YAML & JSON Essentials

Table of Contents

1 What is YAML?

What is YAML?

- **YAML** (a recursive acronym for "**YAML** Ain't Markup Language") is a human-readable data-serialization standard for programming languages.
- It is **commonly used for** configuration files and in applications where data is being stored or transmitted.

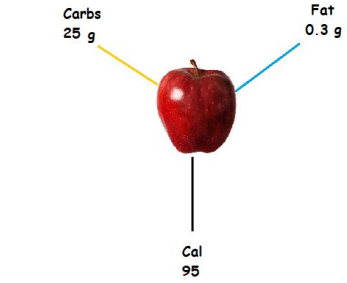
What is YAML?

- The files should have .yaml (or .yml) as the extension.
- YAML is designed by **Clark Evans**, **Ingy döt Net**, and **Oren Ben-Kiki**. (2001).
- https://yaml.org/

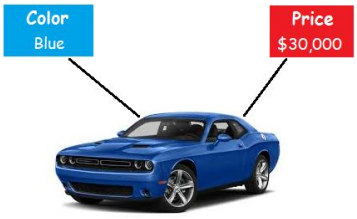
What is YAML?

- The **key-value** is YAML's **basic building block**.
- The key is always a **string**.
- The value is a **scalar** so that it can be various data types.

What is YAML?




```
Apple:
   Carbs: 25 g
   Calories: 95
   Fat: 0.3 g
```




```
car:
   color: blue
   price: $30,000
```

YAML

```yaml
yaml:
  - slim and flexible
  - better for configuration
object:
  key: value
  array:
    - null_value:
    - boolean: true
    - integer: 1
paragraph: >
   Blank lines denote

   paragraph breaks
content: |-
   Or we
   can auto
   convert line breaks
   to save space
```


2 What is JSON?

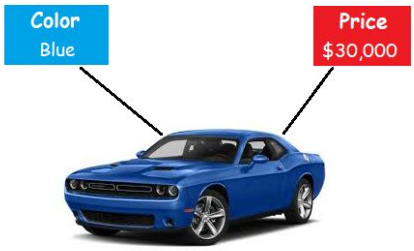
What is JSON?

- The files should have .json as the extension.
- JSON is designed by **Douglas Crockford** (2001).
- https://json.org/

What is JSON?

- **JSON** stands for **J**ava**S**crip**t** **O**bject **N**otation.
- JSON, is a **minimal**, **readable format** for **structuring data**. It is used primarily to transmit data between a server and web application; and also used for storing data.

What is JSON?




```
{
   "car": {
      "color": "blue",
      "price": "$30,000"
   }
}
```

JSON

```
{
  "json": [
    "rigid",
    "better for data interchange"
  ],
  "yaml": [
    "slim and flexible",
    "better for configuration"
  ],
  "object": {
    "key": "value",
    "array": [
      {
        "null_value": null
      },
      {
        "boolean": true
      },
      {
        "integer": 1
      }
    ]
  }
}
```


3 Quick Comparison YAML vs. JSON

Quick Comparison

- **YAML** is best suited for configuration while **JSON** is better as a serialization format or serving up data for your APIs.
- **YAML** has a couple of big advantages including the ability to self-reference, support for complex data types, embedded block literals, comments, and more.
- Easily convertible to each other.

Quick Comparison

Type	YAML	JSON
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.
Date/Timestamp	Supported	Not supported

Quick Comparison

JSON

```
{
  "json": [
    "rigid",
    "better for data interchange"
  ],
  "yaml": [
    "slim and flexible",
    "better for configuration"
  ],
  "object": {
    "key": "value",
    "array": [
      {
        "null_value": null
      },
      {
        "boolean": true
      },
      {
        "integer": 1
      }
    ]
  }
}
```

YAML

```
---
json:
- rigid
- better for data interchange
yaml:
- slim and flexible
- better for configuration
object:
  key: value
  array:
  - null_value:
  - boolean: true
  - integer: 1
```


THANKS!

Any questions?

You can find me at:

- @charlie
- charlie@clarusway.com