Lists

Say we need to get thirty test scores from a user and do something with them, like put them in order. We would create thirty variables, score1, score2, ……., score30, but that would be very tedious. To then put the scores in order would be extremely difficult. The solution is to use list.

Basics: creating lists here is a simple list.

```
L = [1,2,3]
```

Use square brackets to indicate the start and end of the list, and separate the items by commas.

The empty list: the empty list is []. It is the list equivalent of 0 or ‘ ’.

Long lists: if you have a long list to enter, you can split it across several lines, like below.

```
nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
        32, 33, 34, 35, 36, 37, 38, 39, 40]
```

Input: we can use eval(input( )) to allow the user to enter a list. Here is an example:

```
L = eval(input('Enter a list: '))
print('The first element is ', L[0])
```

```
Enter a list: [5,7,9]
The first element is 5
```

E-mail: hasanein.y.m.alhrabi@uomus.edu.iq

Printing list: you can use the print function to print the entire contents of a list.

```
L = [1,2,3]
print(L)
```

```
[1, 2, 3]
```

Data types: list can contain all kinds of things, even other lists. For example, the following is a valid list:

```
[1, 2.718, 'abc', [5,6,7]]
```

Similarities to Strings

There are a number of things which work the same way for lists as for strings.

- len - the number of items in L is given by len(L).
- in – the in operator tells you if a list contains something. Here are some examples:

```
if 2 in L:
    print('Your list contains the number 2.')
if 0 not in L:
    print('Your list has no zeroes.')
```

- Indexing and slicing – these work exactly as with strings. For example, L[0] is the first item of the list L and L[:3] gives the first three items.

- index and count – these methods work the same as they do for string.
- + and * - the + operator adds one list the end of another. The * operator repeats a list here are some examples:

Expression	Result
`[7,8]+[3,4,5]`	`[7,8,3,4,5]`
`[7,8]*3`	`[7,8,7,8,7,8]`
`[0]*5`	`[0,0,0,0,0]`

The last example is particularly useful for quickly creating a list of zeroes.

- Looping – the same two types of loops that work for strings also work for lists. Both of the following examples print out the items of a list, one-by-one on separate lines.

```
for i in range(len(L)):
    print(L[i])
```

```
for item in L:
    print(item)
```

The left loop is useful for problems where you need to use the loop variable i to keep track of where you are in the loop. If that is not needed, then use the right loop, as it is a little simpler.

Built-in functions

There are several built-in functions that operate on lists. Here are some useful ones.

Function	Description
`len`	returns the number of items in the list
`sum`	returns the sum of the items in the list
`min`	returns the minimum of the items in the list
`max`	returns the maximum of the items in the list

For example, the following computes the average of the values in a list L:

```
average = sum(L)/len(L)
```

List methods

Here are some list methods:

Method	Description
`append(x)`	adds x to the end of the list
`sort()`	sorts the list
`count(x)`	returns the number of times x occurs in the list
`index(x)`	returns the location of the first occurrence of x
`reverse()`	reverses the list
`remove(x)`	removes first occurrence of x from the list
`pop(p)`	removes the item at index p and returns its value
`insert(p,x)`	inserts x at index p of the list

Important note: there is a big difference between list methods and string methods do not change the original string, but list methods do change the original list. To sort a list L, just use L.sort( ) and not L = L.sort(). In fact, the latter will not work at all.

wrong	*right*
`s.replace('X','x')`	`s = s.replace('X','x')`
`L = L.sort()`	`L.sort()`

Other list methods: there are a few others list methods. Type help(list) in the python shell to see documentation for them.

E-mail: hasanein.y.m.alhrabi@uomus.edu.iq

Miscellaneous

Making copies of lists: making copies of lists is a little tricky due to the way python handles lists. Say we have a list L and we want to make a copy of the list and call it M. The expression M = L will not work; you have to do the following in place of M=L:

```
M = L[:]
```

Changing lists: changing a specific item in a list is easier than with strings. To change the value in location 2 of L to 100, we simply say L[2] = 100. If we want to inset the value 100 into location 2 without overwriting what is currently there, we can use the insert method. To delete an entry from a list, we can use the del operator. Some examples are shown below. Assume L=[6, 7, 8] for each operation.

Operation	New L	Description
`L[1]=9`	`[6,9,8]`	replace item at index 1 with 9
`L.insert(1,9)`	`[6,9,7,8]`	insert a 9 at index 1 without replacing
`del L[1]`	`[6,8]`	delete second item
`del L[:2]`	`[8]`	delete first two items

Examples

Example 1: write a program that generates a list of 50 random numbers 1 and 100.

```
from random import randint
L = []
for i in range(50):
    L.append(randint(1,100))
```

We use the append method to build up the list one item at a time starting with the empty list, []. An alternative to append is to use the following:

```
L = L + [randint(1,100)]
```

Example 2: replace each element in a list L with its square.

```
for i in range(len(L)):
    L[i] = L[i]**2
```

Example 3: count how many items in a list L are greater than 50.

```
count = 0
for item in L:
    if item>50:
        count=count+1
```

Example 4: write a program that prints out the two largest and two smallest elements of a list called scores.

```
scores.sort()
print('Two smallest: ', scores[0], scores[1])
print('Two largest: ', scores[-1], scores[-2])
```

Exercises

1. Write a program that asks the user to enter a list of integers. Do the following:
    a. Print the total number of items in the list.
    b. Print the last item in the list.
    c. Print the list in reverse order.
    d. Print yes if the list contains a 5 and no otherwise.
    e. Remove the first and last items from the list, sort the remaining items, and print the result.
    f. Print how many integers in the list are less than 5.
2. Write a program that generates a list of 20 random numbers between 1 and 100.
    a. Print the list.

E-mail: hasanein.y.m.alhrabi@uomus.edu.iq

b. Print the average of the elements in the list.
c. Print the largest and smallest values in the list.
d. Print the second largest and second smallest entries in the list.
e. Print how many even numbers are in the list.

3. Start with the list [8,9,10]. Do the following:
   a. Set the second entry (index 1) to 17
   b. Add 4, 5, and 6 to the end of the list
   c. Remove the first entry from the list
   d. Sort the list
   e. Double the list
   f. Insert 25 at index 3

The final list should equal [4,5,6,25,10,17,4,5,6,10,17]

E-mail: hasanein.y.m.alhrabi@uomus.edu.iq