

*JavaScript Object Notation*. Misleading name!!

Inductive definition for JSON values:

**Primitives** null, true, false, numbers and "-quoted strings.  
Minimal set of escape sequences in strings.

**Sequences** Comma-separated JSON values within [ ].

**Maps** Comma-separated key-value pairs within { }. Keys **must** be JSON strings and values are JSON values.

JSON is largely a subset of JavaScript literal notation with differences in some edge cases.

# JSON Example

In `simpsons.json`

```
[  
  { "id": "bart",  
    "firstName": "Bart",  
    "lastName": "Simpson",  
    "birthdate": "1979-04-01"  
  },  
  { "id": "marge",  
    "firstName": "Marge",  
    "lastName": "Simpson"  
  },  
  { "id": "lisa",  
    "firstName": "Lisa",  
    "lastName": "Simpson",  
    "birthDate": "1982-05-09",  
    "email": "smartgirl63_\\@yahoo.com"
```

# JSON Evaluation

- Widely popular for transferring structured data between heterogeneous systems.
- Preferred over XML for structured **data** (XML is good for structured **documents**).
- In JavaScript, built-in JSON object provides `stringify()` and `parse()` methods to convert JavaScript objects to / from a string.

To pretty-print JavaScript object `obj` as JSON with an indent of 2, use `JSON.stringify(obj, null, 2)`.

- *Not suitable* as a configuration format as *no comments allowed*. (TOML, YAML are better formats). Unfortunately, chosen for package configuration by npm and yarn (as `package.json`).
- Some JSON libraries allow comments (and other features like trailing commas) as syntax extensions, but this is not as per JSON standard.

# JavaScript Object Literals vs JSON

- Easier to write JavaScript object literals than JSON (keys need not be quoted, allows trailing commas, allows comments).
- JavaScript object literals allow anonymous functions; JSON does not.
- JSON works across multiple languages; JavaScript object literals do not.
- JSON is more efficient as it is easier to parse than JavaScript object literals.

**Object Literal Example:** [simpsons.js](#):