CS 240: JSON Introduction Transcript

[00:00:00] This video will introduce the JSON data format, which is a really important thing for programmers to know about.

Start visual description. The professor demonstrates the importance of the JSON data format for programmers, explaining how it is crucial for handling data in modern software applications. End visual description.

[00:00:10] So to give this some context, most organizations have lots of data and most software applications deal with some kind of data.

[00:00:22] And back in the old days, different software applications would store their data in proprietary formats.

[00:00:30] For example, Microsoft Word would store its documents in one format and WordPerfect would store its documents in a different format.

[00:00:39] And even though they were very similar kinds of documents, the data formats that they were stored in were totally different.

[00:00:45] And so those were kind of the bad old days where everybody had their own proprietary data formats.

[00:00:51] It was really hard to share data with um bet between applications or between people because the formats just weren't standard.

[00:00:58] And then when the internet came around um you know, 1993 ish, it became really more important than ever to create standard data formats that people could use to represent their data so that their data could be easily shared with, with other people with other applications with other organizations.

[00:01:17] And so that's where data formats like XML and JSON were born is to solve this problem of having standard data interchange formats.

Start visual description. The professor discusses the emergence of standard data formats like XML and JSON, highlighting their role in facilitating data sharing across different applications and organizations. End visual description.

[00:01:29] Now, in this class, we focus on the JSON format because it's, it's probably used more commonly for new applications than XML.

[00:01:38] But the original format that kind of filled this need was XML.

[00:01:42] And it would be well worth your while to, to go learn about XML.

[00:01:46] It's a very simple format and you'll run into it almost for sure in your uh your work at some point.

[00:01:53] So it would be worth your while to do that. But in this class, we're going to focus on the JSON format.

[00:01:58] And so that's where we're going to head next now to give you a sense of what Jason looks like.

[00:02:06] I want to show you a file.

[00:02:13] So here in Intel, I've opened a JSON file.

Start visual description. The professor opens a JSON file containing a catalog of CDs, showing the structure of the JSON data with most of it collapsing initially. End visual description.

[00:02:16] So it contains uh a catalog of C DS.

[00:02:19] Now, you might not see anything here very much because I've got most of it collapsed.

[00:02:24] But um if I, if I expanded the whole thing, this is what you would, you would see.

[00:02:29] So I've got a catalog, and a catalog is basically an array of CD objects.

[00:02:35] And so as I scroll down through here, you can see that the catalog is really just a list or an array of, of C DS and all the C DS in this case have the same structure, they have the same properties.

[00:02:48] So I want to kind of step you through the, the syntax for, for Jason so that you can learn that it's, it's really, really very simple.

[00:02:59] So there's a few data types that the JSON format supports. It has numbers.

[00:03:06] And when they talk about a number in JSON, it's really like AAA float or a double, actually an eight-point floating point or a double.

[00:03:14] So they have numbers which are really just double.

[00:03:16] So if you want to represent integers or floating-point values, you just use a number and they have strings, they have Boolean. So Booleans can be true or false and they also have um a null.

[00:03:30] So they have a null value, which is, which is kind of like a data type, the null type.

Start visual description. The professor explains the primitive data types supported by JSON, including numbers, strings, Booleans, and null values, and how they are represented in the format. End visual description.

[00:03:34] So those are the primitive types that are built into JSA numbers, strings, bulls and null.

[00:03:40] And then the aggregate data types that are built into to JSON are arrays.

[00:03:46] And so in, in JSON, we use square brackets to delimit the elements of an array, just like many programming languages do.

[00:03:53] And the last aggregate data type in JSON is an object.

[00:03:58] So actually in in this file, um you can see that objects are delimited by curly braces.

[00:04:05] And so if I open this object, you can see that this is an object. So, um then, the left curly brace begins the object, the right curly brace ends the object.

Start visual description. The professor demonstrates the use of objects in JSON, showing how properties are defined within curly braces and how objects can contain arrays or other objects. End visual description.

[00:04:18] And then in between the curly braces, we just provide the properties of the object.

[00:04:22] So in this case, there's only one property on this object which is named catalog.

[00:04:26] So you can see here that the properties of an object, their names have to be in, in quotes.

[00:04:32] And then we have, so we have the name of the property and quotes, and we have a colon and then we have the value of the property.

[00:04:38] Now, most objects are going to have more than one property.

[00:04:41] If there were more than one property, we would put a comma right after the value of the previous property.

[00:04:46] And then we could just have the next property.

[00:04:49] Now, in this position here, we can see we have an array which is an array of C DS.

[00:04:54] But in general, that could be a number, it could be a string, it could be a true or false or a null.

[00:05:00] And in this case, it's an array. So, if I open up the catalog's value, you can see that it's an array and, and down at the bottom, you can see the end of the array right here.

[00:05:13] And then the elements of this particular array are objects.

[00:05:17] So the elements of an array can be any primitive type, or they can be other arrays or objects.

[00:05:23] And so in this case, we have CD objects and so I could open those. And um in this case, you can see that the CD objects are wrapped inside another object and So in this case, we have an object that only has one property which is named CD.

[00:05:41] And so if we actually want to see the data for a CD, we need to open it one other level.

[00:05:46] And so if we open at one more level, we can see the title, Artist Country, et cetera, all the properties of, of the CD.

[00:05:53] So in this example, this, this JSON example is quite verbose, it's, it's very descriptive.

[00:05:58] So you can just look at the data and understand that it's a catalog of, of something.

[00:06:03] And then you can see here that the objects that are in the array are in the catalog or C DS and so forth.

[00:06:10] And, and so sometimes we use these extra wrapper objects to just make the data more descriptive.

[00:06:16] But if you think about it, we could have really represented this array of C DS without the catalog wrapper object or the CD wrapper objects, we could have

just had an array at the very top level and just had an array of, of uh C DS. And so, if we did it that way, it would look something like this.

[00:06:36] Now, in this case, we still have the catalog wrapper, but we took out the CD wrappers. So, in this case, you can see it's a little less for both.

[00:06:43] We have a catalog which is an array of objects.

[00:06:46] Now we know that those are those objects represent C DS.

[00:06:50] Um so we can leave out those CD wrapper objects and that makes it a little less for both.

[00:06:55] So it just depends on how, how descriptive you want your, your data to be.

[00:07:00] OK. So that's the basic uh syntax of JSON. Now, you can see numbers are just like floating points, you can have integer, you can have floating points.

[00:07:10] If you do use integers, they are represented exactly.

[00:07:13] So there's no round off error if you use integer values. Of course, when you use floating point values, there is uh round off error potentially as there always is with floating points.

[00:07:23] And then you can see that the strings are delimited with quotes and so on and so forth. So really, it's, it's quite a simple data format.

[00:07:32] Now, it's probably informative to know that the reason that this syntax was chosen for JSON is because this is essentially the same syntax that's used in the JavaScript programming language. So, in JavaScript, if you represent literal data, literal objects, li literal arrays and so forth, this is essentially the, the JavaScript syntax for representing literal data.

Start visual description. The professor explains the simplicity and versatility of the JSON format, emphasizing its widespread use across different programming languages and its ease of parsing. End visual description.

[00:07:55] And so since a lot of the programming that's done on the internet is web-based programming, then it was very convenient at some point for JavaScript programmers to say, you know what XML was a great format and it's, it's, it's fine, but it's kind of a pain to deal with because we always have to convert XML data into JavaScript data? Wouldn't it be nice if our data format was natively JavaScript.

[00:08:18] And so that's where, where Jason was born is essentially just JavaScript S and text.

[00:08:23] And it's been uh co-opted and now it's, it's used to represent data in any language. So, JSON is not JavaScript specific in any way, it's used in many languages.

[00:08:33] And the nice thing about this format is it's simple, it can be parsed easily in any language.

[00:08:38] And so, um it, it's a good choice from, from that perspective.