Python lab 2 – Regex in Python

**Python lab 2, 3 and this week’s group practice will help you get ready for later HWs.** Without these practices, HWs will be very confusing. Please work hard during lab time, discuss with your coding partners, and complete the tasks.

# Grading

To make sure that everyone will work on the lab practices and will not get lose in the first few weeks and get very confused when doing HW, this Lab is graded. Please submit the work following the instructions given under each task. **Regular HW is 10pt. For this lab: each question is 2pt. 2pt -- you complete the whole question. 1pt – you completed part of the requirement.**

# Submit:

Put your code (just need to include part of the code if it is longer than 1 page) and your python shell printing results side by side (if you use Python default IDLE). Take a screenshot (you can use Windows->start up menu->type SnippingTools). **Save this screenshot as 1 picture for each task (.png or .jepg) and submit to cougar course**. If you use Pycharm, just get a screenshot showing both the code and the printout results. Please also submit the source code .py file, not the project.

**Student who are done with the lab early: please help others if you can. Read others’ code and explain technical concepts to others will reinforce your understanding. You will understand and remember a topic for a long time if you teach someone how to do it.**

# Task 1: Pattern matching for strings

Please find your answer keys from Week 2’s “Group practice – regular expression”. You will code those regular expressions in Python in this question. (You can decide whether you want to define functions and how to organize your code. It is for your own practice.)

1). Define 10 string variables that follows the requirement given in Week 2’s “Group practice – regular expression”. Some patterns might show up more than once. Some strings might not fit into any pattern. Some strings might fit into multiple patterns. Here is an example of the 10 strings if you want to use:

[“22.11”,“23”,“66.7f”, “123abcde”,“Case44”, “Happy”,”78”, “66.7”, “yes123”,”Book111”]

2). Write down the regular expressions for these patterns (you already have them from last week’s practice)

3). For each string: use if-elif-else and re library to test which pattern matches this string. Note that some strings might fit into multiple patterns.

4). Print out formatted output messages for the **pattern matching** results.

For example:

Str1=”22.11” Test this string against different regular expression patterns until finding a match and print a message out.

“22.11 matches the pattern: A float consists of 1 or more digits before and after decimal point.”

“22.11 matches the pattern: A float with exactly 2 digits after the decimal point”

“23 matches the pattern: An integer”

# **Task 2**: Remove a part of a string

You can use the same .py file from task 1 to complete this question. Write a function that will find the int number at the beginning of a long string and remove it from the long string. Note that the integer might have space or no space following it.

(Technical googling practice – which function can you use to remove part of a string from a long string?)

Sample:

Testing string “22 street” will give output “Found integer 22 at the beginning of this string, starting at index \_ ending at index \_. The rest of the string is: street.”

Testing string “90years” will give “Found integer 90 at the beginning of this string, starting at index \_ ending at index \_. The rest of the string is: years.”