In [1]:
```
l=[12,32,1,50]
reduce(lambda x,y: x+y,l)
```

Out[1]: 95

In [4]:
```
l=[12,32,1,50]
reduce(lambda a,b: a if a>b else b,l)
```

Out[4]: 50

In [5]:
```
l=[12,32,1,50]
reduce(lambda a,b: a if a<b else b,l)
```

Out[5]: 1

In [6]:
```
l=range(21)
filter(lambda a: True if a%2==0 else False,l)
```

Out[6]: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

In [7]:
```
l=range(21)
filter(lambda a: True if a%2==1 else False,l)
```

Out[7]: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

In [8]:
```
x=[1,2,3]
y=[4,5,6]
zip(x,y)
```

Out[8]: [(1, 4), (2, 5), (3, 6)]

In [9]:
```
x=[1,2,3,4]
y=[4,5,6]
zip(x,y)
```

Out[9]: [(1, 4), (2, 5), (3, 6)]

In [10]:
```
x=[1,2,3]
y=[4,5,6,5,6,7]
zip(x,y)
```

Out[10]: [(1, 4), (2, 5), (3, 6)]

In [11]:
```
for pair in zip(x,y):#return tuple inside list
    print max(pair)
```

```
4
5
6
```

In [12]:
```
map(lambda pair: max(pair),zip(x,y)) #return list
```

Out[12]: [4, 5, 6]

In [13]:
```
d1={'a':1,'b':2}
d2={'c':3,'d':4}
zip(d1,d2)
```

Out[13]:
```
[('a', 'c'), ('b', 'd')]
```

In [14]:
```
zip(d1,d2.itervalues())
```

Out[14]:
```
[('a', 3), ('b', 4)]
```

In [15]:
```
def switch(d1,d2):
    dout={}
    for d2key,d1val in zip(d2,d1.itervalues()):
        dout[d2key]=d1val
    return dout
switch(d1,d2)
```

Out[15]:
```
{'c': 1, 'd': 2}
```

In [23]:
```
l=[100,200,300,400]
```

In [24]:
```
for item in l:
    print item
```

```
100
200
300
400
```

In [25]:
```
count=0
for item in l:
    print item
    print count
    count+=1
```

```
100
0
200
1
300
2
400
3
```

In [26]:
```
for count,item in enumerate(l):
    print (item,count)
```

```
(100, 0)
(200, 1)
(300, 2)
(400, 3)
```

In [27]:
```
for count,item in enumerate(l):
    if count>=2:
        break
    else:
        print (item,count)
```

```
(100, 0)
(200, 1)
```

In [29]:
```
l=[True,True,True,True]
l1=[True,False,True,False]
all(l)
all(l1)
```

Out[29]: False

In [31]:
```
l=[True,True,True,True]
l1=[True,False,True,False]
any(l1)
```

Out[31]: True

In [32]:
```
complex(12,2)
```

Out[32]: (12+2j)

In [34]:
```
complex("sunamya")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-34-f81cd42507b3> in <module>()
----> 1 complex("sunamya")

ValueError: complex() arg is a malformed string
```

In [35]:
```
complex(32,8)
```

Out[35]: (32+8j)