JSON Syntax Rules

JSON syntax is derived from JavaScript object notation syntax:

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays
- JSON Data - A Name and a Value
- 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:
"name":"John"

JSON names require double quotes. JavaScript names don't.

- JSON - Evaluates to JavaScript Objects
- The JSON format is almost identical to JavaScript objects.

In JSON, keys must be strings, written with double quotes:

JSON: *{ "name":"John" }*.

In JavaScript, keys can be strings, numbers, or identifier names:

JavaScript

{ name:"John" }

JSON Values

- In JSON, values must be one of the following data types:
- a string
- a number
- an object (JSON object)
- an array
- a boolean
- null

In JavaScript values can be all of the above, plus any other valid JavaScript expression, including:

- a function
- a date
- undefined

In JSON, string values must be written with double quotes:

JSON *{ "name":"John" }* In JavaScript, you can write string values with double or single quotes:

JavaScript *{ name:'John' }*

- JSON Uses JavaScript Syntax
- Because JSON syntax is derived from JavaScript object notation, very little extra software is needed to work with JSON within JavaScript.

With JavaScript you can create an object and assign data to it, like this:

Example let person = { name: "John", age: 31, city: "New York" };

JavaScript Arrays as JSON The same way JavaScript objects can be used as JSON, JavaScript arrays can also be used as JSON.

JSON values cannot be one of the following data types:

- a function
- a date
- undefined
- JSON Strings
- Strings in JSON must be written in double quotes.

Example *{ "name":"John" }*

JSON Numbers

Example

{ "age":30 }.

JSON Objects * *Values in JSON can be objects.*

Example *{"employee":{ "name":"John", "age":30, "city":"New York" }}*.

JSON Arrays Values in JSON can be arrays.

Example *{"employees":["John", "Anna", "Peter"]}*

JSON Booleans Values in JSON can be true/false.

Example *{ "sale":true }*.

JSON null Values in JSON can be null.

Example *{ "middlename":null }*.

A common use of JSON is to exchange data to/from a web server.

When receiving data from a web server, the data is always a string.

Parse the data with *JSON.parse()to convert to a JavaScript object*..

Example - Parsing JSON Imagine we received this text from a web server:

'{ "name":"John", "age":30, "city":"New York"}' **Use the JavaScript function JSON.parse() to convert text into a JavaScript object:**

let obj = JSON.parse('{ "name":"John", "age":30, "city":"New York"}');

Make sure the text is written in JSON format, or else you will get a syntax error...

Array as JSON *When using the JSON.parse() on a JSON derived from an array, the method will return a JavaScript array, instead of a JavaScript object.*

- A common use of JSON is to exchange data to/from a web server.
- When sending data to a web server, the data has to be a string.

Convert a JavaScript object into a string with JSON.stringify().

Stringify a JavaScript Object Imagine we have this object in JavaScript:

let obj = { name: "John", age: 30, city: "New York" };.

Use the JavaScript function JSON.stringify() to convert it into a string.

let myJSON = JSON.stringify(obj);.

- The result will be a string following the JSON notation.
- myJSON is now a string, and ready to be sent to a server:

Example let obj = { name: "John", age: 30, city: "New York" }; let myJSON = JSON.stringify(obj); document.getElementById("demo").innerHTML = myJSON; .

Stringify a JavaScript Array *It is also possible to stringify JavaScript arrays:*

Imagine we have this array in JavaScript:

let arr = ["John", "Peter", "Sally", "Jane"]; .

Use the JavaScript function JSON.stringify() to convert it into a string.

let myJSON = JSON.stringify(arr); .

- The result will be a string following the JSON notation.
- myJSON is now a string, and ready to be sent to a server:

Object Syntax Example { "name":"John", "age":30, "car":null }.

- **JSON objects are surrounded by curly braces {}.**
- **JSON objects are written in key/value pairs.**

- **Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null).**
- **Keys and values are separated by a colon.**
- **Each key/value pair is separated by a comma.**

Arrays as JSON Objects Example ["Ford", "BMW", "Fiat"].

Arrays in JSON are almost the same as arrays in JavaScript.

In JSON, array values must be of type string, number, object, array, boolean or null. In JavaScript, array values can be all of the above, plus any other valid JavaScript expression, including functions, dates, and undefined.

Arrays in JSON Objects Arrays can be values of an object property:

Example {"name":"John", "age":30, "cars":["Ford", "BMW", "Fiat"]}.