YAML-JSON

Difference Bettween YAML and JSON

*What's the difference between*[*YAML*](https://lms.clarusway.com/mod/lesson/view.php?id=1811)*and*[*JSON*](https://lms.clarusway.com/mod/lesson/view.php?id=1812)  *Below are comparison between*[*YAML*](https://lms.clarusway.com/mod/lesson/view.php?id=1811)*and*[*JSON*](https://lms.clarusway.com/mod/lesson/view.php?id=1812)

* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) is best suited for configuration while [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) is better as a serialization format or serving up data for your APIs.
* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) has a couple of big advantages including the ability to self-reference, support for complex datatypes, embedded block literals, comments, and more.
* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) is by no means a replacement for [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812). You should use the data format that makes the most sense for what you are trying to accomplish.
* [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) learning is faster in comparison to [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) because it is not nearly as robust in its feature set.
* You can parse [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) with a [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) parser.
* [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) is best for data interchange.

[YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811)

car:

color: blue

price: $30,000

[JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812)

{

"car": {

"color":"blue",

"price": "$30000"

}

}

YAML-JSON

Syntax Comparison

[JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) Syntax:

* [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) is a subset of the JavaScript object notation syntax.
* [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) data stored in name/value pairs.
* [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) records separated by commas.
* [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) field names & strings are wrapped by double quotes.

[YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) Syntax:

* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) stands for ain’t markup language and is a superset of [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812) – You Convert [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) to [JSON](https://lms.clarusway.com/mod/lesson/view.php?id=1812)
* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) files begin with ‘—‘, marking the start of the document.
* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) documents end with ‘…’ but it’s optional.
* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) key value pairs are separated by colon.
* [YAML](https://lms.clarusway.com/mod/lesson/view.php?id=1811) lists begin with a hyphen.

| **Type** | [**YAML**](https://lms.clarusway.com/mod/lesson/view.php?id=1811) | [**JSON**](https://lms.clarusway.com/mod/lesson/view.php?id=1812) |
| --- | --- | --- |
| Comments | Denoted with a hash/number sign | Not allowed |
| Hierarchy | Mappings, and sequences can be nested. Hierarchy is determined by the indentation level | Objects and arrays can be nested, and are denoted by braces and brackets, respectively. |
| Arrays | [first, second, 3] | ["first", "second", 3] |
| Strings | Does not require quoting but supports both single and double quotes | Must be double-quoted. Allows character (tabs, newlines, etc.) escaping with a backslash as the escape character. |
| Numbers | Built-in support for integers, floating-point, octal and hexadecimal numbers | Floating point numbers in scientific notation. Infinity is not permitted. |