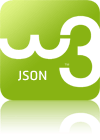
**JSON Tutorial**

[« W3Schools Home](http://www.w3schools.com/default.asp)

[Next Chapter »](http://www.w3schools.com/json/json_intro.asp)



JSON: **J**ava**S**cript **O**bject **N**otation.

JSON is syntax for storing and exchanging text information. Much like XML.

JSON is smaller than XML, and faster and easier to parse.

**JSON Example**

{  
"employees": [  
{ "firstName":"John" , "lastName":"Doe" },   
{ "firstName":"Anna" , "lastName":"Smith" },   
{ "firstName":"Peter" , "lastName":"Jones" }  
]  
}

The employees object is an array of 3 employee records (objects).

**What is JSON?**

* JSON stands for **J**ava**S**cript **O**bject **N**otation
* JSON is lightweight text-data interchange format
* JSON is language independent **\***
* JSON is "self-describing" and easy to understand

|  |  |
| --- | --- |
| **\*** | JSON uses JavaScript syntax for describing data objects, but JSON is still language and platform independent. JSON parsers and JSON libraries exists for many different programming languages. |

**JSON - 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 - Introduction**

[« Previous](http://www.w3schools.com/json/default.asp)

[Next Chapter »](http://www.w3schools.com/json/json_syntax.asp)

**Try it Yourself**

With our editor, you can edit JavaScript code online and click on a button to view the result:

**JSON Example**

<!DOCTYPE html>  
<html>  
<body>  
<h2>JSON Object Creation in JavaScript</h2>

<p>  
Name: <span id="jname"></span><br />   
Age: <span id="jage"></span><br />   
Address: <span id="jstreet"></span><br />   
Phone: <span id="jphone"></span><br />   
</p>

<script>  
var JSONObject= {  
"name":"John Johnson",  
"street":"Oslo West 555",   
"age":33,  
"phone":"555 1234567"};  
document.getElementById("jname").innerHTML=JSONObject.name   
document.getElementById("jage").innerHTML=JSONObject.age   
document.getElementById("jstreet").innerHTML=JSONObject.street   
document.getElementById("jphone").innerHTML=JSONObject.phone   
</script>  
  
</body>  
</html>

[Try it yourself »](http://www.w3schools.com/json/tryit.asp?filename=tryjson_create)   
Click on the "Try it yourself" button to see how it works

**Much Like XML**

* JSON is plain text
* JSON is "self-describing" (human readable)
* JSON is hierarchical (values within values)
* JSON can be parsed by JavaScript
* JSON data can be transported using AJAX

**Much Unlike XML**

* No end tag
* Shorter
* Quicker to read and write
* Can be parsed using built-in JavaScript eval()
* Uses arrays
* No reserved words

**Why JSON?**

For AJAX applications, JSON is faster and easier than XML:

Using XML

* Fetch an XML document
* Use the XML DOM to loop through the document
* Extract values and store in variables

Using JSON

* Fetch a JSON string
* eval() the JSON string

**JSON Syntax**

[« Previous](http://www.w3schools.com/json/json_intro.asp)

[Next Chapter »](http://www.w3schools.com/json/json_eval.asp)

JSON syntax is a subset of JavaScript syntax

**JSON Syntax Rules**

JSON syntax is a subset of the JavaScript object notation syntax:

* Data is in name/value pairs
* Data is separated by commas
* Curly braces hold objects
* Square brackets hold arrays

**JSON Name/Value Pairs**

JSON data is written as name/value pairs.

A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

"firstName" : "John"

This is simple to understand, and equals to the JavaScript statement:

firstName = "John"

**JSON Values**

JSON values can be:

* A number (integer or floating point)
* A string (in double quotes)
* A Boolean (true or false)
* An array (in square brackets)
* An object (in curly brackets)
* null

**JSON Objects**

JSON objects are written inside curly brackets,

Objects can contain multiple name/values pairs:

{ "firstName":"John" , "lastName":"Doe" }

This is also simple to understand, and equals to the JavaScript statements:

firstName = "John"  
lastName = "Doe"

**JSON Arrays**

JSON arrays are written inside square brackets.

An array can contain multiple objects:

{  
"employees": [  
{ "firstName":"John" , "lastName":"Doe" },   
{ "firstName":"Anna" , "lastName":"Smith" },   
{ "firstName":"Peter" , "lastName":"Jones" }  
]  
}

In the example above, the object "employees" is an array containing three objects. Each object is a record of a person (with a first name and a last name).

**JSON Uses JavaScript Syntax**

Because JSON uses JavaScript syntax, no extra software is needed to work with JSON within JavaScript.

With JavaScript you can create an array of objects and assign data to it like this:

**Example**

var employees = [  
{ "firstName":"John" , "lastName":"Doe" },   
{ "firstName":"Anna" , "lastName":"Smith" },   
{ "firstName":"Peter" , "lastName": "Jones" }  
];

The first entry in the JavaScript object array can be accessed like this:

employees[0].firstName + " " + employees[0].lastName;

The returned content will be:

John Doe

The data can be modified like this:

employees[0].firstName = "Gilbert";

[Try it yourself »](http://www.w3schools.com/json/tryit.asp?filename=tryjson_objectarray)

In the next chapter you will learn how to convert a JSON text to a JavaScript object.

**JSON Files**

* The file type for JSON files is ".json"
* The MIME type for JSON text is "application/json"

**JSON HowTo**

[« Previous](http://www.w3schools.com/json/json_syntax.asp)

[Next Chapter »](http://www.w3schools.com/json/default.asp)

**Converting a JSON Text to a JavaScript Object**

One of the most common use of JSON is to fetch JSON data from a web server (as a file or as an HttpRequest), convert the JSON data to a JavaScript object, and then it uses the data in a web page.

For simplicity, this can be demonstrated by using a string as input (instead of a file).

**JSON Example - Object From String**

Create a JavaScript string containing JSON syntax:

var txt = '{ "employees" : [' +  
'{ "firstName":"John" , "lastName":"Doe" },' +  
'{ "firstName":"Anna" , "lastName":"Smith" },' +  
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';

Since JSON syntax is a subset of JavaScript syntax, the JavaScript function eval() can be used to convert a JSON text into a JavaScript object.

The eval() function uses the JavaScript compiler which will parse the JSON text and produce a JavaScript object. The text must be wrapped in parenthesis to avoid a syntax error:

var obj = eval ("(" + txt + ")");

Use the JavaScript object in your page:

**Example**

<p>  
First Name: <span id="fname"></span><br />   
Last Name: <span id="lname"></span><br />   
</p>   
  
<script>  
document.getElementById("fname").innerHTML = obj.employees[1].firstName   
document.getElementById("lname").innerHTML = obj.employees[1].lastName   
</script>

[Try it yourself »](http://www.w3schools.com/json/tryit.asp?filename=tryjson_eval)

**JSON Parser**

lamp  The eval() function can compile and execute any JavaScript. This represents a potential security problem.

It is safer to use a JSON parser to convert a JSON text to a JavaScript object. A JSON parser will recognize only JSON text and will not compile scripts.

In browsers that provide native JSON support, JSON parsers are also faster.

Native JSON support is included in newer browsers and in the newest ECMAScript (JavaScript) standard.

|  |  |
| --- | --- |
| **Web Browsers Support** | **Web Software Support** |
| * Firefox (Mozilla) 3.5 * Internet Explorer 8 * Chrome * Opera 10 * Safari 4 | * jQuery * Yahoo UI * Prototype * Dojo * ECMAScript 1.5 |

[Try it yourself »](http://www.w3schools.com/json/tryit.asp?filename=tryjson_parse)

For older browsers, a JavaScript library is available at <https://github.com/douglascrockford/JSON-js>

The JSON format was [originally specified by Douglas Crockford](http://developer.yahoo.com/yui/theater/video.php?v=crockford-json)