# **60. JSON, XML, AND YAML**

## **DATA SERIALIZATION**

**Data serialization** is the process of converting data into a standardized format/structure that can be stored (in a file) or transmitted (over a network) and reconstructed later (e.g., by a different application). This allows data to be communicated between applications in a way both applications understand.

Data serialization languages allow us to represent *variables* using text.

## **JSON (JavaScript Object Notation)**

**JSON (JavaScript Object Notation)** is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects.

* Standardized in [RFC 8259](https://datatracker.ietf.org/doc/html/rfc8259)
* Derived from JavaScript but language-independent (many modern programming languages can generate and read JSON data)
* Frequently used in REST APIs
* *Whitespace* is insignificant

### **JSON Data Types**

#### **Primitive Data Types:**

* **String**: Text value surrounded by double quotes (" ")
  + Example: "Hello", "Five", "5"
* **Number**: Numeric value without quotes
  + Example: 5, 100
* **Boolean**: Can only be true or false (not surrounded by quotes)
* **Null**: Represents the intentional absence of any object value (null, without quotes)

#### **Structured Data Types:**

* **Object**: Unordered list of *key-value pairs* (also called a *dictionary*)  
  + Surrounded by curly brackets {}
  + Keys must be strings
  + Values can be any valid JSON data type (string, number, boolean, null, object, array)
  + Each key-value pair is separated by a colon (:)
  + Multiple key-value pairs are separated by a comma
* **Array**: A series of *values* separated by commas  
  + Not *key-value pairs*
  + Values can be of different data types

## **XML (Extensible Markup Language)**

**XML (Extensible Markup Language)** was developed as a markup language but is now used as a general data serialization format.

* Originally designed for formatting text (like HTML)
* Less human-readable compared to JSON
* *Whitespace is insignificant*
* Often used by REST APIs
* Uses <key>value</key> syntax (similar to HTML)

## **YAML (YAML Ain’t Markup Language)**

**YAML** originally stood for *Yet Another Markup Language*, but to distinguish its purpose as a data-serialization language rather than a markup language, it was repurposed as *YAML Ain’t Markup Language*.

* Used by the network automation tool **Ansible** (covered later in the course)
* **Very human-readable**
* *Whitespace is significant* (unlike JSON and XML)
  + Indentation is crucial
* YAML files start with --- (three dashes)
* - is used to indicate a list
* Keys and values are represented as key: value

## **JSON vs. YAML Comparison**

Below is a comparison of JSON and YAML using the same data structure:

**JSON Example:**

{

"name": "John Doe",

"age": 30,

"is\_student": false,

"courses": ["Math", "Science"],

"address": {

"street": "123 Main St",

"city": "New York"

}

}

**YAML Example:**

name: John Doe

age: 30

is\_student: false

courses:

- Math

- Science

address:

street: 123 Main St

city: New York

Each serialization format has its strengths and use cases:

* **JSON**: Widely used in web applications and APIs
* **XML**: Common in legacy systems and certain APIs
* **YAML**: Popular in configuration files and automation tools like Ansible