| **Session Objectives** | * Learn how to store, modify, and access data in a text file from a python program. * Recap and Recall different file paths * Understand exceptions and error handling * Use of json modules |
| --- | --- |
| **Key Points** | * open() function * read() method * Relative File Paths * Absolute File Paths * Try-except blocks * Exceptions * json.dump() * json.load() |
| **Assessment** | * Via practical challenges - see task sheet below |
| **Instructor Prep** | * **Note**: These session plans include sections where long explanations are provided simply because it is important to ensure that these tricky concepts are communicated as clearly as possible. However, it is not expected that the instructor will recite these verbatim, feel free to use your own refined and well-honed approach as long as the learning point is covered. The provided narrative is always available if needed. * **Learning and delivery** can be more effective if resources and tasks are personalised. If time permits, feel free to update slides with your own examples such as replacing cat examples with your own pets, or anything else. |
| **Materials** | * [Files and Exceptions Powerpoint](https://docs.google.com/presentation/d/12fpKP7cXjF0v1WLHzUQwFEZlSJoZqHbOi0_Ljw9RQkY/edit?usp=sharing) * [Files and Exceptions Task Sheet](https://docs.google.com/document/d/1Crq75GWD6bwJsCn43DrNps1KZWc3m8xKuTtCV2HCZps/edit?usp=sharing) * Instructor Only: Files and Exceptions Task Solutions |

| **Time** | **Activity** |
| --- | --- |
| 5 minutes | **Slide 2: Learning Objectives**   * **Read from the slide.** |
| 10 minutes | **Slide 3-7: Reading From a Text File**   * **Slide 3: Read from slide.** * **Slide 4: Read from slide, and then walk through the code on the slide.** Explain that the text file is stored separate to the python file. Run the code so learners can see functionality. * **Slide 5: Read from the slide, then go through the code line by line, emphasizing the purpose of the open() function and its syntax for usage.** * **Slide 6: Read from the slide, then go through the code with the learners, describing the purpose of the read() method and its syntax for usage.** * **Slide 7: Read from the slide.** |
| 10 minutes | **Slide 9-12: File Paths**   * **Slide 9: Read from the slide,** then discuss the code examples. One example shows the use case of Linux and the other with a WIndows operating system. Explain that one will need to change the slashes in order to use a Linux or Windows file path. * **Slide 10: Read** from the slide to discuss an absolute file path. The next couple slides will show examples of how to utilize this file path in Linux and Windows operating systems. * **Slide 11:** Example of an absolute file path usage on a Linux/MasOS operating system. Go through the code, remind them about the slashes. * **Slide 12:** Example of an absolute file path on a Windows operating system. Go through the code and show the differences between Linux/MacOS and Windows. |
| 10 minutes | **Slide 13-14: Reading Line By Line**   * **Slide 13: Read from slide** * **Slide 14: Show code on slide and the output.** Recap and Recall the use of a for loop, then explain its purpose within the sample code. Once explained, run the code to show functionality. * **Slide 15:** Read from slide, then run the code to show functionality for making lists from our file |
| 15 minutes | **Slide 16: Writing to an empty file**   * **Walk through the code on the slide line by line. Explain the following points:** * *‘filename = ‘foods.txt’* assigns the ‘foods.txt’ file to the variable ‘filename’ * *‘With open (filename, ‘w’) as file\_object’* opens the file stored in ‘filename’ in write mode (‘w’). Explain that there are different modes in which we can open a file with the open() function *(‘r’ = read, ‘w’ = write, ‘a’ = append, ‘r+’ = read/write, ‘w+’ = write/read, ‘a+’ = append/read)*. Run the sample code located in the presenter comment section of the slide to demo each of the modes, explain the syntax and purpose of each mode. * **Slide 17: Writing Multiple Lines -** Show code running and explain that multiple lines can be written in one ‘with block’. |
| 10 minutes | **Slide 18-19: Exceptions**   * **Slide 18: Read from slide** and ask if they can name any errors they have seen in the past. Recap and Recall the different errors that can occur when working with Python. * **Slide 19: Read from slide,** then show sample codes running to demo each of the errors (located in the presenter notes section of the powerpoint). |
| 10 minutes | **Slide 20: Try-Except Blocks**   * **Slide 20: Read from slide.** * **Slide 21: Read from slide**, then walk through the code and explain its function and syntax. When finished, run the code. |
| 5 minutes | **Slide 22: Tips for Error Handling**   * **Read from the slide and expand on each point. Some examples of what to say:** * Be specific in exception handling: specific exceptions like ‘FileNotFoundError’ can be caught and handled differently compared to another error message. * Print meaningful error messages: Error messages show what went wrong in the code and which file encountered an issue. Make sure the error messages make sense and have a purpose. * Keep exception handling simple: Do not overcomplicate, make sure exception handling is simple and specific to operations where errors might happen. * Test exception handling: exception handling makes sure that if there is an issue with one operation, other operations can still be attempted after each operation. |
| 5 minutes | **Slide 23-24: Storing Data**   * **Slide 23: Read from slide.** * **Slide 24: Read from slide.** |
| 10 minutes | **Slide 25-26: json.dump() and json.load()**   * **Slide 25: Explain the purpose of json.dump() and json.load(). Say,** * *“json.dump() and json.load() are both functions from the ‘json’ module in Python for working with JSON data. json.dump() is used to save Python data as JSON in a file and json.load() is used to read JSON data from a file and convert it back into Python data.* * **Walk through the code on the slide with the learners, line by line and explain what the purpose of each line is and its syntax.** * **When finished running through the code, run the code to show its functionality.** * **Slide 26: Explain that data can also come from user generated input.** Refer to the code on the slide and walk through each line, explaining the functionality. Then run the code in an editor. * **Optional:** also explain that the data stored in cat.json can be accessed from other programs. Demonstrate this by creating a separate python file with another code, then access the content from the cat.json file. |
| 60 minutes | **Slide 25 - Hands-On Challenges**   * **Share link** to task sheet, instruct learners that they now have 45 minutes to work through the task sheet. Once the time is up they should take a screenshot/snip of their last completed challenge and submit it on Canvas. * **Open breakout rooms** - instructor to select number/mix |