tuple

July 18, 2024

1 Tuple

Tuples in Python are ordered collections of items that are immutable (unchangeable) and can contain elements of different types. In this notebook, we'll explore the basic operations and characteristics of tuples.

1.1 Creation

Can be created with tuple() , () , (item,)

[1]:
```
print(tuple((1,2,3,4)))      #(1,2,3,4) can be any iterable
print(())                    #Empty Tuple
print((1,))                  # Single item as tuple
```

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

1.2 Manipulation

Consider a= (1,2,3,4,3,2,1) as a tuple for the following manipulations

[2]:
```
a=(1,2,3,4,3,2,1,0,-1,-2)
```

1.2.1 Tuple Unpacking and slice

[3]:
```
x,y,z=(1,2,3)        # If there are not exactly 3 elements it throws error
print(x,y,z)
s=slice(1,5)         # Creates a slice object which can be used in accessing␣
 ↪elements
print(a[s])
s=slice(1,6,2)       # Also slice can be used to as step
print(a[s])
s=slice(-3,0,-1)     # And we can give negative indexing
print(a[s])
```

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

```
(2, 4, 2)
(0, 1, 2, 3, 4, 3, 2)
```

[4]:
```
s=slice(None,None,-1)      # None in start and stop means beginning and end of␣
 ↪the tuple respectively
a[s]
```

[4]:
```
(-2, -1, 0, 1, 2, 3, 4, 3, 2, 1)
```

[5]:
```
#Count no of item occurance
print(a.count(3))   # Counts the number of occurance of item in the tuple
```

```
2
```

[6]:
```
# Returns Index
print(a.index(1))   # Returns the index of the first occurnce of the item
print(a.index(1,1)) # Returns the index of the first occurance from the 1st␣
 ↪index
```

```
0
6
```

[7]:
```
# Concatenate
print(a+(5,6,7,8))  # Adding a tuple to it
print(a*2)          # Multiplying the tuple with 2 or can be changed
```

```
(1, 2, 3, 4, 3, 2, 1, 0, -1, -2, 5, 6, 7, 8)
(1, 2, 3, 4, 3, 2, 1, 0, -1, -2, 1, 2, 3, 4, 3, 2, 1, 0, -1, -2)
```

[8]:
```
# Returns Length of Tuple
print(len(a))
```

```
10
```

[9]:
```
#Sorting a tuple
print(sorted(a))        # Returns a List of sorted tuple
```

```
[-2, -1, 0, 1, 1, 2, 2, 3, 3, 4]
```

[10]:
```
# Reversing a tuple
print(list(reversed(a)))  # Returns a reversed sequence
```

```
[-2, -1, 0, 1, 2, 3, 4, 3, 2, 1]
```

1.3 Accessing Elements

[11]:
```
print(a[3])      # Returns a item at index 3
print(a[1:5])    # Returns slice of the tuple from index start to end , item at␣
 ↪index at 5 not included
print(a[1:6:2]) # Returns slice of the tuple with a step of 2
```

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

[12]:
```
print(a[4:])        #  From 4th index to end
print(a[:4])        # Upto 4th index (not included)
```

```
(3, 2, 1, 0, -1, -2)
(1, 2, 3, 4)
```

[13]:
```
a[::-1] # Returns a reversed tuple
```

[13]:
```
(-2, -1, 0, 1, 2, 3, 4, 3, 2, 1)
```

1.4 Iterations in tuple

[14]:
```
# for iterations
for i in a:                  # 'For' uses sequence 'a' as length and iterates␣
 ↪every item in the tuple
    print(i,end=",")
```

```
1,2,3,4,3,2,1,0,-1,-2,
```

[15]:
```
# Enumerate                  # Enumerate Creates a enumerate object which␣
 ↪creates index and element pair from the tuple
list(enumerate(a))
```

[15]:
```
[(0, 1),
 (1, 2),
 (2, 3),
 (3, 4),
 (4, 3),
 (5, 2),
 (6, 1),
 (7, 0),
 (8, -1),
 (9, -2)]
```

[16]:
```
# iter
iter(a) # Returns an iterator object from a tuple
```

[16]:
```
<tuple_iterator at 0x1b78c7d74c0>
```

[17]:
```
# Zip
list(zip(a,a,a))        # Zip aggregates elements from multiple iterables into␣
 ↪tuples.
```

[17]:
```
[(1, 1, 1),
 (2, 2, 2),
 (3, 3, 3),
 (4, 4, 4),
 (3, 3, 3),
 (2, 2, 2),
 (1, 1, 1),
 (0, 0, 0),
 (-1, -1, -1),
 (-2, -2, -2)]
```

1.5 Membership Test

[18]:
```
print("3 in a \t\t:",3 in a)      # Returns true or false if item (3) in the␣
 ↪tuple
print("3 not in a \t: ", 3 not in a)    ## Returns true or false if item (3)␣
 ↪not in the tuple
```

```
3 in a          : True
3 not in a      :  False
```

1.6 Coversions

[19]:
```
print(tuple(a))     # Converting to tuple
print(list(a))      # Converting to List
print(set(a))       # Converting to set ( removes duplicates)
str(a)       # Coverts to string
```

```
(1, 2, 3, 4, 3, 2, 1, 0, -1, -2)
[1, 2, 3, 4, 3, 2, 1, 0, -1, -2]
{0, 1, 2, 3, 4, -2, -1}
```

[19]:
```
'(1, 2, 3, 4, 3, 2, 1, 0, -1, -2)'
```

I hope you found this information helpful! Feel free to save this post for future reference. Let's continue to learn and grow together!

Rajendra Prasad