Tuples

September 25, 2024

[1]:
```
Tup1= ('Asif', 25 ,[50, 100],[150, 90] , {'John' , 'David'} , (99,22,33))
Tup1
```

[1]:
```
('Asif', 25, [50, 100], [150, 90], {'David', 'John'}, (99, 22, 33))
```

[2]:
```
tup1 = ('physics', 'chemistry', 1997, 2000);
tup2 = (1, 2, 3, 4, 5 );
tup3 = "a", "b", "c", "d";
tup3
```

[2]:
```
('a', 'b', 'c', 'd')
```

[13]:
```
tup4=tup2+tup3
tup4
```

[13]:
```
(1, 2, 3, 4, 5, 'a', 'b', 'c', 'd')
```

[6]:
```
print(tup1[0])
print(Tup1[3][1])
```

```
physics
90
```

[7]:
```
tup3[-1]
```

[7]:
```
'd'
```

[9]:
```
mytuple = ('one' , 'two' , 'three' , 'four' , 'five' , 'six' , 'seven' , 'eight')
mytuple
```

[9]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

[10]:
```
del mytuple[0] # Tuples are immutable which means we can't DELETE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-10-96051e0b9682> in <module>
----> 1 del mytuple[0] # Tuples are immutable which means we can't DELETE tuple␣
  ↪items
```

```
TypeError: 'tuple' object doesn't support item deletion
```

[11]:
```
mytuple[0] = 1 # Tuples are immutable which means we can't CHANGE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-11-4c2ed09725a9> in <module>
----> 1 mytuple[0] = 1 # Tuples are immutable which means we can't CHANGE tuple␣
 ↪items

TypeError: 'tuple' object does not support item assignment
```

[ ]:
```
del mytuple # Deleting entire tuple object is possible
```

[14]:
```
print 'abc', -4.24e93, 18+6.6j, 'xyz';
```

```
  File "<ipython-input-14-3ca6aed8b79d>", line 1
    print 'abc', -4.24e93, 18+6.6j, 'xyz';
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print('abc', -4.
 ↪24e93, 18+6.6j, 'xyz')?
```

[15]:
```
a=1,2,"three",4.5,"x"
print(type(a))
```

```
<class 'tuple'>
```

[16]:
```
aList = ['xyz', 'zara', 'abc']
aTuple = tuple(aList)
print ("Tuple elements : ", aTuple)
```

```
Tuple elements :  ('xyz', 'zara', 'abc')
```

[17]:
```
tuple1, tuple2 = (123, 'xyz'), (456, 'abc')
print cmp(tuple1, tuple2)
print cmp(tuple2, tuple1)
tuple3 = tuple2 + (786,);
print cmp(tuple2, tuple3)
```

```
  File "<ipython-input-17-1958e5faec8c>", line 2
    print cmp(tuple1, tuple2)
          ^
SyntaxError: invalid syntax
```

[20]:

```
s = 'abc'
t = [0, 1, 2]
z = zip(s, t)
print(tuple(z))
print(z)
```

```
(('a', 0), ('b', 1), ('c', 2))
<zip object at 0x0000026F3F61A100>
```

[21]:

```
t = [('a', 0), ('b', 1), ('c', 2)]
for letter, number in t:
    print (number, letter)
```

```
0 a
1 b
2 c
```

[23]:

```
l=list()
dir(l)
```

[23]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

[24]:

```
t= tuple()
dir(t)
```

[24]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'count',
 'index']
```

[25]:

```python
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = thistuple.index(8)

print(x)
```

```
3
```

[26]:

```python
(x, y) = (4, 'fred')
print(y)

(a, b) = (99, 98)
print(a)
```

```
fred
99
```

[27]:

```python
l=list()
l.append("hi")
l
```

[27]:

```
['hi']
```

[1]:

```python
aList = ['xyz', 'zara', 'abc']
aTuple = tuple(aList)
print ("Tuple elements : ", aTuple)
```

```
Tuple elements :  ('xyz', 'zara', 'abc')
```

[2]:

```python
# iterable dictionary
dict1 = {'Name': 'Rahul', 'Hobby': 'Singing', 'RollNo': 45}
# using tuple() method
res = tuple(dict1)
# printing the result
print("dictionary to tuple:", res)
```

```
dictionary to tuple: ('Name', 'Hobby', 'RollNo')
```

[4]:

```python
# iterable string
string = "Welcome";
# using tuple() method
res = tuple(string)
# printing the result
print("converted string to tuple:", res)
```

```
converted string to tuple: ('W', 'e', 'l', 'c', 'o', 'm', 'e')
```

[5]:

```python
# Input a string from the user
input_str = input("Enter a string: ")

# Convert the string to a list of characters
char_list = list(input_str)

# Reverse the list
reversed_list = char_list[::-1]

# Check if the original list is equal to the reversed list
if char_list == reversed_list:
    print(f"'{input_str}' is a palindrome.")
else:
    print(f"'{input_str}' is not a palindrome.")
```

```
Enter a string: hello
'hello' is not a palindrome.
```

[8]:

```python
# Initialize an empty list to store contacts
contacts = []

# Main program loop
while True:
    print("\nContact Book")
    print("1. Add Contact")
    print("2. Display Contacts")
    print("3. Search Contact")
    print("4. Exit")

    choice = input("Choose an option (1-4): ")
    if choice == '1':
        # Adding a new contact
        name = input("Enter name: ")
        phone = input("Enter phone number: ")
        email = input("Enter email address: ")

        # Create a tuple for the contact
        contact = (name, phone, email)
        # Add the contact to the list
        contacts.append(contact)
        print("Contact added successfully.")
    elif choice == '2':
        # Displaying all contacts
        if not contacts:
            print("No contacts found.")
        else:
            print("\nContacts:")
            for index, contact in enumerate(contacts):
                print(f"{index + 1}. Name: {contact[0]}, Phone: {contact[1]},␣
↪Email: {contact[2]}")
    elif choice == '3':
        # Searching for a contact by name
        search_name = input("Enter the name of the contact to search: ")
        found = False

        for contact in contacts:
            if contact[0].lower() == search_name.lower():  # Case-insensitive␣
↪search
                print(f"Contact found: Name: {contact[0]}, Phone: {contact[1]},␣
↪Email: {contact[2]}")
                found = True
                break

        if not found:
            print("Contact not found.")

    elif choice == '4':
        # Exiting the program
        print("Exiting the program.")
        break

    else:
        print("Invalid choice. Please enter a number between 1 and 4.")
```

```
Contact Book
1. Add Contact
2. Display Contacts
3. Search Contact
4. Exit
Choose an option (1-4): 1
Enter name: Hema
Enter phone number: 254565245
Enter email address: xyz@hmail.com
Contact added successfully.
Contact Book
1. Add Contact
2. Display Contacts
3. Search Contact
4. Exit
Choose an option (1-4): 2

Contacts:
1. Name: Hema, Phone: 254565245, Email: xyz@hmail.com

Contact Book
1. Add Contact
2. Display Contacts
3. Search Contact
4. Exit
Choose an option (1-4):
Invalid choice. Please enter a number between 1 and 4.

Contact Book
1. Add Contact
2. Display Contacts
3. Search Contact
4. Exit
Choose an option (1-4): 3
Enter the name of the contact to search: Hema
Contact found: Name: Hema, Phone: 254565245, Email: xyz@hmail.com

Contact Book
1. Add Contact
2. Display Contacts
3. Search Contact
4. Exit
Choose an option (1-4): 4
Exiting the program.
```

[ ]: