

```
('Hey', (1, 2), 1, ['you'])
# Create tuple with tuple() constructor.
Numbers = tuple()
Print(numbers)
()
Numbers = tuple([1, 2]) # Takes any sequence as input
Print(numbers)
(1,2)
#### Methods on tuples #####
# Get length of list by using len() method.
Numbers = 5, 8, 8
Print(len(numbers))
```

Get index of an element using the index() method.

Numbers = 5, 8, 8

Print(numbers.index(8))

1

Count occurences of an item in a tuple.

Numbers = 5, 8, 8

Print(numbers.count(8))

2

Eggs = ('hello', 42, 0.5)

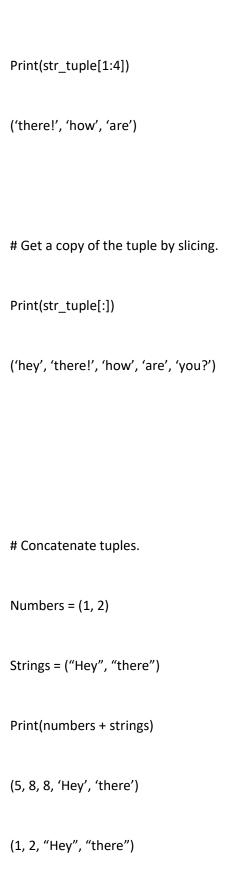
Eggs[0]

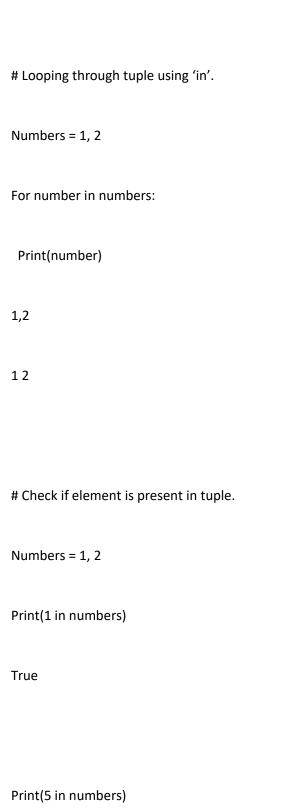
'hello'

Hello

```
Eggs[1:3]
(42, 0.5)
Len(eggs)
3
# Access elements of a tuple by indexing.
Str_tuple = "hey", "there!", "how", "are", "you?"
Print(str_tuple[0])
Hey
Print(str_tuple[len(str_tuple) - 1])
You?
Print(str_tuple[-1])
You?
```

```
# Slicing a tuple.
Str_tuple = "hey", "there!", "how", "are", "you?"
Print(str_tuple[2:])
('how', 'are', 'you?')
Print(str_tuple[:2])
('hey', 'there!')
Print(str_tuple[-3:])
('how', 'are', 'you?')
Print(str_tuple[:-3])
('hey', 'there!')
```





False
Tuple packing.
We are packing two items 1 and 2 into the tuple.
Numbers = 1, 2
Tuple sequence unpacking.
Number of variables used has to be same as the number of items in the tuple.
Unpacking the tuple and assigning its items to x and y.
X, y = numbers
Note that this is also packing the args as a tuple which gets unpacked as the print method's arguments.
Print(x, y)
1 2
LINK:

http://103.53.53.18/mod/hvp/view.php?id=238	;
RESULT:	
The output for the given program is obtained.	