In [1]:
```
l=[1,2,3,4,5
]
```

In [2]:
```
l.append(6)
l
```

Out[2]: [1, 2, 3, 4, 5, 6]

In [4]:
```
l.count()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-4-8a9795af7861> in <module>()
----> 1 l.count()

TypeError: count() takes exactly one argument (0 given)
```

In [5]:
```
l.reverse()
```

In [6]:
```
l
```

Out[6]: [6, 5, 4, 3, 2, 1]

In [7]:
```
l.sort(
)
l
```

Out[7]: [1, 2, 3, 4, 5, 6]

In [8]:
```
l.count(6)
```

Out[8]: 1

In [9]:
```
help(l.count)
```

```
Help on built-in function count:

count(...)
    L.count(value) -> integer -- return number of occurrences of value
```

In [10]:
```
def hi():
    print "hello"
for i in range(1,5):
    print hi()
```

```
hello
None
hello
None
hello
None
hello
None
```

In [15]:
```
def hi(name):
     print "hello %s" %name
for i in range(1,5):
    hi("me")
```

```
hello me
hello me
hello me
hello me
```

In [13]:
```
def hi(name):
    print "hello %s" %name
    return name
for i in range(1,5):
    n=hi("me")
    print n
```

```
hello me
me
hello me
me
hello me
me
hello me
me
```

In [16]:
```
def add_num(num1,num2):
    return num1+num2
add_num(4,5)
```

Out[16]: 9

In [17]:
```
print add_num('one','two')
```

```
onetwo
```

In [20]:
```
def break1():
    for r in range(1,10):
        print r
        if r==5:
            print r
            break
break1()
```

```
1
2
3
4
5
5
```

In [21]:
```
import math

def is_prime(num):
    '''
    Better method of checking for primes.
    '''
    if num % 2 == 0 and num > 2:
        return False
    for i in range(3, int(math.sqrt(num)) + 1, 2):
        if num % i == 0:
            return False
    return True
is_prime(14)
```

Out[21]: False