Introduction

In [3]:
```
# range is an iterator (no elements in it)
```

In [4]:
```
range(10)
```

Out[4]: range(0, 10)

In [5]:
```
range(0,10)
```

Out[5]: range(0, 10)

In [6]:
```
# create a list using iterator
```

In [7]:
```
list(range(0,10))
```

Out[7]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [8]:
```
a = list(range(0,10))
a
```

Out[8]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

loop over one sequence

In [9]:
```
a = list(range(10))
```

In [10]:
```
for i in a:
    print(i,end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

In [11]:
```
for i in range(10):
    print(i,end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

In [ ]:
```
list1 = 'abcde'
```

In [13]:
```
list1
```

Out[13]: 'abcde'

In [22]:
```
for i in list1:
    print(i,end=' ')
```

```
a b c d e
```

In [21]:
```
for i in 'abcde':
    print(i,end=' ')
```

```
a b c d e
```

loop over two sequences

In [6]:
```
list(range(0,10,2))
```

Out[6]:
```
[0, 2, 4, 6, 8]
```

In [7]:
```
list(range(1,11,2))
```

Out[7]:
```
[1, 3, 5, 7, 9]
```

In [ ]:
```
a = list(range(0,10,2))
b = list(range(1,11,2))
```

In [29]:
```
# zip in page 56
```

In [30]:
```
for i,j in zip(a,b):
    print(i,',',j)
```

```
0 , 1
2 , 3
4 , 5
6 , 7
8 , 9
```

In [35]:
```
twotuple = zip(a,b)
twotuple
```

Out[35]:
```
<zip at 0x1ac9ef90fc8>
```

In [36]:
```
# twotuple is an iterator
```

In [37]:
```
# convert it to a list of elements
```

In [38]:
```
twotuple = list(zip(a,b))
twotuple
```

Out[38]:
```
[(0, 1), (1, 3), (2, 5), (3, 7), (4, 9)]
```

list comprehension

In [ ]:
```
# useful for creating a list (see p58)
```

In [40]:
```
a = [i for i in range(20)]
a
```

Out[40]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

In [ ]:
```
# exclude multiples of 3
```

In [41]:
```
a = [i for i in range(20) if i%3 > 0]
a
```

Out[41]: [1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19]

In [ ]:
```
# compare to for loop
```

In [43]:
```
L = []   # empty list
```

In [44]:
```
for n in range(12):
    L.append(n**2)
```

In [45]:
```
L
```

Out[45]: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121]

In [ ]:
```
# or
```

In [47]:
```
L=[n**2 for n in range(12)]
```

In [48]:
```
L
```

Out[48]: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121]

In [ ]:
```
# list of pairs
```

In [50]:
```
L = [(i,j) for i in range(2) for j in range(3)]
```

In [51]:
```
L
```

Out[51]: [(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2)]

In [ ]:
```
# homework: create this list with a for loop
```

lambda function

In [ ]:
```
# for small functions with a single expression   (see p44)
```

In [54]:
```
def f(x):
    return x**2
```

In [55]:
```
f(2.5)
```

Out[55]: 6.25

In [ ]:
```
# lambda function
```

In [56]:
```
g = lambda x:x**2
```

In [57]:
```
g(2.5)
```

Out[57]: 6.25

In [ ]:
```
# lambda function with two arguments
```

In [59]:
```
def f(x,y):
    return x+y
```

In [62]:
```
f(3,2)
```

Out[62]: 5

In [63]:
```
# lambda function
```

In [64]:
```
g = lambda x,y: x+y
```

In [65]:
```
g(3,2)
```

Out[65]: 5

map

In [ ]:
```
# takes a function and applies it to all values in a list/iterator     see p5
```

In [67]:
```
f = lambda x:x*2
```

In [68]:
```
for j in map(f,range(10)):
    print(j,end=' ')
```

```
0 2 4 6 8 10 12 14 16 18
```

In [ ]:
```
# j takes map returned value
```

filter

In [70]:
```
list(range(20))
```

Out[70]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [ ]:
```
f = lambda x: x%3 == 0   # returns True if x is multiple of 3
```

In [71]:
```
for j in map(f,range(20)):
    print(j,end=' ')
```

```
True False False True False False True False False True False False True Fa
lse False True False False True False
```

In [ ]:
```
# filter pass values in range() when the function f returns True, only
```

In [72]:
```
for j in filter(f,range(20)):
    print(j,end=' ')
```

```
0 3 6 9 12 15 18
```

map and lambda

In [ ]:
```
# add two lists
```

In [4]:
```
list1 = [1,2,3]
list2 = [4,5,6]
```

In [5]:
```
iterator3 = map(lambda x,y: x+y, list1,list2)
```

In [6]:
```
list(iterator3)
```

Out[6]:
```
[5, 7, 9]
```

In [7]:
```
# or
```

In [8]:
```
g = lambda x,y: x+y
```

In [9]:
```
iterator3 = map(g, list1,list2)
```

In [10]:
```
list(iterator3)
```

Out[10]: [5, 7, 9]

In [ ]:
```

```