

Extra notes - Client-side Design and Development

- Dr Nick Hayward

JS - JSON

A brief introduction to JSON, and its use with JavaScript.

Contents

- Intro
- Syntax
 - JSON objects
 - JSON arrays
- References

Intro

JSON, or **JavaScript Object Notation**, is considered a lightweight format and wrapper for storing and transporting data. It is inherently language agnostic, easy to read and understand, and growing rapidly in popularity. For many online APIs, it has now replaced XML as the preferred format for data exchange.

Syntax

Whilst the syntax of JSON is itself derived from JS object notation, it is, in fact, a text only format. As such, this allows us to easily write, describe, and manipulate JSON in practically any programming language.

JSON syntax follows a few basic rules,

- data is recorded as name/value pairs
- data is separated by commas
- objects are defined by a start and end curly brace
 - `{ }`
- arrays are defined by a start and end square bracket
 - `[]`

So, the underlying construct for JSON is a pairing of name and value. JSON can, naturally contain many such pairs, and they look as follows,

```
"city":"Marseille"
```

One of the inherent differences between JSON and JavaScript is the strict usage of the double quotes. JSON names require this usage of double quotes, `"..."`.

JSON objects

As mentioned, JSON objects are contained within curly braces. As with JavaScript, these objects can contain multiple name/value pairs,

```
{  
  "country":"France",  
  "city":"Marseille"  
}
```

JSON arrays

JSON can also use arrays, which are contained within square brackets. These arrays can also contain objects.

```
{
  "cities": [
    {
      "name": "Marseille",
      "region": "Provence-Alpes-Côte d'Azur"
    },
    {
      "name": "Paris",
      "region": "Île-de-France"
    }
  ]
}
```

We can then use this with JavaScript, and parse the JSON object.

- [JSFiddle - Parse JSON](#)

References

- [JSONLint - JSON Validator](#)
- [MDN - JS Objects](#)
- [W3 - JS Object](#)