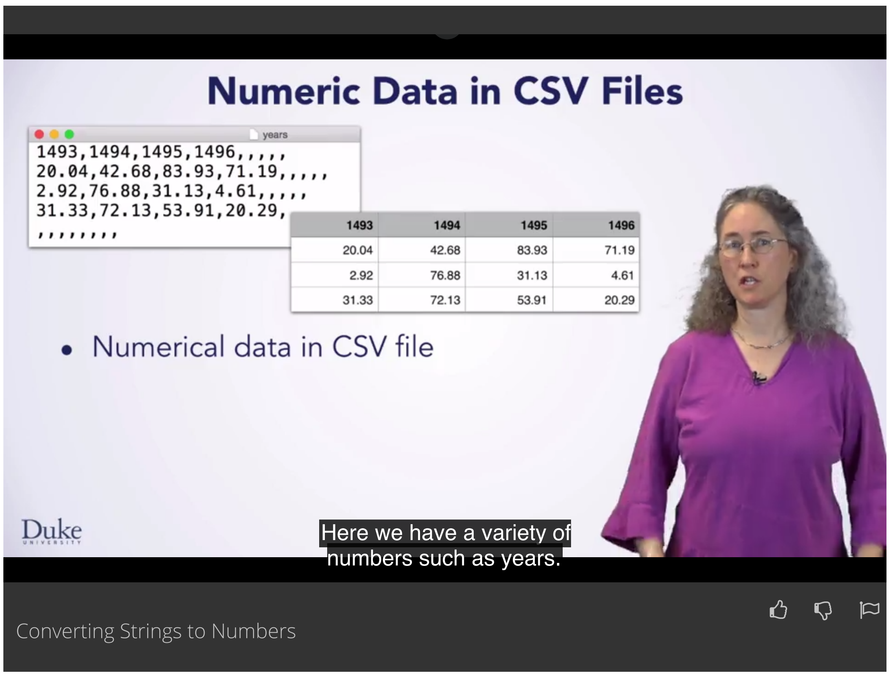
Which Countries Export...?

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



Converting Strings to Numbers

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

Interactive Transcript

Search Transcript

English

0:03
Welcome back. As you work with data in CSV files you may encounter numerical data such as this. Here we have a variety of numbers such as years. However, the libraries you use to read a CSV file will read this data in as a string, the sequence of characters 1, 4, 9, 3, not the integer 1,493. If you want to manipulate this data numerically adding values, finding the largest, or some other task, you will want to work with the data as an int or a double. So what would happen if you tried to write a line of code like this, int value = row.get("year");? When you try to compile it, you would get an error message here, that the types are incompatible. You're trying to assign a string to an int. Let us look a bit more at this error.

0:53
If you look at the documentation for the CSV reader, which we show here, you will see information about the get method. In particular, you can see that its return type is string. Because that method returns a string, this entire expression, row.get("year") has type string. Java knows that it will be a sequence of characters. However, that is not the same type as int, which is the type we declared value to be.

1:20
This raises two questions we will answer for you.

1:23
First, why is this a problem? Sometimes Java can automatically convert between two types, so why can't it convert a string to an int? It seems like if we have the string 1493 Java could just figure out that that should be the number 1493. However, it cannot do this in this case. The second question is, of course, how do you fix it? To see why Java cannot automatically convert a string to an int consider what would happen if the string did not have numerical digits in it. For example, suppose our CSV reader read the string hello instead. What would you expect Java to do for this line of code? The rules of Java require the compiler to reject this line of code because there is no guarantee that you can meaningfully turn any string into a number.

2:12
Another consideration is that the string 1493 has a very different representation from the number 1493. Even though everything is a number, the type of a piece of data describes how that number is represented and interpreted. We will not delve into the details of the representational differences here but it takes some work to convert between the two.

2:35
Accordingly we would have to execute some code to perform the conversion algorithm. Such an algorithm would examine the digits in the string and compute the integer that they represent.

2:47
We are not going to write this algorithm ourselves, because it's already built into Java. You can just call Integer.parseInt to convert a string to a number. Even though it's built in, you still have to call it explicitly to tell Java that that is what you want to do. Here you can see an example of how you can use the Integer.parseInt method. You pass in the string you want to convert and it returns the integer value.

3:16
If you need to work with numbers that have fractional parts, you can use Double.parseDouble, which will accept a string like 42.56 and convert it into the corresponding value of type double. The type you use to work with non-integer numbers. For either of these methods, the string could be invalid. It could be something that does not represent a number. For example, if you passed hello to Integer.parseInt, the method cannot turn hello into an integer. So what happens?

3:47
The method throws an exception, which indicates that an error has occurred. If your program throws an exception, then it will crash. There are ways to make your program handle the exceptions, specifying what to do instead of crashing. Well, we are not going to go into that more advanced topic here. So, now you know that you cannot just assign a string to an int. And that if you need to convert a string to an int, you should use Integer.parseInt. Likewise, for real numbers you have learned to use Double.parseDouble. Now you are ready to manipulate CSV data with numbers in it. Thank you.

Downloads

- **Lecture Video** mp4
- **Subtitles (English)** WebVTT
- **Transcript (English)** txt
- **Lecture Slides** pdf

Would you like to help us translate the transcript and subtitles into additional languages?