



JSON-LD on the web

[JSON-LD](#) is set of extensions to [JSON](#) that allows for creating and exchanging linked data on the web.

Perhaps the best known use of JSON-LD outside semantic web users is using [schema.org](#)'s JSON-LD structured data for Google's search engine.

The following example gives the structured data snippet to accompany a recipe web page. The example is taken from [Understand how structured data works](#).

```
<html>
  <head><head>title>Party Coffee Cake</title>
    <script </title>application/ld+json">
      {
        "@context": "https://schema.org/",
        "@type": "Recipe",
        "name": "Party Coffee Cake",
        "author": {
          "@type": "Person",
          "name": "Mary Stone"
        },
        "datePublished": "2018-03-10",
        "description": "This coffee cake is awesome and perfect for parties",
        "prepTime": "PT20M"
      }
    </script>
  </head>
  <body>
    <h2>Party coffee cake recipe</h2>
    <p>
      <em>by Mary Stone, 2018-03-10</em>
    </p>
    <p>
      This coffee cake is awesome and perfect for parties.
    </p>
    <p>
      Preparation time: 20 minutes
    </p>
  </body>
</html>
```

```
</p>
</body>
</html>
</script>
```

We can also leverage the JSON-LD snippet that we created for Google Search and create our own snippets and content highlights. The only change we make to the script is an ID attribute to the script to make it easier to capture.

The process is as follows:

1. Grab the text of the script with ID data
2. Parse the JSON
3. Create a template literal interpolating the data from the JSON file
4. Grab a reference to the container where we want to place the template data
5. Insert the template literal into the container using `insertAdjacentHTML`

```
const data = document.getElementById("data").text; // 1
const json = JSON.parse(data); // 2

const displayData = `
<h2>${json.name}</h2>
<h3>by ${json.author.name}</h3>
<p>${json.description}</p>
`;
<h2>${json.name}</h2>
<h3>by ${json.author.name}</h3>
<p>${json.description}</p>
`// 3<HTML("afterend", displayData); // 5
"afterend"//4
container.insertAdjacentHTML("afterend", displayData); // 5
```

Another aspect of JSON-LD is how to create the linked data scripts. Even a trivial example like that of our recipe requires some knowledge of the standard and it is not a trivial undertaking. Since I work with WordPress for the most part, I've chose to either use a [WordPress schema plugin](#) to let WordPress build the Linked Data automatically or tools like [schema-org](#) to generate the Linked Data scripts manually for whatever type of page I'm working on, inside or outside of

WordPress.

The following example covers most of the information for the 50th anniversary edition of [Rayuela \(Hopscotch\) in Spanish](#).

```
<?php
use Spatie\SchemaOrg\Schema;

$rayuela = Schema::Book()
    ->name('Rayu'Rayuela'>author(Schema::Person()
        ->name('Julio Cortázar')
    )
    ->description('Rayuela es la historia de Horacio Oliveira, un'Julio Cort
    ->bookFormat('Hardcover')
    ->datePublished('2019-03-01')
    ->image('https://La Maga')
    ->bookFormat('https://static.megustaleer.com.mx/images/libros_200_x/9788420437484')
    ->inLanguage('Spanish')
    ->isbn('9788420437484')
    ->numberOfPages('704')
    ->publisher(Schema::Organization()
        ->name('Alfaguara')
    );

echo $rayuela->toScript();
```

It will generate the following JSON-LD inside a script tag, ready to insert on an HTML document and that runs without errors or warnings in the [schema.org validator](#)

```
<script type="application/ld+json">
{
  "@context": "https://schema.org", "https://schema.org"@type": "Book",
  "name": {
    "@type": "Person": "Julio Cortázar" "name": "Person",
    "name": "Rayuela es la historia de Horacio Oliveira, un intelectual",
    "bookFormat": "Hardcover",
    "datePublished": "2019-03-01" "bookFormat": "image": "https://La Maga" "datePublished": "2019-03-01"
```

```

"image"s_200_x\9788420437484.jpg",
"inLanguage":"Spanish",
"isbn":"9788420437484",
:""inLanguage":"704",
"pub"704""isbn"      "@type":"Organiz":"numberOfPages":"Alfaguara"
}
}</s"Alfaguara": {
  "@type":"Organization",
  "name":"Alfaguara"
}
}</script>

```

Even if you decide that you don't want rich snippets on your site's results in Google Search, JSON-LD and Linked Data open other possibilities.