### **JSON is a syntax for serializing objects, arrays, numbers, strings, booleans, and** [**null**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/null)**. It is based upon JavaScript syntax but is distinct from it: some JavaScript is *not* JSON.**

### **Objects and Arrays**

### **Property names must be double-quoted strings;** [**trailing commas**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Trailing_commas) **are forbidden.**

### **Numbers**

### **Leading zeros are prohibited. A decimal point must be followed by at least one digit. NaN and Infinity are unsupported.**

### **JSON Explained**

If we go further in depth, there are a few more things you need to know about JSON. The code that we just ran can also be expressed by including ‘[]’ to contain the above three nodes

JSonData: [

One: [11, 12, 13, 14]

Two: [22, 23, 24]

Three: [33, 34]

Then, in the following piece of code, a blend of the two looping techniques of For and For In are used to withdraw all the information from JSON.

### **1. The “While” Loop**

A While Loop also doesn’t have a counter and only runs if the specified condition is true. Here’s an example demonstrating the while loop where the loop runs till infinity:

var infiniteValue = true;

while (infiniteValue) {

console.log("Infinite")

}

**‍**

### **2. Do While Loop**

In this loop, there’s a defined condition that needs to be checked at the end of the process.

Here’s how you can use the Do While Loop in order to loop through XML(Extensible Markup Language):

### **3. The ForEach() Loop**

This method is used for looping through an array element. Here’s an example of this:

var names = ["jerry", "tom", "pluto", "micky", "mini"];

names.forEach(function1);

function function1(currentValue, index) {

console.log("Index in array is: "+index + " :: Value is: "+currentValue);

}

**‍**

### **Looping through JSON in more detail with examples**

#### **Example 1:**

We're making a list of books and we enter the title as “Lord of the Rings” and the year as “1994“and the count of pages as “3000”. Similarly, a large number of objects can be created and once it’s done, we will have a large database:

var books = [

{

title: "Lord of the Rings",

year: 1994,

pages: 3000

},

{

title: "1984",

year: 1984,

pages: 2000

},

{

title: "Art of war",

year: 2000,

pages: 5000

}

]

**‍**

For example;

alert(books.length);

**‍**

When we run this, the browser will display three. Essentially, with this, we checked that there are three objects in this array.

**Keep in mind that in JSON, you need to separate your objects with a comma as you can see in the example.**

#### **Example 2**

In this example, we are looping within the array.

for (i=0; i<books.length; i++){

document.write(

books.title + "<br />"

);

}

**‍**

Now, that should help you better understand how to loop through JSON. I've also created a video walkthrough of some examples in real-time to help you better understand this concept.

***JSON RESUME :-***

***{***

***“basics” : {***

*“Name” : “SANTHOSH KUMAR K”,*

*“Date of birth” : “1997/04/02”,*

*“Phone” : “7502221488”,*

*“Email id” : santhoshmech626@gmail.com”,*

*“Address” : “1 339”,”NOCHIKUTTAI VILLAGE”,”THALANATHAM POSY”,”PAPPIREDDIPATTI TALUK”,”DHARMAPURI DISTRICT”*

*“Pincode” : “635302”,*

*}*

*education{*

*“Qualification” : “BE(mechaical engineering)”,*

*“Class” : “first class”,*

*“Cgpa” : “8.12”,*

*“College”: “SNS COLLEGE OF TECHNOLOGY”,*

*“ college location” : “COIMBATORE”,*

Interest{

“Game”: “cricket ”,

“Travelling”: “bike ride”,

“Songs”: “yuvan songs”

}

Language known{

“Reading” : “tamil”, “english”,

“Speaking” : “tamil”, “english”,

“Writting”: “tamil”,”english”,

}

Declecaration{

“Decleration” : “ i hope all the given details 100% true and i will give the full support to work smart and hardwork ”,

}

Yours faithfully{

“Name”: “santhosh kumar k”,

}

***Differences between window, screen and documents:-***

* Window is the main JavaScript object root, aka the global object in a browser, and it can also be treated as the root of the document object model. You can access it as window.
* window.screen or just screen is a small information object about physical screen dimensions.
* window.document or just document is the main object of the potentially visible (or better yet: rendered) document object model/DOM.