Worksheet 3 - First steps towards programming

First steps towards programming.

In this worksheet, you will learn to define functions, write for loops, conditional statements and and continue to learn about lists.

Functions

To define a function in Sage, use the `def` command and a colon after the list of variable names. For example:

```
def is_even(n):
    return n%2 == 0
```

Notice that *body* of the function (the line: **return n%2 == 0**) is indented. The indentation defines the body of the function.

Exercise: Define a function called **square** that returns the square of a number.

If statement

Below is an example of an **if statement**.

```
if n % 2 == 0:
    return True
else:
    return False
```

Notice again how the commands in the first block and the second block of the statement are indented.

Example: The following example defines the **factorial_function**, which takes a number n and returns the product $n(n-1)(n-2)\cdots 1$.

```
def factorial_function(n):
   if n == 0:
       return 1
   elif n == 1:
       return 1
   else:
       return n*factorial_function(n-1)
```

Exercise: Define a function **sign** that returns the sign of a number.

For loop

The following example uses a for loop to create a list of all the numbers between 1 and 1000 that are multiples of 3 and 5.

```
nums = []
for i in range(1,1001):
   if i % 3 == 0 and i % 5 == 0:
       nums.append(i)
print nums
```

```
    [15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210,
    225, 240, 255, 270, 285, 300, 315, 330, 345, 360, 375, 390, 405,
    420, 435, 450, 465, 480, 495, 510, 525, 540, 555, 570, 585, 600,
    615, 630, 645, 660, 675, 690, 705, 720, 735, 750, 765, 780, 795,
    810, 825, 840, 855, 870, 885, 900, 915, 930, 945, 960, 975, 990]
```

Exercise: Recall that the *Fibonacci sequence* is the sequence of numbers that begins with $F_0 = 0$, $F_1 = 1$, and that satisfies the equation $F_n = F_{n-1} + F_{n-2}$ for all $n \geq 2$. Define a function that returns a list of the first **m** terms in the Fibonacci sequence.

Project Euler Problem 2

> Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:
>
> 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
>
> Find the sum of all the even-valued terms in the sequence which do not exceed four million.

Slicing lists

You can *slice* a list to obtain only part of it. The syntax is

> **L[start:stop:step]**

Exercise: Let **L = range(100)**, and try the following **L[0:3], L[:3], L[1:],**

L[1:-1], L[::2].

Exercise. Create a list **L**.

1. Use a slice to obtain the reversal of **L**.
2. Revese the list **L** using **L.reverse()**.
3. What is the difference between these methods of reversing a list?

Project Euler Problem 4

> A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 □ 99.
>
> Find the largest palindrome made from the product of two 3-digit numbers.

Hints:

1. **7%3** returns the remainder of 7 divided by 3.
2. **7//3** returns the integer quotient of 7 by 3.