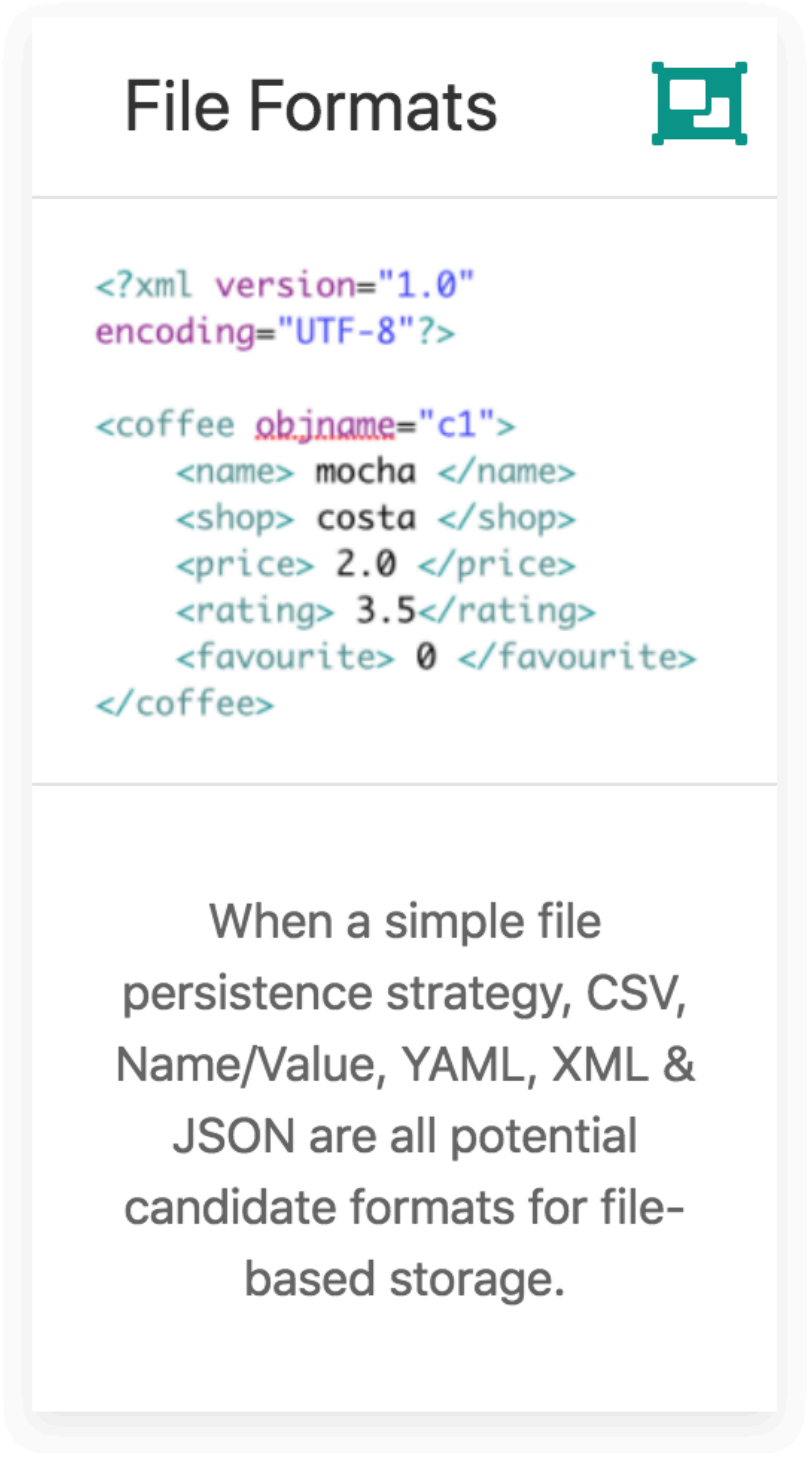
Json



File Formats

```
<?xml version="1.0"
encoding="UTF-8"?>

<coffee objname="c1">
    <name> mocha </name>
    <shop> costa </shop>
    <price> 2.0 </price>
    <rating> 3.5</rating>
    <favourite> 0 </favourite>
</coffee>
```

When a simple file persistence strategy, CSV, Name/Value, YAML, XML & JSON are all potential candidate formats for file-based storage.

JSON

http://en.wikipedia.org/wiki/JSON

- JavaScript Object Notation, is a text-based open standard designed for human-readable data interchange.
- Derived from the JavaScript scripting language, JSON is a language for representing simple data structures and associative arrays, called objects.
- Despite its relationship to JavaScript, JSON is language-independent, with parsers available for many languages.
- The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML.

```
{
  "name":"mocha",
  "shop":"costa",
  "rating":3.5,
  "price":2.0,
  "favourite":0,
  "id":1
},
{
  "name":"americano",
  "shop":"costa",
  "rating":4.5,
  "price":3.0,
  "favourite":1,
  "id":2
},
{
  "name":"cappuccino
lite",
  "shop":"starbucks",
  "rating":1.5,
  "price":4.0,
  "favourite":1,
  "id":3
}
```

JavaScript Object Notation

- Language Independent.
- Text-based.
- Light-weight.
- Easy to parse.

- Language independent
  - uses JavaScript syntax for describing data objects
  - parsers and JSON libraries exists for many different programming languages
- "self-describing" and easy to understand
- Evaluates to JavaScript Objects:
  - The JSON text format is syntactically identical to the code for creating JavaScript objects.
  - Because of this similarity, instead of using a parser, a JavaScript program can use the built-in eval()*** function and execute JSON data to produce native JavaScript objects.

JSON

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address":
    {
      "streetAddress": "21 2nd Street",
      "city": "New York",
      "state": "NY",
      "postalCode": 10021
    },
  “phoneNumbers":
    [
      {
        "type": "home",
        "number": "212 555-1234"
      },
      {
        "type": "fax",
        "number": "646 555-4567"
      }
    ]
}
```

- XML: 549 characters
- JSON: 326 characters

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<persons>
  <person>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <age>25</age>
    <address>
      <streetAddress>21 2nd Street</streetAddress>
      <city>New York</city>
      <state>NY</state>
      <postalCode>10021</postalCode>
    </address>
    <phoneNumbers>
      <phoneNumber>
        <number>212 555-1234</number>
        <type>home</type>
      </phoneNumber>
    <phoneNumber>
      <number>646 555-4567</number>
      <type>fax</type>
      </phoneNumber>
    </phoneNumbers>
  </person>
</persons>
```

JSON vs XML

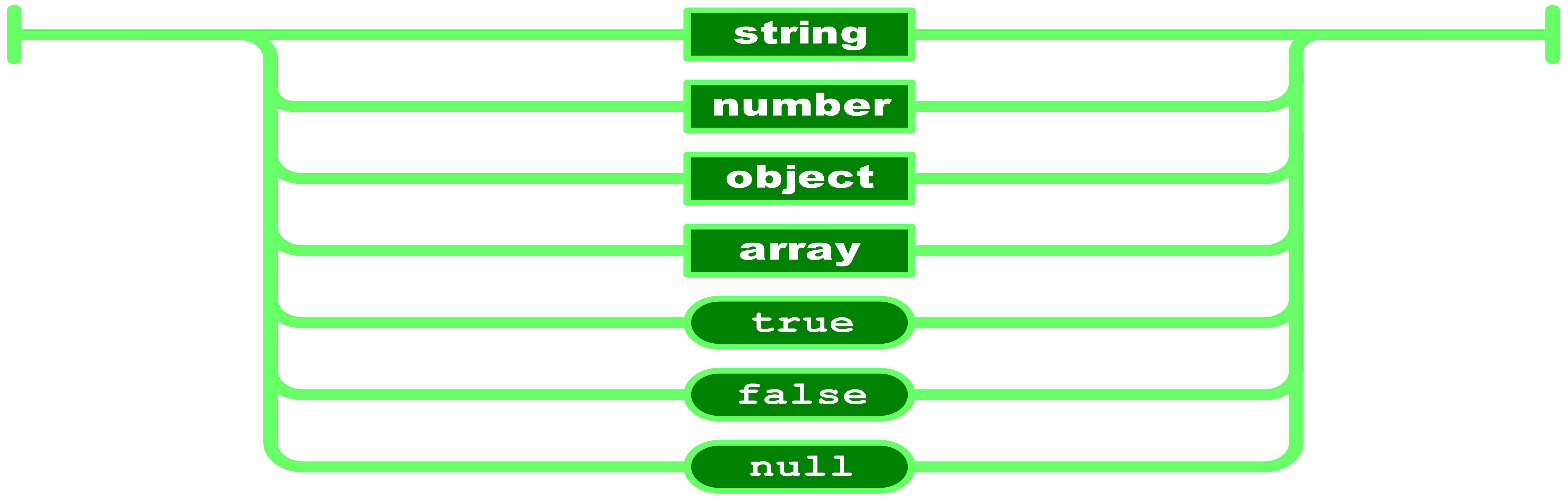
- Advantages of XML

  - Message validation needed
  - Transformation in XSLT (possibly)
  - Excessive marked-up text
  - Tooling less accessible

- Advantages of JSON

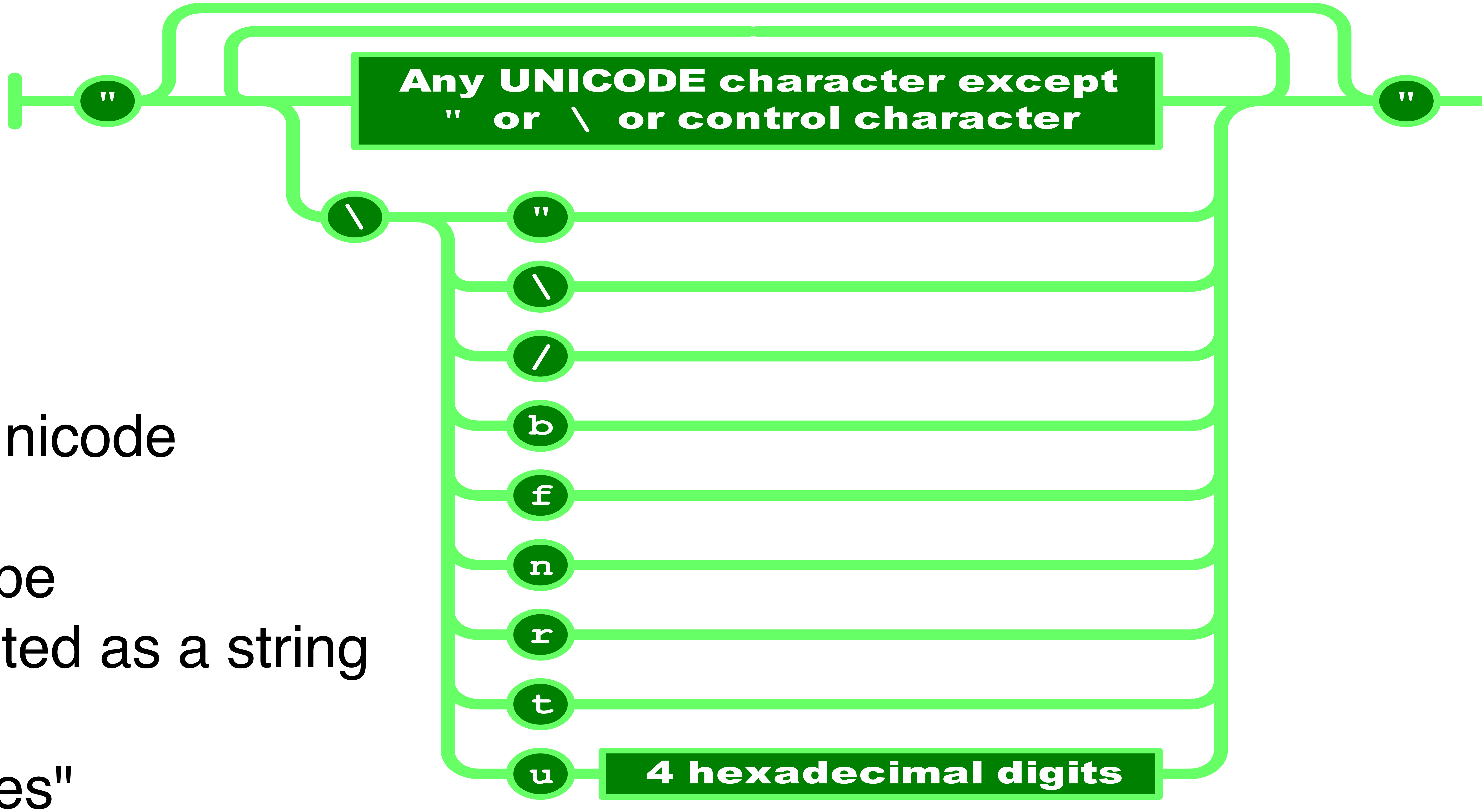
  - Validation simplified
  - Transformation in Javascript (possibly)
  - No markup text
  - Tooling widely available

JSON Values



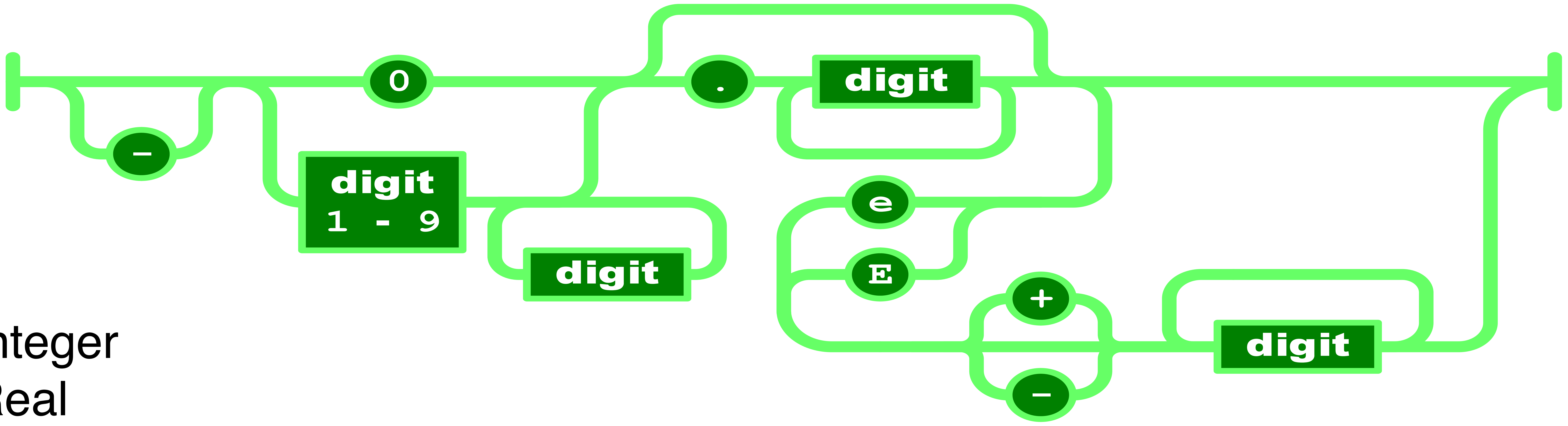
- JSON values can be:
  - A number (integer or floating point)
  - A string (in double quotes)
  - A boolean (true or false)
  - An object (in curly brackets)
  - An array (in square brackets)
  - null

Strings



- Sequence of 0 or more Unicode characters
- No separate character type
  - A character is represented as a string with a length of 1
- Wrapped in "double quotes"
- Backslash escapement

Numbers



- Integer
- Real
- Scientific

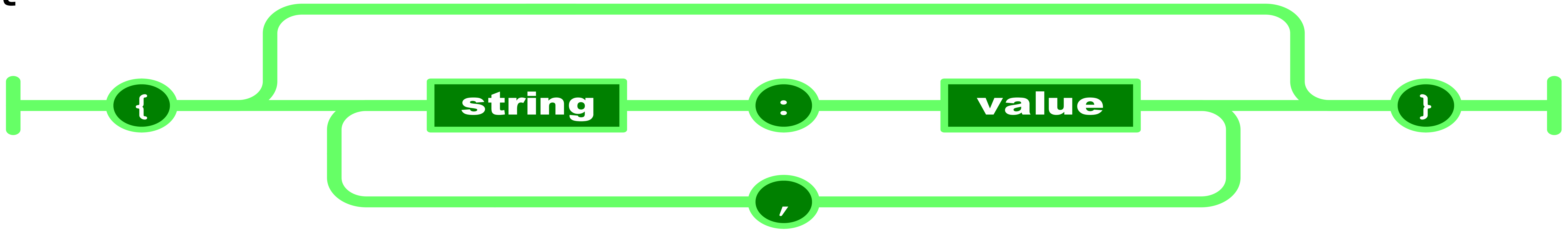
- No octal or hex
- No NaN or Infinity
  - Use null instead

Object & Arrays

- Object
  - Unordered set of name-value pairs
  - names must be strings
  - { name1 : value1, name2 : value2, …, nameN : valueN }

- Array
  - Ordered list of values
  - [value1, value2, … valueN]

Object



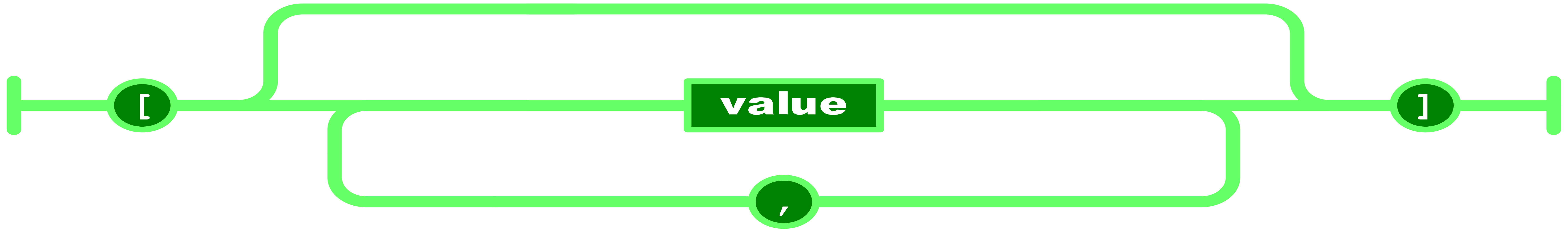
- Objects are unordered containers of key/value pairs
- Objects are wrapped in { }
- , separates key/value pairs
- : separates keys and values
- Keys are strings
- Values are JSON values

  - struct, record, hashtable, object

Object

```
{
  "_id":"560515770f76130300c69953",
  “usertoken":"11343761234567808125",
  "paymenttype":"PayPal",
  "__v":0,
  "upvotes":0,
  “amount":1999
}
```

Array



- Arrays are ordered sequences of values
- Arrays are wrapped in `[]`
- `,` separates values
- JSON does not talk about indexing.
  - An implementation can start array indexing at 0 or 1.

Array

```
[
  {
    "_id": "560515770f76130300c69953",
    "usertoken": "11343761234567808125",
    "paymenttype": "PayPal",
    "__v": 0,
    "upvotes": 0,
    "amount": 1999
  },
  {
    "_id": "561252404218920300484 03d",
    "usertoken": "11343761234567808125",
    "paymenttype": "PayPal",
    "__v": 0,
    "upvotes": 5,
    "amount": 1234
  },
  {
    "_id": "5627620ac9e9e303005b113c",
    "usertoken": "11343761234567808125",
    "paymenttype": "Direct",
    "__v": 0,
    "upvotes": 2,
    "amount": 1001
  }
]
```

Versionless

- JSON has no version number.
- No revisions to the JSON grammar are anticipated.
- JSON is very stable.

Rules

- A JSON decoder must accept all well-formed JSON text.
- A JSON decoder may also accept non-JSON text.
- A JSON encoder must only produce well-formed JSON text.
- Be conservative in what you do, be liberal in what you accept from others.

Google’s Gson

- https://sites.google.com/site/gson/gson-user-guide

- Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Gson is an open-source project hosted at http://code.google.com/p/google-gson.

- Gson can work with arbitrary Java objects including pre-existing objects that you do not have source-code of.

JSON Summary

- JSON is a standard way to exchange data
  - Easily parsed by machines
  - Human readable form

- JSON uses dictionaries and lists
  - Dictionaries are unordered
  - Lists are ordered

- GSON is Googles JSON parser
  - Very simple to use