Lesson Plan

# Lesson 07: Getting Data from Files

|  |  |  |  |
| --- | --- | --- | --- |
| **Lesson Overview** | | | |
| **Lesson Length** | 3 hours (180 minutes) | | |
| **Overview** | This session will introduce students to the methods of loading local data into Python. | | |
| **Objectives**   * **Domain:** Cognitive * **Level:** Applying | Using conditionals, loops, Python dictionaries, local data, the CSV Library, and the Glob Library, students will be able to:   * TLO 1: Examine the implications of using computation to solve a problem   + ELO 1.1: Discuss best practices for using computation to solve a problem   + ELO 1.2: Suggest types of problems that can be solved through computation   + ELO 1.3: Show how computation can solve a problem * TLO 2: Recognize key computer science concepts   + ELO 2.4: Recognize how queries operate * TLO 3: Demonstrate the ability to build basic scripts using Python scripting language   + ELO 3.1: Use various data types and structures in Python Scripting   + ELO 3.2: Collect data using Python scripting   + ELO 3.3: Extract data using Python scripting   + ELO 3.4: Develop advanced data structures using Python scripting | | |
| **Instructional Methods** | Informal lecture, demonstration, guided discussion, practical exercises | | |
| **Assessment Strategies** | Informal: Guided and practical exercises  Formal: N/A | | |
| **Materials and Equipment Needed** | Required:   * SBU * Jupyter Notebook * Python   Optional:   * N/A | | |
| **Background Resources** | Resource:   * NGA SME * Technical facilitators   Subject matter/content questions may be referred to:   * Jeremy Debrow, Course Manager   jeremy.r.debrow@nga.mil  jeremy.r.debrow@coe.ic.gov  National Geospatial-Intelligence College (NGC) HDNPE Branch  Unclassified: 571-557-9683 | | |
| **Comments** | ELOs 1.1, 1.2, and 1.3 are meant to be covered throughout the entire lesson (informal lecture and assessment). Instructors will be expected to facilitate classroom discussion that identifies problems best suited to be solved computationally, best practices for solving those problems, and potential solutions. | | |
| **Lesson Sequence** | | | |
| **Lesson Topic** | **Instructional Method** | | **Time**  **(mins)** |
| Introduction | Informal lecture, guided discussion | | 05 |
| Review | Informal lecture, guided discussion, demonstration | | 10 |
| Lesson: Getting Tabular Data | Informal lecture, guided discussion, demonstration | | 30 |
| Assessment | Guided exercise, practical exercises | | 130 |
| Conclusion | Informal lecture | | 05 |
| **Lesson Outline** | | | |
| **Introduction** | | | |
| Introduction | * **Attention** (to be personalized by instructor) * **Motivation** (to be personalized by instructor) * **Overview** (to be personalized by instructor)   + Learning objectives   + Lesson topics/main points * **Rules of Engagement** (to be personalized by instructor) | | |
| **Body** | | | |
| ***Lesson Topic*** | ***Main Points/Notes*** | ***Personalization*** | |
| Review | **Informal Lecture/Guided Discussion/Demonstration (10 minutes)**  **7.1. Objectives**  **7.2. Overview**  *(Facilitator Note: Have students load the U\_CSCI2011\_L07\_Get\_Data\_Files\_SG\_V3.0.ipynb file to begin the session.)*  *(Facilitator Note: Refer back to Lesson 1 and relate the four steps of problem-solving using Computational Thinking (Decomposition, Pattern Recognition, Abstraction, & Algorithm Design) to lessons, exercises, examples, student questions/comments, etc., as appropriate throughout this lesson.)*  **7.3. Review**  *(Facilitator Note: See instructor’s notebook for instructor guidance for discussion points and guided exercises.)*   * 7.3.1. For Loops |  | |
| Lesson: Getting Data From Files | **Informal Lecture/Guided Discussion/Demonstration (30 minutes)**  **7.4. Lesson: Getting Data From Files**   * 7.4.1. Reading and Writing Plain Text   + 7.4.1.1. The open() Function   + 7.4.1.2. Read a Text File   + 7.4.1.3. Write a Text File * 7.4.2. Getting Data with the CSV Library   + 7.4.2.1. Using .DictReader() to read a CSV File   + 7.4.2.2. Using .DictWriter() to write a CSV File * 7.4.3. The Glob Library: How to Find and Read Multiple Files   *(Facilitator Note: Have the class enter and execute the code in the notebook and discuss the questions below.)*   * + 7.4.3.1. Combining the Glob and CSV Libraries |  | |
| Assessment | **Guided Exercise/Practical Exercise (130 minutes)**  *(Instructor Note: Facilitators must maintain a proper pace ensuring all essential guided and practical exercises are performed and reviewed. The demonstration at the end of this lesson requires ~20 minutes to perform. )*  *(Instructor Note: All assessments should incorporate a facilitator directed discussion on computational thinking techniques as they relate to the assigned problem. Utilize student handouts, performance support tools, or projected code cells to capture facilitator/student discussion.)*  **7.5. Guided Exercise: Data Exploration and Engineering**   * Example FlightRadar 24 ADS-B Flight Track Dataset – look at data files in data/fr24/ folder (see notebook).   *(Facilitator Note: The file was imported as a comma separated value file. However, sometimes Excel changes the data. So when you are troubleshooting your code, be sure to look at the raw data and not an Excel formatted version of the data.)*   * Step 1. Read in the data * Step 2. Split the Position column into two separate columns for Latitude and Longitude * Step 3. Cast Latitude, Longitude, Altitude, Speed, and Direction as numeric data types * Step 4. Add a column to identify what file the data came from (there are many files in this data set)   **7.6. Practical Exercises**  *Facilitator Notes:*   * *Refer back to Lesson 1 and relate the four steps of problem-solving using Computational Thinking (Decomposition, Pattern Recognition, Abstraction, & Algorithm Design) as appropriate throughout these exercises.* * *The practical exercises deemed most important due to content and/or a cumulative result, which should be completed first in the interest of maximum training value in relation to time are Practical Exercises 1, 2, 3, and 4. Ensure you go over the exercise solutions and (as necessary) the processes to arrive at the solutions with the students.* * *Follow-up questions are designed to be asked by the facilitators individually as each student completes the task and has it looked at by a facilitator.*   **7.6.1. Practical Exercise 1: Use Glob to Create a List of Filenames**   * Problem 1: Use the Glob library to create pull in all of the filenames of the csv files in the data/fr24/ folder. * Problem 2: What type of data structure did you just create? Have you read in the contents of any of the files yet?   **7.6.2. Practical Exercise 2: Read in Multiple Files in a Loop**   * Using the list of filenames that you created in Practical Exercise 1, iterate over that list and read in each file. Add all the data to a single data structure.   HINT: Keep in mind the difference between .append() and .extend() when you are adding to the data structure.  **7.6.3. Practical Exercise 3: Analyze FR24 Data**   * Problem 1: The Callsign key holds a value that identifies individual aircraft. How many data points are associated with each callsign? Create a new dictionary where the keys are the callsigns from the data set and the values are a count of how many rows contained that callsign. HINT: Use a for loop to loop through each row. * Problem 2: Using the dictionary that you just created, find the callsigns that appear in more than 1500 rows. Add those callsigns to a new list. How many are there? HINT: The answer is 39 (your list should be 40 items long, but one of the items is the empty string (''))   **7.6.4. Practical Exercise 4: Personnel Data**   * Problem 1: Read in the following file using the CSV Library. * Problem 2: For each row in the data you just read in, create a new entry called 'Area Code' that stores the 3-digit area code from the phone number in that row. HINT: Do not create an entirely new data structure for this exercise; we are simply adding a column to our existing CSV data. * Problem 3: Create a new dictionary where the area codes are the keys and the values are a count of the number of times that area code appears in the data set.   **7.6.5. Practical Exercise 5: Netflix Data**   * Problem 1: Read in the following file using the CSV Library. * Problem 2: By looping through the list of dictionaries you just created, create a new dictionary that maps a release year (e.g. 2004) to the number of titles in the data set that were released in that year. The keys in your new dictionary should be the years and the values should be the count of titles released in that year. * Problem 3: Write a function that takes a release year (int or str) as its only input, and returns a set of all release titles from the data set that were released in that year. * Problem 4: Write a function that takes a user rating score (int) as input and outputs a set of release titles that have a score greater or equal to it. HINT: A release's rating can be found in the key called 'user rating score'. * Problem 5: [Challenge] Combine the functionality of the two functions you wrote above. Write a function that takes a user rating score (int) and a year (int) as inputs and outputs a set of release titles from the given year that have a score greater or equal to the given rating.   **7.6.6. Practical Exercise 6: Resume Analysis**   * Problem 1: Read in the following file using a with open() block and .read(). * Problem 2: Now loop through the resume\_filenames list below. As you loop, create a new dictionary. The key for each entry should be the file name without the path and extension (e.g. '01'), and the value should be the full resume text. * Problem 3: Using input(), ask the user for a keyword to search for. Then, by looping over the dictionary you just created (.items()), search for and print out the IDs of all the resumes containing the input keyword. * Problem 4: Edit the code from the last section to take multiple keywords from the user, separated by commas. Print out resume names that contain all the keywords. HINT: Strip whitespace from the beginning and end of each keyword.   **7.7. Appendix** |  | |
| Administrative Notes | N/A |  | |
| **Assessment** | | | |
| ***Assessment Type*** | ***Instructions/Prompts/Notes*** | | |
| Guided Exercise | See the facilitator notes located above for additional guidance. All exercises will be conducted inside the Jupyter Notebook lesson file. | | |
| Practical Exercise | See the facilitator notes located above for additional guidance. All exercises will be conducted inside the Jupyter Notebook lesson file. | | |
| **Conclusion** | | | |
| Conclusion | * **Final Summary** (to be personalized by instructor)   + Review learning objectives   + Review lesson topics/main points * **Remotivation** (to be personalized by instructor) * **Closure** (to be personalized by instructor) * **Next Lesson Introduction** (to be personalized by instructor) | | |