**1. What is JSON and why would I use it?**

**JSON** (JavaScript Object Notation) is a lightweight format that is used for data interchanging. It is based on a subset of JavaScript language (the way objects are built in JavaScript). Some JavaScript is not JSON, and some JSON is not JavaScript.

JSON stands for “JavaScript Object Notation”.

JSON is a simple data exchange format.  JSON means JavaScript Object Notation; it is language and platform independent.

JSON is derived from the JavaScript programming language, it is a natural choice to use as a data format in JavaScript. JSON, short for JavaScript Object Notation.

[JSON](https://en.wikipedia.org/wiki/JSON) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA.

JSON is a data format. It could be classified as a language, but not a programming language. Its relationship to JavaScript is that it shares its syntax (more or less) with a subset of JavaScript literals.

JSON is like XML in that it is used to structure data in a text format and is commonly used to exchange data over the Internet. JSON is not a markup language. JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write.

File extension of JSON is .json

There really is no substitute for JSON. Well supported, great tooling and native support for JSON in Javascript, numerous server-side languages and even new solutions for API’s like GraphQL still rely on JSON for data.

**2. List out the uses of JSON?**

Uses of JSON includes

* When writing application based on JavaScript it uses JSON, which includes browser extension and websites
* JSON is used for transmitting and serializing structured data over network connection
* JSON is mainly used to transfer data between server and web application
* Web service and API’s use JSON format to provide public data
* JSON can be used with modern programming language
* The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML.
* It is used while writing JavaScript based applications that includes browser extensions and websites.
* JSON format is used for serializing and transmitting structured data over network connection.
* It is primarily used to transmit data between a server and web applications.
* Web services and APIs use JSON format to provide public data.
* It can be used with modern programming languages.

**3. What are the limitations and uses of JSON?**

There are some:

* JSON is not suitable for handling very large and complex data. When the data gets complex with several nested and hierarchical structures, it becomes complex for human readability.
* JSON does not support the comments.
* It does not support to handle the multimedia formats like image or rich text format.

**4. Mention what are the drawbacks of JSON?**

Drawbacks of json are

* It does not contain type definition
* It lacks some sort of DTD

**5. What are the properties of JSON?**

These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.

An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

**6. Explain the structure of JSON?**

JSON is built on two structures:

* A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
* An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

**7. Mention what is the rule for JSON syntax rules? Give an example of JSON object?**

JSON syntax is a set of the JavaScript object notation syntax.

* Data is in name/value pairs
* Data is separated by comma
* Curly brackets hold objects
* Square bracket holds arrays

**8. What programming languages supported by JSON?**

JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.

**9. Which browser provides native JSON support?**

All modern browsers support native JSON encoding/decoding (Internet Explorer 8+, Firefox 3.1+, Safari 4+, and Chrome 3+). Basically, JSON.parse(str) will parse the JSON string in str and return an object, and JSON.stringify(obj) will return the JSON representation of the object obj.

**10. What is the correct JSON content type?**

The MIME media type for JSON text is **application/json**. The default encoding is UTF-8. (Source: RFC 4627).

**11. Are Javascript objects and JSON equivalent?**

People often assume all Javascript objects are JSON and that JSON is a Javascript object. This is incorrect.

In Javascript var x = {x:y} is **not JSON**, this is a **Javascript object**. The two are not the same thing. The JSON equivalent (represented in the Javascript language) would be var x = '{"x":"y"}'. x is an object of type **string** not an object in it's own right. To turn this into a fully fledged Javascript object you must first parse it, var x = JSON.parse('{"x":"y"}');, x is now an object but this is not JSON anymore.

Remember: **JSON (in Javascript) is a string!**

**12. What is JSON String?**

A string is a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.

A string is a sequence of Unicode code points wrapped with quotation marks (U+0022). All code points may be placed within the quotation marks except for the code points that must be escaped: quotation mark (U+0022), reverse solidus (U+005C), and the control characters U+0000 to U+001F. There are two-character escape sequence representations of some characters.

\” represents the quotation mark character (U+0022).  
\\ represents the reverse solidus character (U+005C).  
\/ represents the solidus character (U+002F).  
\b represents the backspace character (U+0008).  
\f represents the form feed character (U+000C).  
\n represents the line feed character (U+000A).  
\r represents the carriage return character (U+000D).  
\t represents the character tabulation character (U+0009).

To convert a string into a JSON array, you need to create a JSONObject object for each of your objects, and add those to your JSON array.

**13. What is the use of JSON Stringify?**

The JSON.stringify() method converts a JavaScript value to a JSON string, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.

**14. What does JSON**parse**do?**

The JSON.parse() method parses a JSON string, constructing the JavaScript value or object described by the string. An optional reviver function can be provided to perform a transformation on the resulting object before it is returned.

**15. How should I parse a JSON string in JavaScript?**

The standard way to parse JSON in JavaScript is JSON.parse()

The JSON API was introduced with ES5 (2011) and has since been implemented in >99% of browsers by market share, and Node.js. Its usage is simple:

const json = '{ "fruit": "pineapple", "fingers": 10 }';

const obj = JSON.parse(json);

console.log(obj.fruit, obj.fingers);

**16. Explain the difference between JSON.stringify() and JSON.parse()**

JSON.stringify() is to create a JSON string out of an object/array. They are the inverse of each other. JSON.stringify() serializes a JS object into a JSON string, whereas JSON.parse() will deserialize a JSON string into a JS object.

JSON.stringify turns a JavaScript object into JSON text and stores that JSON text in a string, eg:

var my\_object = {

key\_1: "some text",

key\_2: true,

key\_3: 5

};

var object\_as\_string = JSON.stringify(my\_object);

// "{"key\_1":"some text","key\_2":true,"key\_3":5}"

typeof(object\_as\_string);

// "string"

JSON.parse turns a string of JSON text into a JavaScript object, eg:

var object\_as\_string\_as\_object = JSON.parse(object\_as\_string);

// {key\_1: "some text", key\_2: true, key\_3: 5}

typeof(object\_as\_string\_as\_object);

// "object"

**17. Mention what is the role of JSON.stringify?**

JSON.stringify() converts an object into a JSON text and saves that JSON text in a string.

**18. Show how to parse JSON in JQuery?**

To parse JSON in JQuery, we will see the example

var json = '{"name": "Guru 99", "Description ": "Learn PHP Interactively with PHP Beginner Tutorials"}'

obj = $.parseJSON(json);

//alert(obj.name);

**19. Explain what is JSON objects?**

An object can be defined as an unordered set of name/value pairs.  An object in JSON starts with {left brace} and finish or ends with {right brace}.  Every name is followed by: (colon) and the name/value pairs are parted by, (comma).

An object is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

**20. What is JSON Syntax?**

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.

**21. What is JSON Value?**

A JSON value can be an object, array, number, string, true, false, or null.

In JSON, value holds some data. A value can be a string in double quotes, or a number, or true or false or null, or an object or an array. These structures can be nested.

Values in JSON must be one of the following data types:

* a string
* a number
* an object (JSON object)
* an array
* a boolean
* null

**22. What is JSON Array?**

An array structure is a pair of square bracket tokens surrounding zero or more values. An array is an ordered collection of values. An array begins with [ (left bracket) and ends with ] (right bracket). Values are separated by , (comma).

The values are separated by commas. The JSON syntax does not define any specific meaning to the ordering of the values. However, the JSON array structure is often used in situations where there is some semantics to the ordering.

**23. What is JSON data?**

JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. In JSON data is nothing but a information. It is used primarily to transmit data between a server and web application, as an alternative to XML.

**24. What is JSON Text?**

A JSON text is a sequence of tokens formed from Unicode code points that conforms to the JSON value grammar. The set of tokens includes six structural tokens, strings, numbers, and three literal name tokens.

The six structural tokens:  
[ U+005B left square bracket  
{ U+007B left curly bracket  
] U+005D right square bracket  
} U+007D right curly bracket  
: U+003A colon  
, U+002C comma

These are the three literal name tokens:  
true U+0074 U+0072 U+0075 U+0065  
false U+0066 U+0061 U+006C U+0073 U+0065  
null U+006E U+0075 U+006C U+006C

Insignificant whitespace is allowed before or after any token. Whitespace is any sequence of one or more of the following code points: character tabulation (U+0009), line feed (U+000A), carriage return (U+000D), and space (U+0020). Whitespace is not allowed within any token, except that space is allowed in strings.

**25. Explain how to transform JSON text to a JavaScript object?**

One of the common use of JSON is to collect JSON data from a web server as a file or HTTP request, and convert the JSON data to a JavaScript, ant then it avails the data in a web page.

**26. Mention which function is used to convert a JSON text into an object?**

To convert JSON text into an object “eval” function is used.

**27. What is JSON Formatter?**

The JSON Formatter & Validator helps debugging JSON data by formatting and validating JSON data so that it can easily be read by human beings.

**28. What is JSON Viewer?**

JSON Viewer – Convert JSON Strings to a Friendly Readable Format.

**29. What is JSON Validator?**

The JSON Validator helps debugging JSON data by formatting and validating JSON data so that it can easily be read by human beings.

**30. What is JSON Schema?**

JSON Schema is a specification for JSON based format for defining the structure of JSON data.

**31. What is Polyfill?**

The JSON object is not supported in older browsers. We can work around this by inserting a piece of code at the beginning of your scripts, allowing use of JSON object in implementations which do not natively support it (like Internet Explorer 6) is called Polyfill.

**32. What is toJSON() method in JSON?**

The toJSON() method returns a string representation of the Date object.

**33. What is serialization in Javascript?**

The serialize() method creates a URL encoded text string by serializing form values. You can select one or more form elements (like input and/or text area), or the form element itself. The serialized values can be used in the URL query string when making an AJAX request.

**34. What is serialization and deserialization in JSON?**

JSON is a format that encodes objects in a string. Serialization means to convert an object into that string, and deserialization is its inverse operation. When transmitting data or storing them in a file, the data are required to be byte strings, but complex objects are seldom in this format.

**35. What is**serialization**of an object?**

To serialize an object means to convert its state to a byte stream so that the byte stream can be reverted back into a copy of the object.

**36. Why do we use JSON in Android?**

JSON stands for JavaScript Object Notation.It is an independent data exchange format and is the best alternative for XML.

Android provides four different classes to manipulate JSON data. These classes are JSONArray,JSONObject,JSONStringer and JSONTokenizer.

**37.Can I use comments inside a JSON file? If so, how?**

No.

The JSON should all be data, and if you include a comment, then it will be data too.

You could have a designated data element called "\_comment" (or something) that would be ignored by apps that use the JSON data.

{

"\_comment": "comment text goes here...",

"glossary": {

"title": "example glossary",

"GlossDiv": {

"title": "S",

"GlossList": {

"GlossEntry": {

"ID": "SGML",

"SortAs": "SGML",

"GlossTerm": "Standard Generalized Markup Language",

"Acronym": "SGML",

"Abbrev": "ISO 8879:1986",

"GlossDef": {

"para": "A meta-markup language, used to create markup languages such as DocBook.",

"GlossSeeAlso": ["GML", "XML"]

},

"GlossSee": "markup"

}

}

}

}

}

**38. What are the differences between JSON and JSONP?**

**JSONP** is **JSON with padding**, that is, you put a string at the beginning and a pair of parenthesis around it. For example:

//JSON

{

"name": "stackoverflow",

"id": 5

}

//JSONP

func({

"name": "stackoverflow",

"id": 5

});

The result is that you can load the JSON as a script file. If you previously set up a function called func, then that function will be called with one argument, which is the JSON data, when the script file is done loading. This is usually used to allow for cross-site AJAX with JSON data. If you know that **example.com** is serving JSON files that look like the JSONP example given above, then you can use code like this to retrieve it, even if you are not on the **example.com** domain:

function func(json) {

alert(json.name);

}

var elm = document.createElement("script");

elm.setAttribute("type", "text/javascript");

elm.src = "http://example.com/jsonp";

document.body.appendChild(elm);

* JSONP is a simple way to overcome browser restrictions when sending JSON responses from different domains from the client. But the practical implementation of the approach involves subtle differences that are often not explained clearly. Here is a simple tutorial that shows JSON and JSONP side by side.
* **JSON:** JSON is a simple data format used for communication medium between different systems
* **JSONP:** It is a methodology for using that format with cross domain [ajax](https://career.guru99.com/top-50-ajax-interview-questions-answers/) requests while not being affected by same origin policy issue. JSONP stands for JSON with padding. It is a method used to bypass the cross-domain policies in web browsers. In other words, JSONP is the simple way to deal with browser restrictions when sending JSON responses from different domains from the client. JSONP stands for JSON with Padding. JSONP is a method for sending JSON data without worrying about cross-domain issues. JSONP does not use the XMLHttpRequest object. JSONP uses the <script> tag instead.

**39. Why must one use JSON over**[**XML**](https://career.guru99.com/xml-interview-questions/)**?**

* It is faster and lighter than XML as on the wire data format
* XML data is typeless while JSON objects are typed
* JSON types: Number, [Array](https://career.guru99.com/top-50-array-interview-questions-answers/), Boolean, String
* XML data are all string
* Data is readily available as JSON object is in your JavaScript
* Fetching values is as simple as reading from an object property in your JavaScript code
* JSON and XML used different formats. When compared both JSON is easy to write and use it applications then XML. The XML format can also be determined by the XML DTD or XML Schema (XSL) and can be tested.
* The JSON a data-exchange format which is getting more popular as the JavaScript applications possible format. Basically this is an object notation array. JSON has a very simple syntax so can be easily learned.
* It is faster and lighter than XML as on the wire data format
* XML data is typeless while JSON objects are typed
* JSON types: Number, Array, Boolean, String
* XML data are all string
* Data is readily available as JSON object is in your JavaScript
* Fetching values is as simple as reading from an object property in your JavaScript code

The fundamental difference, which no other answer seems to have mentioned, is that XML is a markup language (as it actually says in its name), whereas JSON is a way of representing objects (as also noted in its name). This is what makes markup languages so useful for representing documents.

**40. Mention what is JSON-RPC and JSON Parser?**

* **JSON RPC:** It is a simple remote procedure call protocol same as XML-RPC although it uses the lightweight JSON format instead of XML
* **JSON Parser:** JSON parser is used to parse the JSON data into objects to use its value. It can be parsed by javaScript, PHP and [jQuery](https://career.guru99.com/top-50-jquery-interview-questions/).

JSON parser to parse JSON object and MAINTAIN comments. By using JSON, when receiving data from a web server, the data should be always in a string format. We use JSON.parse() to parse the data and it becomes a JavaScript object.

The JSON.parse() method parses a JSON string, constructing the JavaScript value or object described by the string. An optional reviver function can be provided to perform a transformation on the resulting object before it is returned.

**41. What is Number in JSON?**

JSON Numbers – A number is very much like a C or Java number, except that the octal and hexadecimal formats are not used. A number is a sequence of decimal digits with no superfluous leading zero.

It may have a preceding minus sign (U+002D). It may have a fractional part prefixed by a decimal point (U+002E). It may have an exponent, prefixed by e (U+0065) or E (U+0045) and optionally + (U+002B) or – (U+002D). The digits are the code points U+0030 through U+0039.

Numeric values that cannot be represented as sequences of digits (such as Infinity and NaN) are not permitted.

**42. Can you write an example code in JSON?**

The following example shows how to use JSON to store information related to books based on their topic and edition.

{

"book": [

{

"id":"01",

"language": "Java",

"edition": "third",

"author": "Herbert Schildt"

},

{

"id":"07",

"language": "C++",

"edition": "second"

"author": "E.Balagurusamy"

}

]

}