Using JSON | Lesson 3

Practice: Creating JSON Objects

Instructions

Once you have created the JSON object representing a movie in a catalog, answer the following questions to consider the choices you made in your code.

Add the link to your shared GitHub repository in the space below.	
Strings	**What textual information did you include as strings? Why?**
	There are string data types in the objects and arrays but the title is the only string value.
Numbers	**Could you imagine other numeric values you might add (e.g., ratings, runtime)?**
	Adding a runtime value numerically would be beneficial to see how long you need to see the movie. You could also use a rating system on a scale of 1 to 5 and use numbers for that. If you wanted to add a total number of copies of something to keep track of what is available.

Booleans	**How might the `isAvailable` field be used in a real application?**
	It could be a quick check to see if the movie is available for the user to watch. If it isn't available then they can't watch it. That true/false boolean value would let us know.
Arrays	**Why is it helpful to store genres and cast as arrays instead of strings?**
	They are ordered lists and contain multiple values. They don't need key value pairs and are accessible by the index. Movies below to multiple genres so there is flexibility. The data is structured and expandable if you want to add. Easy to loop through and display lists dynamically and enhances the user experience.
Nested Objects	**How does the use of nested objects (e.g., the `cast` array) make the data more flexible?**
	Each actor has their own object with details and allows for additional attributes to be added later. Nested Objects make queries more precise, instead of searching the text blob the system can search the structured attributes directly. Weel organized arrays ensure seamless integration across different

	services, like recommendation engines or third-party analytics.