Worksheet 2 - Lists

Working with Lists

To create a *list* of objects, use square brackets.

Exercise: Create the list **[63, 12, -10, 'a', 12]**, assign it to the variable **L**, and print the list. (*Hint*: Variable assignment in Sage/Python is done with **=**. For example, **a = 3** defines the **a** to be **3**.)

Exercise: Use the **len** command to find the length of the list **L**.

To access an element of the list, use the syntax **L[i]**, where **i** is the index of the item.

Exercise: What is **L[3]**?

Exercise: What is **L[1]**?

Exercise: What is the index of the first item of **L**?

Exercise: What is **L[-1], L[-2]**?

Exercise: Access the last item in **L**.

Exercise: Change **L[3]** to **17**.

Exercise: By typing **L.<tab key>**, you get a list of methods for **L**. Use one of these methods to *append* 17 to the end of **L**.

Exercise: Insert the letter 'b' at index position 2 (do not *change* the element in position 2, but add a new element).

Exercise: Remove the second occurrence of 12 from L.

Note: The above methods modified the list and, importantly, did not return a copy of the list!

The range command

The **range** command provides an easy way to construct a list of integers.

Exercise: Read the documentation (type: **range?** and hit enter or tab). Use it to create the list $[1, 2, \ldots, 50]$.

Exercise: Create the list of even numbers between 1 and 100 (including 100).

Exercise: The **step** argument in the **range** command can be negative. Use **range** to construct the list $[10, 7, 4, 1, -2]$.

Sage (*but not Python!*) includes syntax to simplify creating lists like the above easier.

Exercise: What is the output of the command **[2, 4, .., 100]**?

Exercise: Create the list $[1, 1.5, 2.0, 2.5, ..., 5]$ using Sage's special syntax. Compare this with the output of **range(1,5,0.5)**.

List Comprehensions

Example. We already know how to create the list $[1, 2, \ldots, 25]$:

```
[1,2,..,25]
```

Using a *list comprehension,* we can now create the list $[1^2, 2^2, 3^2, ..., 25^2]$:

```
[i^2 for i in [1,2,..,25]]
```

Exercise.

1. Create lists two lists:

$$x = [1, 2, \ldots, 100],$$

$$y = [1^2, 2^2, \ldots, 100^2].$$

2. Use a list comprehension to construct the list

$$[x_1 + y_1, x_2 + y_2, \ldots, x_{100} + y_{100}].$$

3. Using a list comprehension and the command **sum**, compute

$$\sum_{i=1}^{100} x_i y_i.$$

Project Euler Problem 6

The sum of the squares of the first ten natural numbers is,

$$1^2 + 2^2 + \ldots + 10^2 = 385$$

The square of the sum of the first ten natural numbers is,

$$(1 + 2 + \ldots + 10)^2 = 55^2 = 3025$$

Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is 3025 − 385 = 2640.

Find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum.

Filtering lists with a list comprehension

A list can be *filtered* using a list comprehension.

Example: To create a list of the squares of the prime numbers between 1 and 100, we use a list comprehension as follows.

```
[p^2 for p in [1,2,..,100] if is_prime(p)]
```

Exercise: Use a *list comprehension* to list all the natural numbers below 20 that are multiples of 3 or 5.

Hints:

1. To get the remainder of 7 divided by 3 use **7%3**.
2. To test for equality use two equal signs (**==**); for example, **3 == 7**.

Project Euler Problem 1

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.