



## Introduction to JSON

### What is JSON? (From w3schools.com)

JSON stands for JavaScript Object Notation. It is a lightweight data interchange format, frequently used to send data from a server to a web browser.

It is language-independent, self-describing and easy to humans to read and easy for programming languages to parse.

Nearly all programming languages have libraries available to handle JSON formatted data. LiveCode is no exception.

Here is a nice introduction by a blogger named Scott Lowe.

<http://blog.scottlowe.org/2013/11/08/a-non-programmers-introduction-to-json/>

Here's a more in-depth treatment by w3resource.com:

<http://www.w3resource.com/JSON/introduction.php>

And here's another good one by elated.com:

<http://www.elated.com/articles/json-basics/>

Each of the above tutorials does a nice job of describing JSON-formatted strings, so I won't repeat them in detail here. In a nutshell, objects are defined between curly braces { }, and elements within the objects are given as name-value pairs separated by a colon. Individual elements are separated by commas. Elements may also contain arrays, which are defined between square brackets [ ] .

Sample JSON String:

```
{
  "Title": "The Cuckoo's Calling",
  "Author": "Robert Galbraith",
  "Genre": "classic crime novel",
  "Detail": {
    "Publisher": "Little Brown",
    "Publication_Year": 2013,
    "ISBN-13": 9781408704004,
    "Language": "English",
    "Pages": 494
  },
  "Price": [
    {
      "type": "Hardcover",
      "price": 16.65
    }
  ]
}
```

```
    },  
    {  
        "type": "Kindle Edition",  
        "price": 7.03  
    }  
]  
}
```

Beyond this basic knowledge you don't need to know too much more to work with JSON strings in LiveCode. There is a free, open-source library called EasyJSON available that we will use to convert JSON-formatted data returned from web services into LiveCode arrays. Once the data is in an array, it is a simple matter to pull out the bits we need using LiveCode array notation.

In LiveCode 8.0 and later there is a JSON library included, which gives you functions for converting JSON text to an array and vice-versa. Coupled with a tree view widget it is simple to create a visual arrangement of JSON data.

Example from Google Geocoding API

<https://developers.google.com/maps/documentation/geocoding/>

Show brute-force parsing in LC script vs. using JSON library to turn it into an array.