**JSON**

**Naveena SS**

**JSON (Java Script Object Notation)**

JSON is a format for storing and transporting data.

It is used when data is sent from a server to a webpage.

It is language independent.

**JSON Syntax Rules**

* The JSON syntax is a subset of the JavaScript syntax.
* Data is in name/value pairs.
* A name/value pair consists of a field name (in double

quotes),followed by a colon, followed by a value.

* Data is separated by commas.
* Curly braces hold objects.
* Square brackets hold arrays.

**JSON Data Types**

In JSON, values must be one of the following data types:

* a string
* a number
* an object
* an array
* a boolean
* null

JSON values cannot be one of the following data types:

* a function
* a date
* undefined

**JSON.parse()**

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(), and the data becomes a JavaScript

object.

**Example:-**

***<!DOCTYPE html>***

***<html>***

***<body>***

***<p id="demo"></p>***

***<script>***

***const txt = '{"name":"Naveena", "age":25, "city":"Varkala"}'***

***const obj = JSON.parse(txt);***

***document.getElementById("demo").innerHTML = obj.name + ", " + obj.age;***

***</script>***

***</body>***

***</html>***

**Output:- Naveena,25**

**Exceptions**

**Parsing Dates and Parsing Functions**

Date objects and Functions are not allowed in JSON.

If you need to include a date or a function, write it as a string.

You can convert it back into a date object or a function later.

**JSON.stringify().**

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

When sending data to a web server, the data had to be a string.

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

**Example:-**

***<!DOCTYPE html>***

***<html>***

***<body>***

***<p id="demo"></p>***

***<script>***

***const obj = {name: "Naveena", age:25, city: "Varkala"};***

***const myJSON = JSON.stringify(obj);***

***document.getElementById("demo").innerHTML = myJSON;***

***</script>***

***</body>***

***</html>***

**Output:-** {"name":"Naveena","age":25,"city":"Varkala"}

**Exceptions**

**Stringify Dates and Functions**

In JSON, date objects are not allowed. The JSON.stringify() function will convert any dates into strings.

In JSON, functions are not allowed as object values.The JSON.stringify() function will remove any functions from a JavaScript object, both the key and the value. This can be omitted if you convert your functions into strings before running the JSON.stringify() function.

**JSON Object Literals**

This is a JSON string:

'{"name":"John", "age":30, "car":null}'

Inside the JSON string there is a JSON object literal:

{"name":"John", "age":30, "car":null}

JSON object literals are surrounded by curly braces {}.

JSON object literals contains key/value pairs.

Keys and values are separated by a colon.

You can create a JavaScript object from a JSON object literal:

**Example:-**

***<!DOCTYPE html>***

***<html>***

***<body>***

***<p id="demo"></p>***

***<script>***

***const myObj = {"name":"Naveena", "age":25, "car":null};***

***document.getElementById("demo").innerHTML = myObj.name;***

***</script>***

***</body>***

***</html>***

**Output:-Naveena**

**Accessing Object Values**

You can access object values by using dot (.) notation and by using bracket ([]) notation:

**Example:-**

const myJSON = '{"name":"Naveena", "age":25, "car":null}';

const myObj = JSON.parse(myJSON);

x = myObj.name;

x = myObj["name"];

**Output:-Naveena**

**JSON Array Literals**

This is a JSON string:

'["Ford", "BMW", "Fiat"]'

Inside the JSON string there is a JSON array literal:

["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.

You can create a JavaScript array from a literal:

**Example:-**

***<!DOCTYPE html>***

***<html>***

***<body>***

***<h2>Creating an Array from a Literal</h2>***

***<p id="demo"></p>***

***<script>***

***const myArray = ["Ford", "BMW", "Fiat"];***

***document.getElementById("demo").innerHTML = myArray;***

***</script>***

***</body>***

***</html>***

**Output:-** Ford,BMW,Fiat

**Accessing Array Values**

You access array values by index:

**Example:-**

const myJSON = '["Ford", "BMW", "Fiat"]';

const myArray = JSON.parse(myJSON);

document.getElementById("demo").innerHTML = myArray[0];

**Output:-** Ford