

لو انا عايز اعرف Tuple بس تكون بعنصر واحد بس، فى الطبيعى لو انا عملت متغير وحطيت عنصر واحد بس هيتعرف على انه string. فعلشان اعرفه على انه Tuple بحط (,) comma بعد قيمة المتغير.

ال concatenation فى ال tuple بيكون عادى جدا شبه ال list بيشم عناصر كل ال tuples فى tuple واحد.

لو ضربت string او list او tuple فى رقم بيعمل repeat يعنى بيكرر القيم عدد مرات الرقم اللى ضربت فيه ويحطهم فى tuple فى حالة ال tuple و list واحده فى حالة ال list وفى حالة ال string بيحطهم جنب بعض.

=====

count()

بتعدلى عدد وجود مرات عنصر معين ف ال Tuple (زى ال list).

ال Argument بتاخذ العنصر اللى عايز ابحت عنه متكرر كام مره وهى بترجعلى عدد مرات تكراره جوا ال Tuple.

index()

بيرجعلى ال index بتاع عنصر معين جوا ال Tuple (زى ال list).

ال Argument بتاخذ العنصر اللى عايز اعرف رقم ال index بتاعه جوا ال Tuple .

=====

لما كنت بعرف اكثر من متغير على نفس السطر بالطريقه دى (x, y, z = 1, 2, 3) ينفع انى بدل م احط القيم اللى هى 1 و 2 و 3 دى انى احطهم ف tuple (a = (1,2,3)) واقول ان (x, y, z = a) وهيبقا كأنى بعرفهم بالطريقه العاديه اللى هو اكثر من متغير ف سطر واحد، طيب لو انا عندي ال tuple فيه عناصر اكثر من عدد المتغيرات اللى عايز اعرفها كدا هيديني error لان بقله ان 3 متغيرات = 4 قيم مثلا، فعلشان اهمل قيمه من جوا ال tuple بحط مكانها فى المتغيرات (_) underscore وكدا هيتم اهمال القيمه المناظره ليها من ال tuple. (الكود برضو يفهم اوضح)

```
# -----
# -- Tuple --
# -----

# Tuple With One Element

myTuple1 = ("Muhammad",)
myTuple2 = "Muhammad",

print(myTuple1)
print(myTuple2)

print(type(myTuple1))
print(type(myTuple2))

print(len(myTuple1))
print(len(myTuple2))
```

```

# Tuple Concatenation

a = (1, 2, 3, 4)
b = (5, 6)

c = a + b
d = a + ("A", "B", True) + b

print(c)
print(d)

# Tuple, List, String Repeat (*)

myString = "Muhammad"
myList = [1, 2]
myTuple = ("A", "B")

print(myString * 6)
print(myList * 6)
print(myTuple * 6)

# Methods => count()

a = (1, 3, 7, 8, 2, 6, 5, 8)
print(a.count(8))

# Methods => index()

b = (1, 3, 7, 8, 2, 6, 5)
# print("The Position of Index Is: " + b.index(7)) # Error
print("The Position of Index Is: {:d}".format(b.index(7)))
print(f"The Position of Index Is: {b.index(7)}")

# Tuple Destruct

a = ("A", "B", 4, "C")

x, y, _, z = a

print(x)
print(y)
print(z)

```

CODE

```
exe "c:/Users/Muhammad/Documents/Python Course/first.py"
('Muhammad',)
('Muhammad',)
<class 'tuple'>
<class 'tuple'>
1
1
(1, 2, 3, 4, 5, 6)
(1, 2, 3, 4, 'A', 'B', True, 5, 6)
MuhammadMuhammadMuhammadMuhammadMuhammadMuhammad
[1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2]
('A', 'B', 'A', 'B', 'A', 'B', 'A', 'B', 'A', 'B', 'A', 'B')
2
The Position of Index Is: 2
The Position of Index Is: 2
A
B
C
```

OUTPUT