# Exercise 1

These computational exercises should be completed by **January 6** at **11:59AM**. Solutions should be turned in through the course website.

---

1. **Introduction to Python**

   Browse through the following tutorials: *A Byte of Python*, *A Quick, Painless Tutorial on the Python Language*, *Python Programming*, and the website `www.python.org`, or any of the other tutorials listed on the Blackboard website.

2. **Reading and Writing your Name**

   We've seen how to get python to say `"Hello World!"`. Write a program that prompts the user (you) to enter their name and favorite quality. Have the program output `"Sir <name> the <quality>"`

   (E.g. `"Sir Robin the Brave"` or `"Sir Guido the Pythonista"`).

3. **Numerical Integration**

   One of the simplest tasks a scientific programmer may encounter is numerical integration. The general method is to replace the integral by a sum

   $$\int_a^b f(x)dx \approx \sum_{j=1}^{N} w_j f(x_j), \tag{1}$$

   where the points $x_j$ and weights $w_j$ are chosen to make the approximation as accurate as possible (we will discuss how to evaluate our choices in future sessions). The simplest method is the **rectangle method**, in which we let

   $$x_j = a + \frac{b-a}{N} j \tag{2}$$

   and

   $$w_j = \frac{b-a}{N}. \tag{3}$$

   Write a program that prompts the user to enter $a$, $b$, and $N$, and apply this method to $f(x) = 1$, $f(x) = x$, and $f(x) = x^2$. Test this program for several values of $N$ (e.g. 10, 100, 1000) with $a = 0$ and $b = 1$. To evaluate the sum, use a `for` loop.

4. **Challenges**

   - Modify your integration program to use the **trapezoidal rule** or **Simpson's rule**. For some ideas, check out the **Newton-Cotes Formulas** page at `www.mathworld.com`.
   - Write a program that approximates the **derivative** of a function. For some ideas, check out the **Finite Difference** page at `www.mathworld.com`.