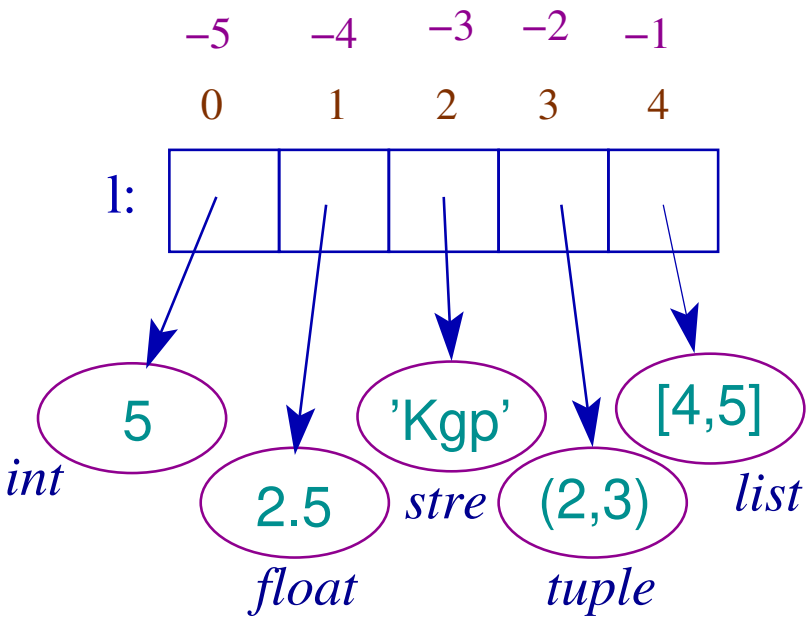
List

List

A list is a finite sequence of object references.
An element of a list can be of any type.

```
>>> l = [5, 2.5, "Kgp", (2,3), [4,5]]
>>> l
[5, 2.5, 'Kgp', (2, 3), [4, 5]]
>>> n = []  # empty list
>>> n
[]
```

Representation and Indices



Mutable Object

A list is a mutable object and its elements can be modified.

```
>>> l
[5, 2.5, 'Kgp', (2, 3), [4, 5]]
>>> l[0] = 7.5
>>> l
[7.5, 2.5, 'Kgp', (2, 3), [4, 5]]
>>> l[4][0] = "IIT"
>>> l
[7.5, 2.5, 'Kgp', (2, 3), ['IIT', 5]]
```

Slicing

Indexing and slicing of a *list* is similar to a *string*.

```
>>> l
[5, 2.5, 'Kgp', (2, 3), [4, 5]]
>>> l[2]
'Kgp'
>>> l[2][1]
'g'
>>> l[4][1]
5 >>> l[1:3]
[2.5, 'Kgp']
>>> l[-1:-3:-1]
[['IIT', 5], (2, 3)]
```

Operations

```
>>> l1 = [1,2,3]
>>> l2 = [2,3,4]
>>> l1+l2
[1, 2, 3, 2, 3, 4]
>>> 2*l1
[1, 2, 3, 1, 2, 3]
>>> len(l1)
3
>>> 2 in l1
True
```

Operations

```
>>> l1
[1, 2, 3]
>>> l1[len(l1):] = [5]    # l1.append(5)
>>> l1
[1, 2, 3, 5]
>>> l1.pop()
5
>>> l1
[1, 2, 3]
```

There are large number of operations available on list.

Note

If n is a positive integer, then `range(n)` is the list $[0, 1, \cdots, n-1]$.

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
>>> range(0)
[]
>>> range(-5)
[]
>>> range(2.5)
__main__:1: DeprecationWarning:
integer argument expected, got float
[0, 1]
```

Note

If `s` is a string, then `s` can be split into list of strings on some character.

```
>>> s = "1 22 13 -14 50 6 7"
>>> s.split(' ')
['1', '22', '13', '-14', '50', '6', '7']
>>> s.split('1')
['', ' 22 ', '3 -', '4 50 6 7']
```

Example

```
# sumList.py : adds elements of an
#                          integer list
l = input("Enter an integer list: ")
sum = 0
for n in l:
    sum = sum + n
print "sum(", l, ") = ", sum
```

Note

What happens if the list has data other than number e.g. `[12, "IIT", 7]`?

Note

```
$ python sumList.py
Enter an integer list: [2, "a", 3]
Traceback (most recent call last):
  File "sumList.py", line 5, in <module>
    sum = sum + n
TypeError: .....
```

Example

```
# sumListExc.py : adds elements
l = input("Enter an integer list: ")
sum = 0
try:
   for n in l:
       sum = sum + n
   print "sum(", l, ") = ", sum
except TypeError:
   print l, "contains non-number"
```

raw_input()

Reads a line from input and converts it to string (excluding the newline character) and returns it.
We may use a prompt with it which is written on the standard output.

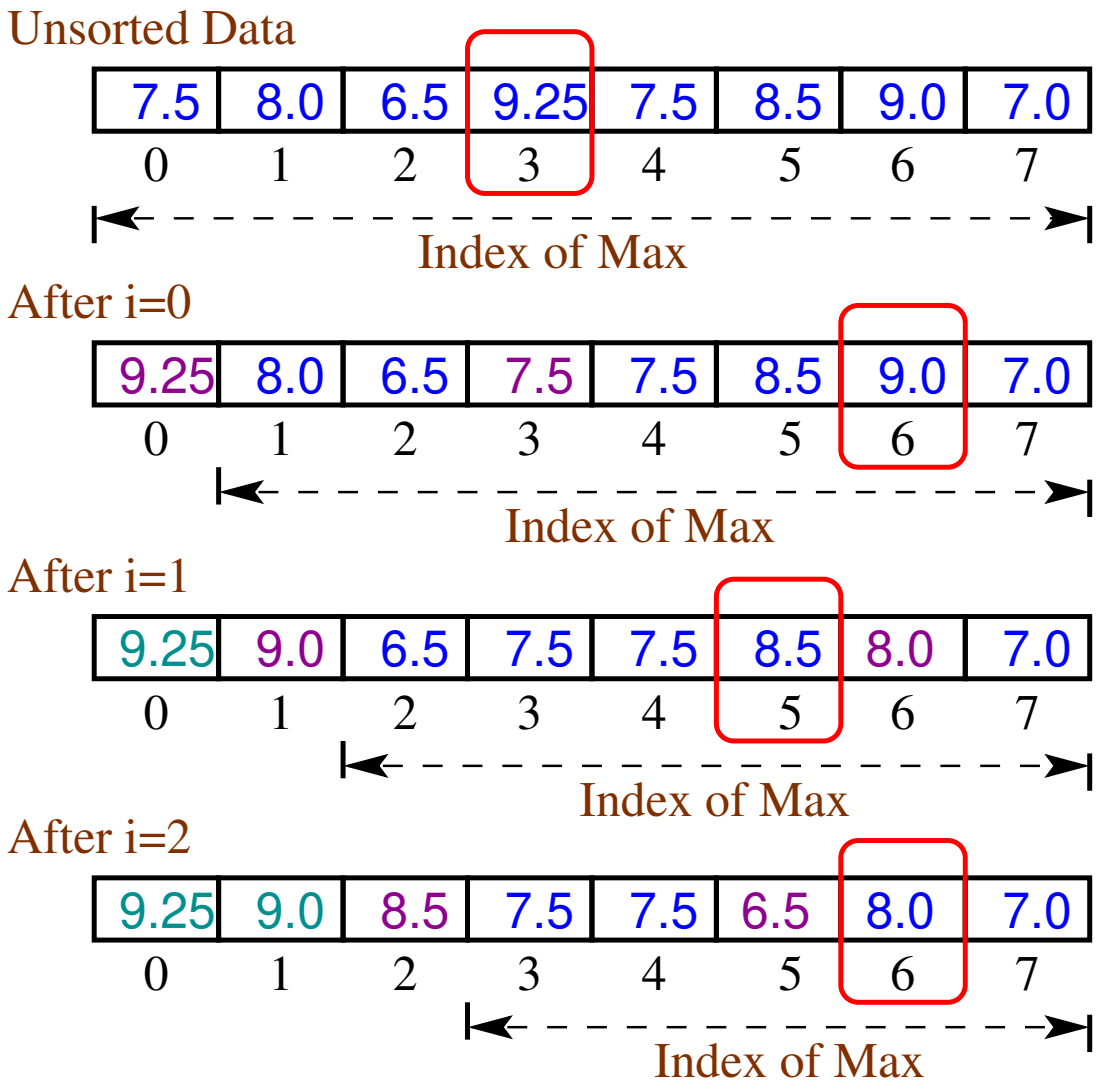
```
>>> s = raw_input("Enter Data: ")
21 34 -7 22 56 7 -15
>>> s
'21 34 -7 22 56 7 -15'
>>> ls = s.split(' ')
>>> ls
['21', '34', '-7', '22', '56', '7', '-15']
```

Selection Sort

The data is stored in a list and we sort them in non-ascending order. Let the length of the list be n.

Selection Sort Algorithm

```
for i ← 0 to n − 2 do
    max ← a[i], maxIndex ← i
    for j ← i + 1 to n − 1 do
        if max < a[j] then
            max ← a[j], maxIndex ← j
        endIf
    endFor
    a[i] ↔ a[maxIndex]      #Exchange
endFor
```


Unsorted Data
7.5 8.0 6.5 9.25 7.5 8.5 9.0 7.0
0 1 2 3 4 5 6 7
Index of Max
After i=0
9.25 8.0 6.5 7.5 7.5 8.5 9.0 7.0
0 1 2 3 4 5 6 7
Index of Max
After i=1
9.25 9.0 6.5 7.5 7.5 8.5 8.0 7.0
0 1 2 3 4 5 6 7
Index of Max
After i=2
9.25 9.0 8.5 7.5 7.5 6.5 8.0 7.0
0 1 2 3 4 5 6 7
Index of Max