
Python
Data Structures
Tuples
Lists
Sets
Dictionaries

Tuple in Python

- is a sequence, like a list
- indexed by int; leftmost element has an index 0
- Select the element using []
- Immutable
- once created, cannot be changed
- length of the tuple cannot change
- Heterogeneous
- Iterable

Tuples

Creating

```
a = (11, 33, 22, 44, 55)
print(a)
```

Tuples Packing

```
b= 1, 2.0, 'three‘
print(b)
```

Tuples Unpacking

```
marks=(99,95,90,89,93,96)
a,b,c,d,e,f=marks
print(a,b,c)
```

```
vowels = ('a', 'e', 'i', 'o', 'u')
print(vowels)
```

Creating a tuple with a single item

a=(1)

type(a) # int element

a=(1,)

type(a) # tuple element

A tuple with a single item

tuple of one element requires an extra comma

Accessing Python Tuples

Accessing the entire tuple

```
a=(10,20,20)
print(a)
```

Accessing a single item

```
print(a[0],a[1],a[2])
```

Slicing Tuples

Same as list

Negative indexing

Same as list

Deleting a Python Tuple

Deleting an element

```
a=(10,20,20)
del a[0]
```

Deleting an entire tuple

```
a=(10,20,20)
del a
```

Functions on Tuples

A=(10,20,30)

len(A)

max(A)

min(A)

sum(A)

sorted(A)

s="abc"

tuple() //converting some thing to a type called Tuple ('a','b','c')

Built in functions, used directly without object

Methods on Python Tuples

tuple methods

S=(10,20,30,40,50)

s.index() #position

s.count() #frequency of an item

Built in method,
used with object
Object.method()

Operations on Tuples in Python

Membership

10 in s #True

20 not in s #False

Concatenation

S1=(1,2,3)

S2=(4,5,6)

Print(s1+s2)

Logical (<,>…)

Identity

```
>>> a=(1,2)
>>> b=a
>>> a is b
        true
```

Accessing/printing/Iterating a Python Tuple

```
sub=('P','C','M','B')
print(sub)
for i in sub:
    print(i)
i=0
while(i<len(sub)):
    print(sub[i])
    i=i+1
print(sub[0:len(sub):1])
```

Accessing/printing tuple elements

sub=('P','C','M','B')

```
print(sub)
```

```
for  i in sub:
        print(i)
```

```
i=0
while(i<len(sub):
        print(sub[i])
        i=i+1
```

```
for  i in range(0,len(sub))
        print(sub[i])
```

```
for i in range(-len(sub),0,1):
    print(sub[i])
```

```
print(sub[0:len(sub):1])
```

More on Tuples

```
a = (11, 33, 22, 44, 55)
print(a)
print(a[2]) # 22
print(a[2:4]) # (22, 44)

#a[2] = 222 # NO
#a.append(66)  #AttributeError: 'tuple' object has no attribute 'append'
#ok; a new tuple created
a = a + (111, 222)
print(a)
```

More on Tuples

```
b = ([12, 23], {34 : 45}, "56" )
print(b, len(b))

b[0].append(67)  #ok
#b[0] = [78, 89]  #no
#del b[0] # no
#b[0] += [100] # no ; assignment forbidden
```

Nested structures

Nested Tuples

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

Nested lists

a=[[10,20,30],[1,2,3]]

len(a),len(t)

More on Tuples: Accessing/printing tuple elements

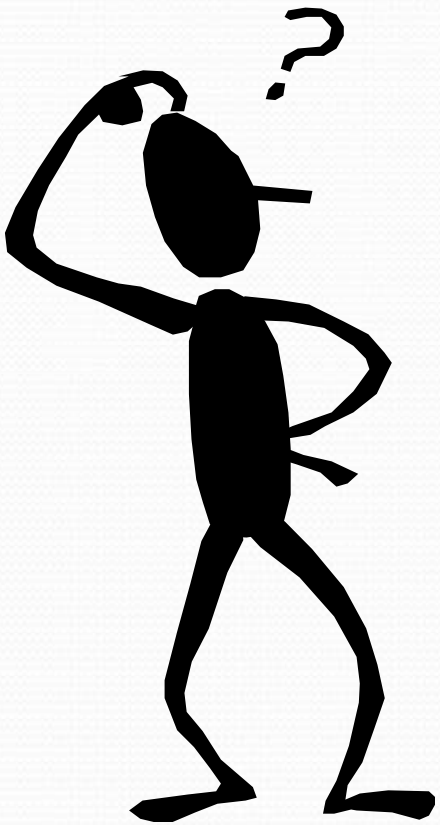
```
a = (11, 33, 22, 44, 55)
for i in a :
    print(i, end = " ")

c = [11, 22, 33, 44]
for i in c :
    print("one")  # 4 times

for i in [c] :
    print("two")  # once

for i in (c) : #  not a tuple
    print("three") # 4 times

for i in (c,) : #  a tuple
    print("four") # once
```



```
print( (3, 4) * 2) # (3, 4, 3, 4)
print( (3 + 4) * 2) #  14
print( (3 + 4,) * 2) #  (7, 7)
            # tuple of one element requires an extra comma
d = ()
print(d, type(d))   #tuple

d = (10)
print(d, type(d))  #int

e = (11, 33, 11, 11, 44, 33)
print(e.count(11)) # 3
print(e.count(33)) # 2
print(e.count(55)) # 0
print(e.index(44)) # 4
print(e.index(11)) # 0
print(e.index(55)) # error
```

```
a = 1, 2, 3          #packing
print(a, type(a))
x, y, z = a          #unpacking
print(x, y, z)
#q, w = a # error;
 # of variables on the left should match the # of elem in the tuple

# use of unnamed tuple , no name but behaves as tuple
a, b = 11, 22
# (a, b) = (11, 22)
print("a : ", a,  " b : ", b)

# in case of assignment, the right hand side is completely evaluated
    before assignment
(a, b) = (b, a) # swaps two variables
# (a, b) = (22, 11)
print("a : ", a,  " b : ", b)
```

Swapping two numbers

```
a=4
b=7
print(a,b)
temp=a
a=b
b=temp
print(a,b)
```

```
a=3
b=5
print(a,b)
a = a + b
b = a - b
a = a - b
print(a,b)
```

```
a=7
b=8
print(a,b)
a,b=b,a
print(a,b)
```

```
a = 5
b = 6
print(a,b)
a = a ^ b  # 0101 ^ 0110 => 0011 => 3
b = a ^ b  # 0011 ^ 0110 => 0101 => 5
a = a ^ b  # 0011 ^ 0101 => 0110 => 6
print(a,b)
```

Tuple assignment

```
a=18
b=20
print(a,b)
(a,b)=(b,a)
(a,b)=(8,7)

print(a,b)
```

```
a="PCMB"
b=list(a)
b.append('K') #ok

c=tuple(a)
c.append('E')  #error

print(a,b,c)
```