

CH-1 JSON

JavaScript object Notation

* Introduction

- JSON is an open standard for exchange data on the web. It supports data structures like object and array. so, it is easy to write & read data from JSON.
- JSON is a syntax for storing & exchanging data, it is text written with JS object notation.
- JSON is lightweight & self describing.
- JSON is language-independent, supports datastructure such as array & objects.

* Why JSON?

- Since JSON format is text only it can easily send to and from a server, and used as data format by any programming language.

→ Here data is in name / value pairs and they are separated by comma. It uses curly bracket {} to hold objects & [] to hold arrays.

ex { "FSO": [{ "name": "Trees", "course": "Intro of Trees", "content": ["Binary tree", "X"] }, { "name": "Graph", "Topics": ["BFS", "DFS", "X"] }] }

→ JSON Built-in function.

① JSON.parse():

↳ to convert JSON string to JS object.

② JSON.stringify():

↳ to convert object into a JSON string.

Note: `'{"name": "ABC", "age": 20}'` ← backtic
example of string.
→ JSON file name extension `.json`

Note: on server side parsing is most imp. part that developer wants. if parsing will be fast on server side then user can get a fast response fast-response.

★ JSON vs. XML:-

| JSON | XML |
|--|----------------------------------|
| → JavaScript object Notation | → extensible Markup language. |
| → data oriented | → document oriented |
| → it supports array | → doesn't support array. |
| → less secure than XML | → XML is more secure |
| → JSON has no tags | → XML is represented using tags. |
| → JSON contains string, boolean, number, array, objects. | → it is in string format. |
| → files are more human readable than XML. | → less human readable. |

- Basic datatypes supported by JSON
 - ① String = collection of char
 - ② Number = no could be int or decimal
 - ③ Boolean = true/false
 - ④ Null = No value to be specified.

ex { "student": [] } JSON file

```
{
  "name": "Raj",
  "age": 23
}
```

```
{
  "name": "ZMP",
  "age": 20
}
```

key is always
string

XML Example

```

<students>
  <student>
    <name>XYZ </name>
    <age>23 </age>
  </student>
  <student>
    <name>PQR </name>
    <age>25 </age>
  </student>
</students>
```

* Similarities betⁿ JSON & XML:

- both are simple & open
- both supports hierarchical structure.
- Both are language independent.
- Both are self describing as both XML data & JSON data are human readable text.

* JSON syntax:

- it derived from JS obj. notation syntax
 - data is in Name/Value pair
 - data is separated by comma
 - {} holds objects.
 - [] arrays.

* Name-value pair:

"name": "ABC" or "name": 'ABC'

* JSON Datatypes:

- | | |
|--|---|
| <ul style="list-style-type: none">- string- number- object (JSON)- array- boolean- Null | <ul style="list-style-type: none">- a function- date- undefined- net following |
|--|---|

⇒ JSON Strings:-
↳ string must be written in double quote

Ex { "name": "LJU" }

⇒ JSON Numbers:-

Ex { "age": 30 }

⇒ JSON Object:-

values in JSON can be object.

Ex { "employee": { "name": "LJU",
"age": 30 } }

⇒ JSON Array:-

Ex ["student": ["John", "Anna", "Peter"]]

⇒ JSON Boolean:-

Ex { "sale": true }

⇒ JSON null:-

Ex { "middlename": null }

~~★ JSON.parse():-~~
~~function for converting an object/string~~
~~into JSON string.~~

~~★ JSON_object: "How to access JSON
object value?"~~

~~Ex~~

~~<html>~~

~~<body>~~

~~<p> access JSON obj value </p>~~

~~<div id="myid">~~

~~<div>~~

~~<script>~~

~~var obj = {"name": "ZMP", "num": 123};~~

~~var name = obj.name;~~

~~var num = obj.num;~~

~~document.getElementById.innerHTML~~

~~= "name: " + name + "
 num: " + num;~~

~~</script>~~

~~</body>~~

~~</html>~~

O/P

Access JSON obj value

name : ZMP

num : 123

Q2

```
var name = obj["name"];
var num = obj["city"];
d.get... Id.innerHTML = "name:" + name;
```

* Looping JSON object: (for-in)

```
<html>
<body>
<div id="myid">
</div>
<script>
var obj = {"name": "ZMP", "num": 123};
for (var i in obj)
{
    document.getElementById("myid").  

    access { innerHTML += i + "<br>"  

    key } or document.write(i)  

    value } document.getElementById("myid").  

    access { innerHTML += obj[i] obj[i] + "<br>"  

    both key & value }  

    ↑  

    you can  

    put indexnum  

    for particular value
</script>
</body>
</html>
```

Nested JSON Objects

① How to access Nested JSON object.

Ex <body>

<div id="myid">

<div>

<script>

var obj = {

"name": "ZMP",

"courses": {

"course1": "FSD-2",

"course2": "COA" },

"city": "Ahmedabad"

}

C1 = obj.address.course1;

C2 = obj.address.course2;

document.getElementById("myid").

innerHTML = C1 + "," + C2;

</script>

</body>

② How to access Nested Object using Loop

for (var i in obj.courses)

{
to store value → xt = obj.courses[i] + "
"; we want to access inside object

document.getElementById("myid").innerHTML
= xt;

★ JSON Array:

① How to access JSON array values?

Ex

<body>

<div id="myid"></div>

<script>

var obj = {

"name": "ZMP",

"mobile": 94089,

"course": ["FSD-2", "JAVA", "COA"]

}

var c1 = obj.course[0];

var c2 = obj.course[1];

document.getElementById("myid").innerHTML

= c1 + ", " + c2;

</script>

</body>

② Using for-in Loop:

In above example.

var x = "";

for (var i in obj.course)

{ x += obj.course[i] + "
"

}

document.getElementById("myid").

innerHTML = x;

③ Using for loop:

```
ex <div id="myid">  
  </div>  
  <script>  
    var obj = {  
      "name": "xyz",  
      "marks": 23,  
      "course": ["PHP", "FSD-2", "XML"]  
    };  
    var x = "";  
    obj.course[0] = "COA"; // modify index  
    delete obj.course[1]; // delete index  
    for(var i=0; i<obj.course.length; i++)  
    {  
      x += obj.course[i] + "<br>";  
    }  
    document.getElementById("myid").  
    innerHTML = x;
```

```
obj.name = "ABC"; // modify key  
delete obj.name;  
do...get...Id("myid") = "name:"  
  + obj.name + "<br>";  
</script>  
#100
```

~~★~~ Array of object:

e.g. <body>

<script>

const a = {

"Datastructures": [

{

"name": "tree",

"course": "intro",

"content": ["I", "B", "C"]

},

{

"name": "Graph",

"course": "intro",

"content": ["I", "B", "C", "D"]

}

]

};

console.log(a.Datastructures[0].name)

// tree

console.log(a.xiz.name)

console.log(a.Datastructures[1]);

console.log(a.name); // undefined

</script>

</body>

* Array of object using for-in :

ex <script>

```
var jsonobj = {
    "FIFA": [
        {
            "country": "Brazil", "player": "xyz"
        },
        {
            "country": "portugal", "player": "PQR"
        },
        {
            "country": "argentina", "player": "ABC"
        }
    ]
}

for (var x in jsonobj.FIFA) // for index value
{
    for (var i in jsonobj.FIFA[x])
        console.log(jsonobj.FIFA[x][i]);
        ↑ show O/P in console
    document.write(jsonobj.FIFA[x][i]);
}

</script>
```

* JSON.Parse()
→ This function used to convert string to JSON object.

ex <body>
<h2>JSON parse </h2>
<div id="myid">
</div>
<script> `backtick
var obj = {
 "name": "xyz",
 "marks": 23,
 "course": ["FSO", "PHP", "JSON"]
};

var data = JSON.parse(obj);
var name = data.name;
var course1 = data.course[0];
var course2 = data.course[1];
var course3 = data.course[2];

① console.log(name); // xyz

② ^{or} document.write("Name = " + name); // xyz

③ ^{or} document.getElementById("myid").

innerHTML = "Name = " + name
 + "course = " + course1 + "course"
 + course2;
</script>

→ Date objects are directly not allowed in JSON. If you need to include date, write it as a string

Ex

```
<body>
  <div id="myid">
    <div>
      <script>
        var obj = {"name": "ZMP", "dob": "2023-4-10"};
        var data = JSON.parse(obj);
        data.dob = new Date(data.dob);
        document.getElementById("myid").innerHTML = "Name : " + data.name +
          " DOB : " + data.dob;
      </script>
    </div>
  </body>
```

Note casting to date obj is not mandatory. You can write directly to print date in string format as given. If you cast to Date obj then it will give IST (Indian Standard Time) format.

* JSON.stringify()

convert JS object to JSON string

Ex

```
<body>
<script>
var obj = {
    name : "ZMP",
    marks : 23,
    city : "xyz"
}
```

```
var data = JSON.stringify(obj);
document.write(data);
```

"to convert array"

```
var arr = ["xyz", "ABC", "PQR"]
```

```
var a = JSON.stringify(arr);
document.write(a);
```

```
</script>
</body>
```