Module 4

From CSE330 Wiki

Jump to: navigation, search

In Module 4, you will learn the fundamentals of Python, a popular scripting language, and Regular Expressions, which give you power in text processing.

This article contains your assignments for Module 4.

Reading

The following articles on the online class wiki textbook contain information that will help you complete the assignments.

- Python
- Regular Expressions

Individual Assignments

Learn Python and Regular Expressions

Up to this point in CSE 330, you have used only one programming language: PHP. In Module 4, you will learn a new language: Python. Before starting this assignment, you should look over the online textbook article about Python. You should be using version 3 (or greater) of Python. Additionally, you may find it helpful to look over the online textbook article about Regular Expressions.

Write some Regular Expressions

Please write regular expressions that do each the following:

1. Match the substring "hello world" in a string.
2. Find all words in an input string that contains three consecutive vowels, regardless of case.
3. Match an input string that is **entirely** a flight code, of the format AA####, where **AA** is a two-letter uppercase airline code, and **####** is a three- or four-digit flight number.

Examples:

```
  Note: text in red is matched by the regular expression. Each line is a separate
test input string.
  Regex 1:  Programmers will often write hello world as their first project with a
programming language.
  Regex 2:  The gooey peanut butter and jelly sandwich was a beauty.
  Regex 3:  AA312
            AA1298
```

```
NW1234
US443
US31344
AA123 extratext
```

Save your regexes to text files – one each.

Baseball Stats Counter

The St. Louis Cardinals are the most legendary baseball team in the national league. In this exercise, you will be creating a Python script that reads box scores from a file and computes the Cardinals' players' batting averages in a particular season.

Tips and Instructions

- You should write a Python script file to solve this problem.
- You should use a regular expression to parse players' names, at-bats, hits, and runs from the input file.
- You may want to create a class to hold and compute information about each player.
- Your file should take one command-line argument: the path to an input file. If no path is given, your program should print a usage message.

Input

Sample input files may be found here: http://classes.engineering.wustl.edu/cse330/content/cardinals/

The top of the file contains a citation, and what follows is an entire season of Cardinals Baseball box scores. Each game starts with a title; for example:

```
=== St.Louis Cardinals vs. Chicago Cubs, 1940-04-19 ===
```

What follows are the performance of each player in that game. The format of each line is as follows:

```
XXX batted # times with # hits and # runs
```

where **XXX** is the player's name (you may assume that the name is unique), and each **#** are integers representing the number of "at bats", the number of "hits", and the number of "runs", respectively.

Output

Your script needs to output all players' batting average across the season. A player's batting average is that player's total hits divided by that player's total at-bats throughout the entire season. For example, Johnny Hopp was at bat 149 times in 1940, and he had 41 hits, so his batting average was 0.275.

Each line in the output should have the format:

```
XXX: #.###
```

where **XXX** is the player's name, and **#.###** is the player's batting average, **rounded** (not truncated) to three decimal places.

The players should be sorted by batting average, with the highest batting average on top. If two players have the exact same batting average (before rounding occurs), the order between those two players is unimportant.

For example, the correct output for the 1940 season is:

```
Pepper Martin: 0.316
Walker Cooper: 0.316
Johnny Mize: 0.314
Ernie Koy: 0.310
Enos Slaughter: 0.306
Joe Medwick: 0.304
Terry Moore: 0.304
Joe Orengo: 0.286
Jimmy Brown: 0.280
Marty Marion: 0.279
Don Gutteridge: 0.276
Johnny Hopp: 0.275
Creepy Crespi: 0.273
Mickey Owen: 0.265
Bill DeLancey: 0.250
Don Padgett: 0.242
Stu Martin: 0.238
Eddie Lake: 0.222
Hal Epps: 0.214
Lon Warneke: 0.209
Harry Walker: 0.185
Max Lanier: 0.179
Bill McGee: 0.178
Carl Doyle: 0.174
Mort Cooper: 0.157
Clyde Shoun: 0.145
Carden Gillenwater: 0.130
Bob Bowman: 0.067
```

Grading

We will be grading the following aspects of your work. There are 50 points total.

Assignments (including code) must be committed to Bitbucket by the end of class on the due date (commit early and often). Failing to commit by the end of class on the due date will result in a 0.

Upload your baseball code and a file with your Regular expressions.

1. **Regular Expressions (15 Points):**
   - **Your regex files should contain your regex (with no delimiters) and nothing else.**
   - Regular Expression 1 Correct and saved in a file named `regex1.txt` (5 points)
   - Regular Expression 2 Correct and saved in a file named `regex2.txt`(5 points)
   - Regular Expression 3 Correct and saved in a file named `regex3.txt`(5 points)
2. **Baseball Stats Counter (35 Points):**
   - Solution is written entirely in Python and saved in a file named `baseball.py` (8 points)

     Note: Failing to write your code in Python 3 will result in losing, at a minimum, points for this category.
   - Correct usage of one or more regular expressions to parse and extract data from each line of the input file (8 points)

     Note: You should not be using `str.split` to extract data.
   - Script prints a usage message if a command line argument is not present (4 points)

     Note: For an example of a usage message, see this link.
   - Output is correct for all test cases (15 points)

     This includes sorting and rounding.

Retrieved from "https://classes.engineering.wustl.edu/cse330/index.php?title=Module_4&oldid=2541"

Categories:

- Module 4
- Modules

Navigation menu

Views

- Page
- Discussion
- View source
- History

Personal tools

- Log in

Navigation

- Table of Contents
- Recent changes
- Random page

Search

Go Search

Tools

- What links here
- Related changes
- Special pages
- Permanent link
- Page information

Print/export

- Create a book
- Download as PDF
- Printable version

- This page was last modified on 18 March 2018, at 23:56.
- Privacy policy
- About CSE330 Wiki
- Disclaimers