python Lab. 2

2012.5.10
• integer number
  – 2 * 2 = 4
  – 5 / 2 = 2
• float number
  – 2.0 * 2.0 = 4.0
  – 5.0 / 2.0 = 2.5
• complex number
  – (3 + 1j) * 3 = 9 + 3j
string (1)

• can be concatenated with + operator
  – word = ‘Help’ + ‘A’
    >>> ‘HelpA’
• can be repeated with * operator
  – Word * 5
    >>> ‘HelpAHelpAHelpAHelpAHelpA’
string (2)

• can be indexed
  – word[4]
    >>> ‘A’
• substring can be specified with slice notation
  – word[0:2]
    >>> ‘He’
  – word[2:4]
    >>> ‘lp’
• string cannot be changed
  – word[0] = ‘x’ -> error
list (1)

• a list of comma-separated values between square brackets
• used to group a number of data types
  >>> a = ['spam', 'eggs', 100, 1234]
• list indices start at 0,
  >>> a[0]
  ‘spam’
• can be sliced
  >>> a[0:2]
  [ ‘spam’, ‘eggs’ ]
• can be concatenated
  >>> a[0:2] + [‘bacon’, 2*2]
• It is possible to change
  >>> a
  a = [‘spam’, ‘eggs’, 123, 1234]
• It is possible to nest lists
  >>> q = [2, 3]
  >>> p = [1, q, 4]
list (3)

- append(x)
  - add an item to the end of the list
- insert(i, x)
  - insert an item at a given position
- index(x)
  - return the index in the list of the first item whose value is x
- len()
  - return the number of elements in the list
**Tuple (1)**

- consists a number of values separated by commas
  
  ```python
  >>> t = 12345, 54321, 'hello!
  >>> t
  (12345, 54321, 'hello!')
  ```

- It is possible to nest tuples

- It is not possible to assign to the individual items of a tuple
  
  ```python
  >>> t[0] = 23456 -> error
  ```
Tuple (2)

• empty tuples are constructed by an empty pair of parentheses
  >>> empty = ()
  >>> len(empty)
  0

• a tuple with one item is constructed by following a value with a comma
  >>> single = 'hello',
  >>> single
  ( 'hello', )
**set**

- is an unordered collection with no duplicate elements
  ```python
  >>> basket= ['apple', 'orange', 'apple']
  >>> fruit = set(basket)
  >>> fruit
  set(['orange', 'apple'])
  ```
- Membership testing
  ```python
  >>> 'orange' in fruit
  True
  ```
- supports operations like union, intersection, difference
dictionary

• is an unordered set of key: value
• is indexed by keys which is immutable type
  >>> tel = {'a':1, 'b': 2, 'c': 3}
  >>> tel['d'] = 4
  >>> tel
  {'a':1, 'c':3, 'b': 2, 'd': 4}
• keys() and values()
  – Return a list of all the keys or values used in the dictionary
  >>> tel.keys()
  [‘a’, ‘c’, ‘b’, ‘d’]
functions

• range()
  – generates lists containing arithmetic progressions
  >>> range(0, 10)

• eval()
  – Evaluate the a string representing a python expression
  >>> a = input()
  >>> a = eval(a)
for a in range(10):
    print a,

The above code acts exactly the same as:

for a in range(0, 10):
    print a,

with 0 implied as the starting point. The output is

0 1 2 3 4 5 6 7 8 9
Problem

• sorting a list
• ex) $a = 10, 3, 9, 1, 7, 6, 4, 5, 2, 8$
hint: bubble sort

6  4  1  9  3
4  6  1  9  3
4  1  6  9  3
4  1  6  9  3
4  1  6  3  9

4  1  6  3  9
1  4  6  3  9
1  4  6  3  9
1  4  3  6  9
1  4  3  6  9