

Which Countries Export...?

Weather CSV Problem

- Hottest Day in a Year:  
Comma Separated Values

2 min
- Converting Strings to  
Numbers

4 min
- Maximum Temperature:  
Developing an Algorithm

5 min
- Java for Nothing—null:  
When You Don't Have an  
Object

4 min
- Maximum Temperature:  
Translating into Code

4 min
- Maximum Temperature:  
Testing Code

3 min
- Maximum Temperature  
from Multiple Datasets

5 min
- Maximum Temperature  
Refactored

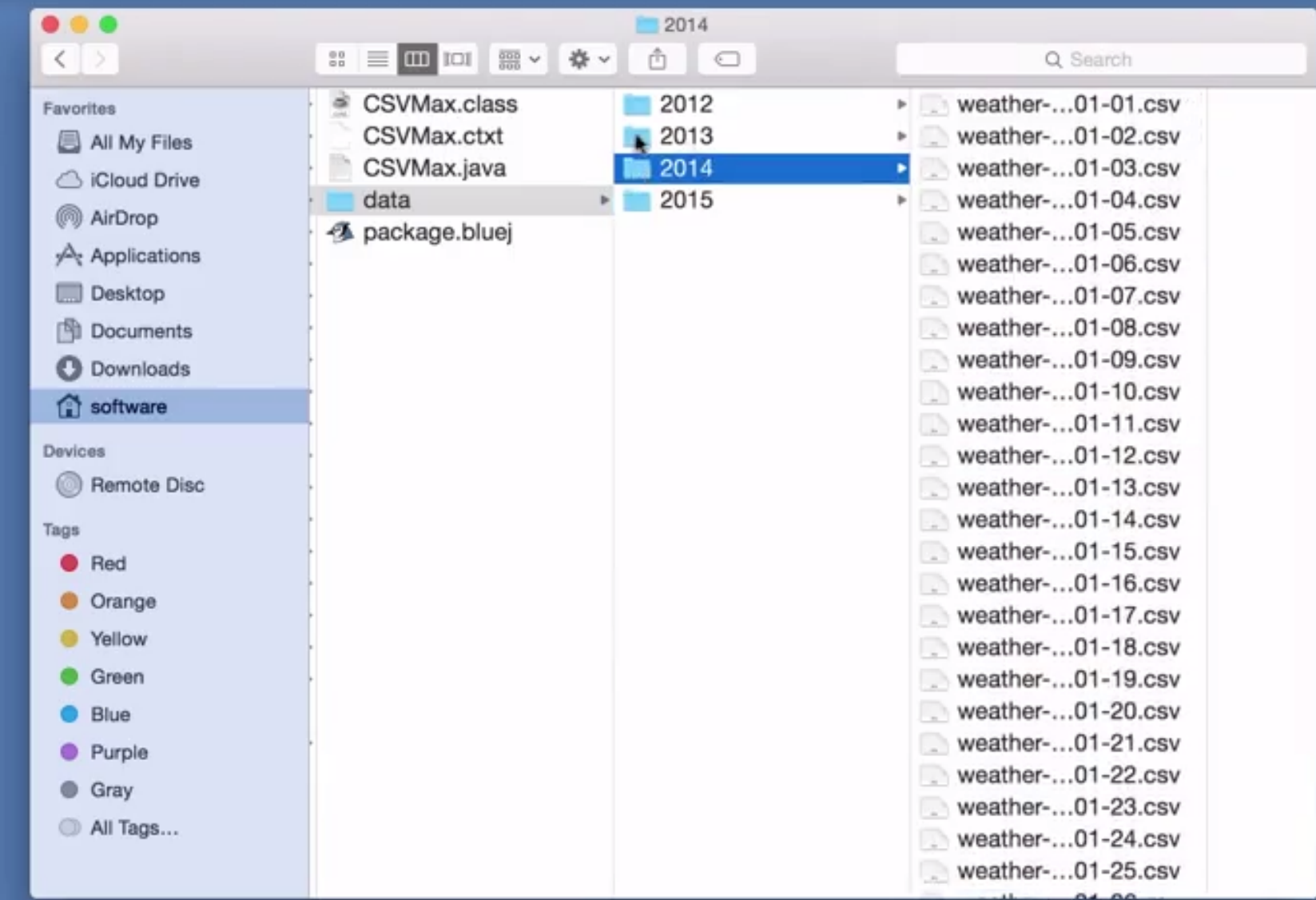
4 min
- CSVMax: Summary

41 sec
- Programming Exercise:  
Parsing Weather Data

10 min
- Practice Quiz:  
Weather Data

9 questions

Review



of those files for each of the years.

Maximum Temperature: Testing Code

Have a question? Discuss this lecture in the week forums.



Interactive Transcript

Search Transcript

English

0:03

Let's take a quick at how the data is organized on the file system and what an actual file looks like. So we have a data folder and inside of that data folder it's organized into years and then within each year we have, a weather file for each day. So there's 365 or 6, depending on if it's leap year, [of those files for each of the years](#). So that's how it looks on the file system. And then here's how an individual file looks, so in this case I've chosen January 1st, 2015 as the one that I'm gonna test that, the file I'm gonna test. And you can see we've got the column information up top and we've got the column data for each of those. We have two files loaded up in a spreadsheet because CSV files are easily represented as spreadsheets. So this is a nice way to visualize them. The other method that I've already included in the CSV Max class, is testHottestInDay, so we can test our code to make sure that it's right. First line in there creates a file resource already set to January 1st, 2015. The second line calls our hottest hour on file method that we just wrote, passing it the parser that we created for that particular data set. And then that returns us the CSV record that is the largest one. And then we print out that hottest temperature and the time that it appeared at just to see when that was, and that's gonna give us a sense of whether or not a program is correct. So, I'm gonna compile our code, and I'm gonna come over to the Bluej environment and create a new CSVMax object.

1:49

And then call my test hottest day. And it says the hottest temperature was 51.1 at 2:51 PM. And if I go over to my data file, and I look at the temperatures, I see that they are in the 20s and the 30s, 50, 51.1, back down the 50, back to the 40s, and 42.1. So, indeed, the hottest temperature was 51.1 occurring at 2:51 PM. Let's also check the 2nd of January just to try it on a second data file to see how we're doing. So I'm going to come over and change that from January 1st to January 2nd. I'm going to recompile. I'm going to go ahead and create a new CSVMax, and call test hottest day. This time it thinks it was 54 at 12:51 PM. And again, if I come over and I look at my temperature. I'm in the 40s.

2:52

I'm now in the 50s, 51.1. I have 54s and then back down to the 40s. You'll notice that there was actually 4 times when it was recorded as being 54 degrees. And we've recorded the first time that that occurred, 12:51 PM, as opposed to the last time that that occurred, 3:51 PM. So now that I've tested it on two separate files, and I've seen that it worked for those. I feel reasonably confident that my code is correct, but I encourage you to test it on more files, and try it out on your own and see what you get.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Would you like to [help us](#) [translate](#) the transcript and subtitles into additional languages?