

XML, JSON and YAML are the most popular data serialization languages. This means we use them to represent data structures and values, which enables data storage, transfer and distribution, often for use in configurations.

XML stands for “eXtensible Markup Language”

XML looks like this:

<Configurations>

<Config>

<Name>Ingress</Name>

<Value>data/input</Value>

</Config>

<Config>

<Name>Egress</Name>

<Value>data/output</Value>

</Config>

</Configurations>

JSON stands for “JavaScript Object Notation”

JSON looks like this:

{

"configurations":[

{

"name": "Ingress",

"value": "data/input"

},

{

"name": "Egress",

"value": "data/output"

}

]

}

YAML stands for “YAML Ain't Markup Language”

YAML looks like this:

---

configurations:

- name: Ingress

value: data/input

- name: Egress

value: data/output

Now that we know the meaning of their names and what they look like, let’s compare their functional qualities.

Data representation:

XML is a markup language, whereas JSON and YAML are data formats. XML uses tags to define the elements and stores data in a tree structure, whereas data in JSON is stored like a map with key/value pairs. YAML, on the other hand, allows representation of data both in list or sequence format and in the form of a map with key/value pairs. JSON and YAML uses different indentation styles: JSON uses tabs, whereas YAML uses a hyphen (-) followed by whitespace.

Comments:

Comments makes it easier to understand and interpret data. Whereas JSON has no concept of comments, XML allows you to add comments within a document. YAML was designed for readability and thus allows comments.

Data types:

XML supports complex data types such as charts, images, and other non-primitive data types. JSON supports only strings, numbers, arrays, Boolean, and objects. YAML, on the other hand, supports complex data types such as date and time stamps, sequences, nested and recursive values, and primitive data types.

Data readability:

It is difficult to read and interpret data written in XML, but it is fairly easy to interpret data in JSON format, and it is much easier to read data in YAML than in JSON format.

Usability and purpose:

XML is used for data interchange (that is, when a user wants to exchange data between two applications). JSON is better as a serialization format and is used for serving data to application programming interfaces (APIs). YAML is best suited for configuration.

Speed:

XML is bulky and slow in parsing, leading to relatively slow data transmission. JSON files are considerably smaller than XML files, and JSON data is quickly parsed by the JavaScript engine, enabling faster data transmission. YAML, as a super-set of JSON, also delivers faster data transmission, but it's important to remember that JSON and YAML are used in different scenarios.

Much of the article above is an excerpt from [Network Programmability and Automation Fundamentals](https://www.ciscopress.com/store/network-programmability-and-automation-fundamentals-9780135183656) by Khaled Abuelenain, Jeff Doyle, Anton Karneliuk, Vinit Jain, published May 6, 2021 by [Cisco Press](http://www.ciscopress.com/). Part of the [Networking Technology](https://www.ciscopress.com/series/series.asp?ser=334950) series.

An MP3 audio version of this article is attached as a ZIP file. Download, unZIP it and play.