**Finding Numbers in a Haystack**

In this assignment you will read through and parse a file with text and numbers. You will extract all the numbers in the file and compute the sum of the numbers.

**Data Files**

We provide two files for this assignment. One is a sample file where we give you the sum for your testing and the other is the actual data you need to process for the assignment.

* Sample data: <http://py4e-data.dr-chuck.net/regex_sum_42.txt> (There are 90 values with a sum=445833)
* Actual data: <http://py4e-data.dr-chuck.net/regex_sum_902457.txt> (There are 54 values and the sum ends with 572)

These links open in a new window. Make sure to save the file into the same folder as you will be writing your Python program. **Note:** Each student will have a distinct data file for the assignment - so only use your own data file for analysis.

**Data Format**

The file contains much of the text from the introduction of the textbook except that random numbers are inserted throughout the text. Here is a sample of the output you might see:

Why should you learn to write programs? 7746

12 1929 8827

Writing programs (or programming) is a very creative

7 and rewarding activity. You can write programs for

many reasons, ranging from making your living to solving

8837 a difficult data analysis problem to having fun to helping 128

someone else solve a problem. This book assumes that

everyone needs to know how to program ...

The sum for the sample text above is **27486**. The numbers can appear anywhere in the line. There can be any number of numbers in each line (including none).

**Handling The Data**

The basic outline of this problem is to read the file, look for integers using the **re.findall()**, looking for a regular expression of **'[0-9]+'** and then converting the extracted strings to integers and summing up the integers.

**Turn in Assignent**

Top of Form

Enter the sum from the actual data and your Python code below:  
Sum: (ends with 572)

Bottom of Form

Top of Form

Python code:  


Bottom of Form

**Optional: Just for Fun**

There are a number of different ways to approach this problem. While we don't recommend trying to write the most compact code possible, it can sometimes be a fun exercise. Here is a a redacted version of two-line version of this program using list comprehension:

Python 2

import re

print sum( [ \*\*\*\*\*\* \*\*\* \* in \*\*\*\*\*\*\*\*\*\*('[0-9]+',\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.read()) ] )

Python 3:

import re

print( sum( [ \*\*\*\*\*\* \*\*\* \* in \*\*\*\*\*\*\*\*\*\*('[0-9]+',\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.read()) ] ) )

Please don't waste a lot of time trying to figure out the shortest solution until you have completed the homework. List comprehension is mentioned in Chapter 10 and the **read()** method is covered in Chapter 7.