21,31,14,16,28,37,69)
print(len(a))
print(a[0:6])
```

```
8
(45, 21, 31, 14, 16, 28)
```

In [19]:

```
a = (45,21,31,14,16,28,37,69)
print(len(a))
print(a[:6])
```

```
8
(45, 21, 31, 14, 16, 28)
```

In [20]:

```
a = (45,21,31,14,16,28,37,69)
print(len(a))
print(a[:9])
```

```
8
(45, 21, 31, 14, 16, 28, 37, 69)
```

In [21]:

```
a = (45,21,31,14,16,28,37,69)
print(len(a))
print(a[:])
```

```
8
(45, 21, 31, 14, 16, 28, 37, 69)
```

In [22]:

```
a = (45,21,31,14,16,28,37,69)
print(len(a))
print(a[::2])
```

```
8
(45, 31, 16, 37)
```

reversing the tuple

In [24]:

```
a = (45,21,31,14,16,28,37,69)
print(len(a))
print(a[::-1])
```

```
8
(69, 37, 28, 16, 14, 31, 21, 45)
```

Write the output of the Following Code

In [15]:

```
T = (1,2,3,4,5,6,7,8)
print(T[::])
print(T[3:])
print(T[:4])
print(T[-2:-5])
```

```
(1, 2, 3, 4, 5, 6, 7, 8)
(4, 5, 6, 7, 8)
(1, 2, 3, 4)
()
```

Duplicates are allowed in tuples

In [25]:

```
a = (12,13,12,15,12,13,15,17,18)
print(a)
print(type(a))
```

```
(12, 13, 12, 15, 12, 13, 15, 17, 18)
<class 'tuple'>
```

Once a tuple is created, you cannot change its values. Tuples are unchangeable, or immutable as it also is called.

Convert the tuple into a list to be able to change it:

In [26]:

```
x = ("data", "analyst", "python")
y = list(x)
y[1] = "science"
x = tuple(y)

print(x)
```

```
('data', 'science', 'python')
```

Add Items

Since tuples are immutable, they do not have a built-in append() method, but there are other ways to add items to a tuple

In [28]:

```
a = (25,26,27,28,29)
y = list(a)
y.append("orange")
a = tuple(y)
print(a)
```

```
(25, 26, 27, 28, 29, 'orange')
```

Add tuple to a tuple

In [29]:

```
tuple = ("python", "data", "work")
y = ("code",)
tuple += y

print(tuple)
```

```
('python', 'data', 'work', 'code')
```

Remove Items

Tuples are unchangeable, so you cannot remove items from it, but you can use the same workaround as we used for changing and adding tuple items:

In [3]:

```
x = ("apple", "banana", "cherry")
y = list(x)
y.remove("apple")
x = tuple(y)
print(x)
```

```
('banana', 'cherry')
```

The del keyword can delete the tuple completely:

In [5]:
```
a = ("a", "b", "c",25,28,2.36)
del a
print(a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_9780\1854212602.py in <module>
      1 a = ("a", "b", "c",25,28,2.36)
      2 del a
----> 3 print(a)

NameError: name 'a' is not defined
```

Loop Through a Tuple

In [6]:
```
thistuple = ("a", "b", "c")
for x in thistuple:
    print(x)
```

```
a
b
c
```

In [7]:
```
thistuple = ("apple", "banana", "cherry")
i = 0
while i < len(thistuple):
  print(thistuple[i])
  i = i + 1
```

```
apple
banana
cherry
```

Join Two Tuples

In [8]:
```
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)

tuple3 = tuple1 + tuple2
print(tuple3)
```

```
('a', 'b', 'c', 1, 2, 3)
```

Multiply Tuples

In [9]:

```
fruits = ("apple", "banana", "cherry")
mytuple = fruits * 2

print(mytuple)
```

```
('apple', 'banana', 'cherry', 'apple', 'banana', 'cherry')
```

Write a Python program to find repeated items in a tuple.

In [10]:

```
#create a tuple
tuplex = 2, 4, 5, 6, 2, 3, 4, 4, 7
print(tuplex)
#return the number of times it appears in the tuple.
count = tuplex.count(4)
print(count)
```

```
(2, 4, 5, 6, 2, 3, 4, 4, 7)
3
```

write the output of the following code

In [14]:

```
T = (1,2,3,4,5,6,7,8)
print(T*2)
print(T+T)
print(len(T))
```

```
(1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8)
(1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8)
8
```

Write a program to accept 5 numbers from the user and store it in a Tuple

In [1]:

```
T = ()
i = 0
while i<=5:
    num= int(input("Enter the number"))
    T = T+ (num,)
    i=i+1
    print(T)
```

```
Enter the number10
(10,)
Enter the number11
(10, 11)
Enter the number12
(10, 11, 12)
Enter the number13
(10, 11, 12, 13)
Enter the number14
(10, 11, 12, 13, 14)
Enter the number15
(10, 11, 12, 13, 14, 15)
```

In [ ]: