Tuple comprehensions and methods

Comprehension: comprehensions are short cut for creating a sequence object or tuple object.

Let's Try:

```
>>>
>>> l1= [ x for x in range(10)]
>>> l1
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> t1 = (x for x in range(10))
>>> t1
<generator object <genexpr> at 0x7fafa245fc10>
>>>
>>> t1 = tuple(x for x in range(10))
>>> t1
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
>>> t2 = *(x for x in range(10))
SyntaxError: can't use starred expression here
>>> t2 = (*(x for x in range(10)),)
>>> t2
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
>>>
>>> t3 =(*(x for x in range(1,10,2)),)
>>> t3
(1, 3, 5, 7, 9)
>>> t4 =(*(x for x in 'python'),)
>>> t4
('p', 'y', 't', 'h', 'o', 'n')
>>>
```

Now creating another tuple

```
>>>
>>> t5 =(*(x for x in 'PyThOn' if x.islower()),)
>>> t5
('y', 'h', 'n')
>>> t6 =tuple( x for x in 'PyThOn' if x.islower() )
>>> t6
('y', 'h', 'n')
>>> t7 =tuple( x**2 for x in  (1,3,5,7,9 ))
>>> t7
(1, 9, 25, 49, 81)
>>>
>>> t8 = (1,2,3,4,5,4,5,6,7)
>>> t8.count(4)
2
>>> t8.index(3)
2
>>> t8.index(10)
Traceback (most recent call last):
 File "<pyshell#11>", line 1, in <module>
  t8.index(10)
ValueError: tuple.index(x): x not in tuple
>>>
```

Index method gives an error when the element is not found