JSON

What is JSON?

- JSON is a lightweight text-based open standard data-interchange format.
- It is human readable.
- JSON is derived from a subset of JavaScript programming language (Standard ECMA-262 3rd Edition—December 1999).
- It is entirely language independent and can be used with most of the modern programming languages.
- JSON is often used to serialize and transfer data over a network connection

What is JSON?..

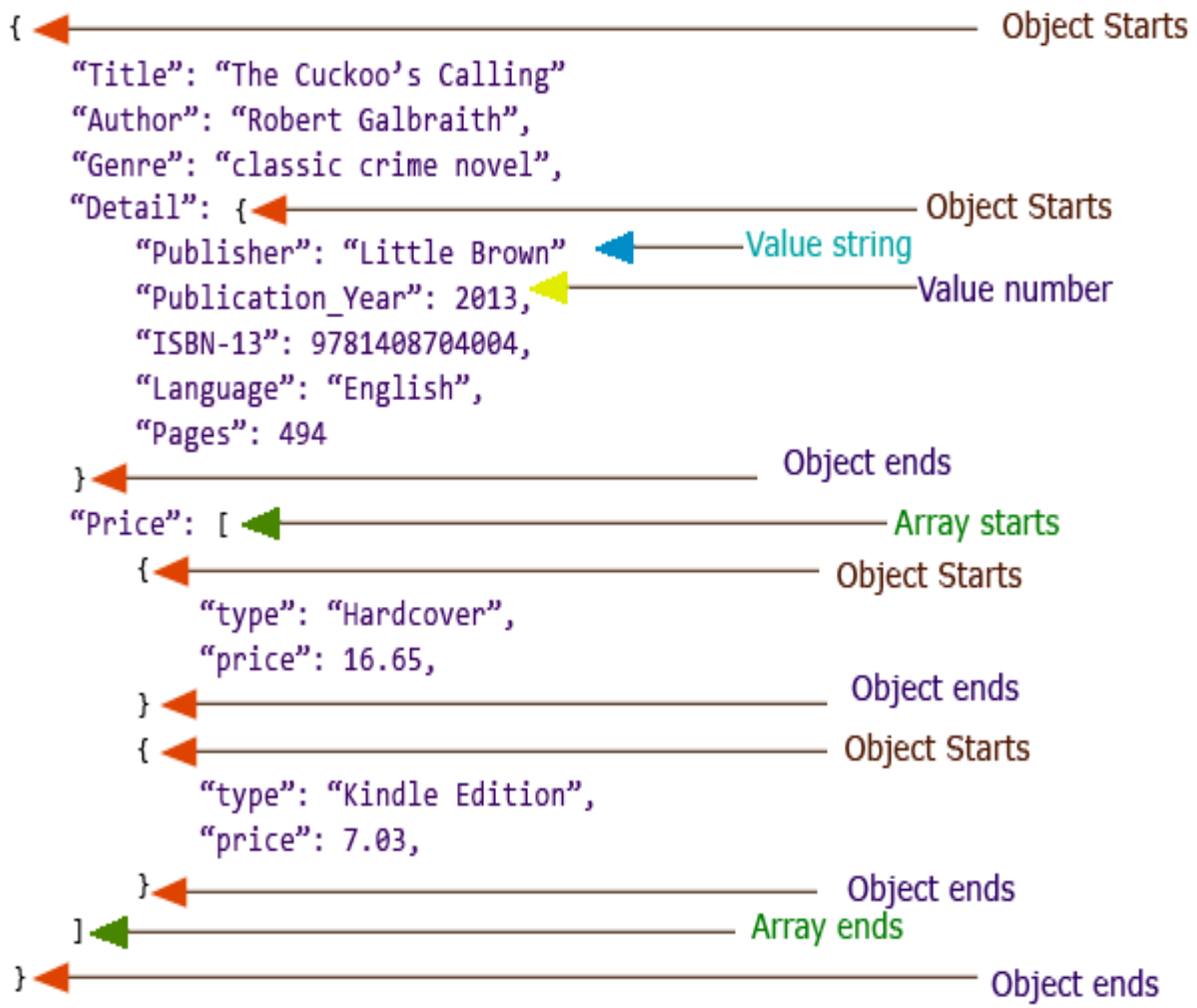
- Serialization is a process to transforming data structures and objects in a format suitable to be stored in a file or memory buffer or transmitted over a network connection
- JSON is a standard and is specified on RFC4627 on IETF (International Engineering Task Force).
  - The specification is made by Doglus Crockford on July 2006
- JSON files are saved with .json extension.
- Internet media type of JSON is "application/json

Example

```
{
        "name": "ABC",
        "age": 20,
}
```

Basic Constructs

- There four basic and built-in data types in JSON.
- They are strings, numbers, booleans (i.e true and false) and null. Besides, there are two data types which are structured - objects and arrays.
- Objects are wrapped within '{' and '}'. Arrays are enclosed by '[' and ']'. Objects are a list of label-value pairs. Arrays are list of values.
- Both objects and arrays can be nested.
- strings, numbers, booleans (i.e true and false) and null can be used as values.



https://www.w3resource.com/JSON/introduction.php

Valid DataTypes

- In JSON, values must be one of the following data types:
- a string
- a number
- an object (JSON object)
- an array
- a boolean
- *null*

Valid DataTypes…

- JSON values **cannot** be one of the following data types:
- a function
- a date
- *undefined*

Data Types in JSON

- JSON supports an array of data types
  - **Object**
  - { string : value, ...}
  - An object starts and ends with '{' and '}'. Between them, a number of string value pairs can reside.
  - String and value is separated by a ':' and if there are more than one string value pairs, they are separated by ',‘
  - **Example**

{ "firstName": "Bidhan", "lastName": "Chatterjee", "age": 40, "email":"bidhan@example.com" }

Array of Object (Example)

- In JSON, objects can nest arrays (starts and ends with '[' and ']') within it

```
{
        "Students": [{
                "Name": "Amit Goenka",
                "Major": "Physics",
                "Minor": ["Mathematics", "Chemistry"]
        }, {
                "Name": "Smita Pallod",
                "Major": "Chemistry",
                "Minor": ["Mathematics", "Physics"]
        }, {
                "Name": "Rajeev Sen",
                "Major": "Mathematics",
                "Minor": ["Chemistry", "Physics", "Biology"]
        }]
}
```

Value

- **Syntax**
- String || Number || Object || Array || TRUE || FALSE || NULL
- A value can be a string, a number, an object, an Array, a Boolean value (i.e. true or false) or Null. This structure can be nested

String

- Syntax
- A string is a sequence of zero or more Unicode characters, enclosed by double quotes, using backslash escapes.

String Types	Description
"	A double quotation mark.
\	Reverse Solidus
/	Solidus
b	Backspace
f	form feed
n	newline
r	Carriage return
t	Horizontal tab
u	Four hexadecimal digits

Number

- The following table shows supported number types.

Number Types	**Description**
Integer	Positive or negative Digits.1-9. And 0.
Fraction	Fractions like .8.
Exponent	e, e+, e-, E, E+, E-

Whitespace

- Whitespace can be placed between any pair of supported data-types.

Validation

- validate your JSON data using JSONLint.
- JSONLint is an open source project which allows you to validate your JSON data
- JSONLint online (http://jsonlint.com/)

Client-side JSON methods

- **Parse()**
  - The JSON.parse() method parses a JSON string, constructing the JavaScript value or object described by the string.
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse
- **Stringify()**
  - The JSON.stringify() method converts a JavaScript object or value to a JSON string.
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify

Installation of JSON in PHP and PHP json_decode function

- JSON supports three PHP functions :
  - json_decode, json_encode and json_last_error

  - **json_decode() Function**
  - json_decode() function decodes a JSON string.
  - Suppose you have obtained some data in JSON format and you want to convert it into PHP variable for the purpose of presenting that data to a user.

JSON Decode function

- **Syntax**
- json_decode(json_string, assoc, depth, options)

Parameters	Type	Description
json_string	String	A JSON encoded string. It must be a UTF-8 encoded data.
assoc	Boolean	If it is true, returned object will be converted into an associative array upon using json_decode function.
depth	Integer	Specifies recursion depth. This is user specified.
Options	Integer	Bitmask of JSON decode. As of this writing, only JSON_BIGINT_AS_STRING is supported.

JSON Decode function

- Json_decode **return value**
- json_decode() function returns an supported PHP type.
- If the available JSON **string can not be decoded** or if the encoded data is deeper than the recursion limit, **it returns NULL.**
- Values true and false are returned as TRUE, FALSE.
- https://www.php.net/manual/en/function.json-decode.php

json_encode() Function

- PHP json_encode() function converts a PHP value into a JSON value
- **json_encode() function returns a string**
- json_encode(value, options)
- https://www.php.net/manual/en/function.json-encode.php

Parameters	Type	Description
value	Mixed	Any PHP type except resource. Must be UTF character encoded data.
options	Integer	Bitmask comprising of JSON_HEX_QUOT, JSON_HEX_TAG, JSON_HEX_AMP, JSON_HEX_APOS, JSON_NUMERIC_CHECK, JSON_PRETTY_PRINT, JSON_UNESCAPED_SLASHES, JSON_FORCE_OBJECT.

json_last_error() Function

- json_last_error()
- While working on encoding or decoding JSON, if an error occur, json_last_error() function returns the last error.
- json_last_error() function returns an integer.
- https://www.php.net/manual/en/function.json-last-error.php

Error constants

Constants	Description
JSON_ERROR_NONE	Specifies that no error occurred.
JSON_ERROR_DEPTH	Specifies that the maximum stack depth has been exceeded.
JSON_ERROR_STATE_MISMATCH	Indicates that the associated JSON is not properly formed or invalid.
JSON_ERROR_CTRL_CHAR	Indicates that the error is in control characters. This usually happens incorrect encoding.
JSON_ERROR_SYNTAX	Indicates that this is a syntax error.
JSON_ERROR_UTF8	Indicates that error occurred due to malformed UTF-8 characters, which usually happens because of incorrect encoding.

Try this out

Problem:

(Create a database table EMP with id, name, designation and salary as attributes in MySql database

1. Create a html page (searchemp.html) to input “Employee Name” and submit (to same page).
2. On submission, call a client-side functionality in the same html document to define an asynchronous request to “searchprocess.php” with the “Employee Name” data (http methods GET or POST)
3. In “searchprocess.php”, retrieve employee information based on Employee Name data communicated and display in suitable JSON encoded format
4. In searchemp.html, receive the response as text and parse the JSON data to meet the client-side requirement. Display the retrieved employee details in HTML table format

Hint: https://www.w3schools.com/js/js_json_php.asp