

JSON

- JSON stands for JavaScript Object Notation.
- JSON is a lightweight format for storing and transporting data.
- JSON is often used when data is sent from a server to a web page
- JSON is self-describing and easy to understand.

WRITING JSON

JavaScript > Lecture 27 > JS FirstWritingJSON.js > ...

```
1 //npm init
2 // npm install minimist
3 // node FirstWritingJSON.js --dest=teams.json
4
5 let minimist = require("minimist");
6 let fs = require("fs");
7
8 let args = minimist(process.argv);
9 console.log(args.dest)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json teams.json
→ Lecture 27 git:(main) x
```

JS FirstWritingJSON.js U X

JavaScript > Lecture 27 > JS FirstWritingJSON.js > ...

```
11 // JSON = JavaScript Object Notation
12 // JSON means saving data in the same format as of javascript objects
13
14 let s1 = {
15   name: "Hemakshi",
16   age: 20
17 }; // an object
18
19 let s2 = {
20   name: "Morgan",
21   age: 22
22 }; // another object
23
24 let stdntsWay1 = [s1, s2]; // an array of object which is again an object = JSO (Javascript object)
25
26 let json = JSON.stringify(stdntsWay1); // JSO to JSON (Javascript object notation)
27 fs.writeFileSync(args.dest, json, "utf-8");
```

{ } teams.json U X

JavaScript > Lecture 27 > { } teams.json > ...

```
1 [{"name": "Hemakshi", "age": 20}, {"name": "Morgan", "age": 22}]
```

JSON.stringify() → a common use of JSON is to exchange data to/from a web server. When send data to a web server, the data has to be string. Convert a Javascript object into a string with JSON.stringify()

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json teams.json
→ Lecture 27 git:(main) x
```

```
JS FirstWritingJSON.js U x
Javascript > Lecture 27 > JS FirstWritingJSON.js > ...
11 // JSON = JavaScript Object Notation
12 // JSON means saving data in the same format as of javascript objects
13
14 let s1 = {
15     name: "Hemakshi",
16     age: 20
17 }; // an object
18
19 let s2 = {
20     name: "Morgan",
21     age: 22
22 }; // another object
23
24 let stdntsWay1 = [s1, s2]; // an array of object which is again an object = JSO (Javas
25
26 let json = JSON.stringify(stdntsWay1); // JSO to JSON (Javascript object notation)
27 fs.writeFileSync(args.dest, json, "utf-8");

JSON = JSON.stringify()
```

```
{ teams.json U x
Javascript > Lecture 27 > {} teams.json > ..
1 {}
2 {
3     "name": "Hemakshi",
4     "age": 20
5 },
6 {
7     "name": "Morgan",
8     "age": 22
9 }
10 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json
teams.json
→ Lecture 27 git:(main) x

```
JS FirstWritingJSON.js U x
Javascript > Lecture 27 > JS FirstWritingJSON.js > ...
3 // node FirstWritingJSON.js --dest=teams.json
4
5 let minimist = require("minimist");
6 let fs = require("fs");
7
8 let args = minimist(process.argv);
9 console.log(args.dest);
10
11 // JSON = JavaScript Object Notation
12 // JSON means saving data in the same format as of javascript objects
13
14 let s1 = {
15     name: "Hemakshi",
16     age: 20
17 }; // an object
18
19 let s2 = {
20     name: "Morgan",
21     age: 22
22 }; // another object
23
24 // let stdntsWay1 = [s1, s2]; // an array of object which is again an object = JSO (Ja
25
26 // let json = JSON.stringify(stdntsWay1); // JSO to JSON (Javascript object notation)
27 // fs.writeFileSync(args.dest, json, "utf-8");
28
29 console.log(s1.name);
30 console.log(s1.age);
31
```

→ s1 is an object. name and age are properties.
→ name and age are also called data members.

→ Accessing Object properties
ObjectName.propertyName

```
{ teams.json U x
Javascript > Lecture 27 > {} teams.json > ..
1 {}
2 {
3     "name": "Hemakshi",
4     "age": 20
5 },
6 {
7     "name": "Morgan",
8     "age": 22
9 }
10 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json
teams.json
Hemakshi
20
→ Lecture 27 git:(main) x

JavaScript Objects

The name : values pairs in JavaScript object are called properties.

```
let person = {
    firstName: "John",
    lastName: "Doe",
    age: 50,
    eyeColor: "blue"
};
```

Accessing Object Properties

you can access object properties in two ways:-

```
let person = { firstName: "John",  
               lastName: "Doe",  
               age: 50,  
               eyeColor: "blue"  
};
```

① `ObjectName.propertyName;`

eg → `person.lastName;`

② `ObjectName["propertyName"];`

eg → `person["lastName"];`

→ double quotes.

Array of Objects

```
32  
33 let ages_Arr = [10, 20, 30];  
34 console.log(ages_Arr[0]);  
35 console.log(ages_Arr[1]);  
36 console.log(ages_Arr[2]);  
37  
38 let names_Arr = ["Hemakshi", "Morgan", "Audrey"];  
39 console.log(names_Arr[0]);  
40 console.log(names_Arr[1]);  
41 console.log(names_Arr[2]);  
42  
43
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json
teams.json

10

20

30

Hemakshi

Morgan

Audrey

→ Lecture 27 git:(main) x

```

42  let arrOfObjects = [
43      {
44          name: "Hemakshi",
45          age: 20
46      },
47      {
48          name: "Morgan",
49          age: 22
50      },
51      {
52          name: "Audrey",
53          age: 21
54      }
55  ]
56
57  console.log(arrOfObjects[0].name);
58  console.log(arrOfObjects[0].age);
59
60  console.log(arrOfObjects[1].name);
61  console.log(arrOfObjects[1].age);
62
63  console.log(arrOfObjects[2].name);
64  console.log(arrOfObjects[2].age);
65

```

Handwritten annotations on the right side of the code block:

- A closing bracket `}` next to line 45, with a handwritten `0` to its right.
- A closing bracket `}` next to line 49, with a handwritten `1` to its right.
- A closing bracket `}` next to line 53, with a handwritten `2` to its right.

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

```

→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json
Hemakshi
20
Morgan
22
Audrey
21
→ Lecture 27 git:(main) x

```

JS FirstJSON.js U ×

JavaScript > Lecture 27 > JS FirstJSON.js > ...

```
2 // npm install minimist
3 // node FirstJSON.js
4
5 let minimist = require("minimist");
6 let fs = require("fs");
7
8 let args = minimist(process.argv);
9
10 // JSON = JavaScript Object Notation
11 // JSON means saving data in the same format as of javascript objects
12
13 let s1 = {
14     name: "Hemakshi",
15     age: 20
16 }; // s1 is an object. name and age are properties
17 // name and age are also called data memeber
18
19 console.log(s1.name);
20 console.log(s1.age);
21
22 let ages_Arr = [10, 20, 30];
23 console.log(ages_Arr[0]);
24 console.log(ages_Arr[1]);
25 console.log(ages_Arr[2]);
26
27 let names_Arr = ["Hemakshi", "Morgan", "Audrey"];
28 console.log(names_Arr[0]);
29 console.log(names_Arr[1]);
30 console.log(names_Arr[2]);
31
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 27 git:(main) x node FirstJSON.js
Hemakshi
20
10
20
30
Hemakshi
Morgan
Audrey
Hemakshi
20
Morgan
22
Audrey
21
```

JS FirstJSON.js U ×

JavaScript > Lecture 27 > JS FirstJSON.js > ...

```
31
32 let arrOf0bjects = [
33     {
34         name: "Hemakshi",
35         age: 20
36     },
37     {
38         name: "Morgan",
39         age: 22
40     },
41     {
42         name: "Audrey",
43         age: 21
44     }
45 ]
46
47 console.log(arrOf0bjects[0].name);
48 console.log(arrOf0bjects[0].age);
49
50 console.log(arrOf0bjects[1].name);
51 console.log(arrOf0bjects[1].age);
52
53 console.log(arrOf0bjects[2].name);
54 console.log(arrOf0bjects[2].age);
55
56
57
58
59
60
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

JS FirstWritingJSON.js U ×

JavaScript > Lecture 27 > JS FirstWritingJSON.js > teams > rank

```
1 // node FirstWritingJSON.js --dest=teams.json
2
3 let minimist = require("minimist");
4 let fs = require("fs");
5
6 let args = minimist(process.argv);
7
8 let teams = [
9     {
10         name: "India",
11         rank: 1
12     },
13     {
14         name: "Australia",
15         rank: 3
16     },
17     {
18         name: "England",
19         rank: 2
20     }
21 ];
22
23
24
25
26 let json = JSON.stringify(teams); // JSO -> JSON
27 fs.writeFileSync(args.dest, json, "utf-8");
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json
→ Lecture 27 git:(main) x
```

{ teams.json U ×

JavaScript > Lecture 27 > {} teams.json > {}

```
1 [
2     {
3         "name": "India",
4         "rank": 1
5     },
6     {
7         "name": "Australia",
8         "rank": 3
9     },
10    {
11        "name": "England",
12        "rank": 2
13    }
14 ]
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Handwritten annotations:

- A bracket on the left side of the `teams` array in the first editor points to the text **JSO**.
- A bracket on the right side of the `teams.json` array in the second editor points to the text **JSON**.
- An arrow points from the `JSON.stringify` line in the first editor to the `teams.json` editor, with the text **JSON.stringify** written along the arrow.
- An arrow points from the `fs.writeFileSync` line in the first editor to the `teams.json` editor, with the text **writing JSON** written along the arrow.

JS FirstWritingJSON.js U ×

JavaScript > Lecture 27 > JS FirstWritingJSON.js > [🔍] teams > [🔗] matches > [🔗] result

```
1 // node FirstWritingJSON.js --dest=teams.json
2
3 let minimist = require("minimist");
4 let fs = require("fs");
5
6 let args = minimist(process.argv);
7
8 let teams = [
9   {
10     name: "India",
11     rank: 1,
12     matches: [
13       {
14         vs: "England",
15         result: "Win"
16       },
17     ],
18     vs: "Australia",
19     result: "Win"
20   },
21   {
22     name: "Australia",
23     rank: 3,
24     matches: [
25       {
26         vs: "India",
27         result: "Loss"
28       },
29       {
30         vs: "England",
31         result: "Win"
32       }
33     ]
34   }
35 ];
```

arr of objects
→ JSO

JS FirstWritingJSON.js U ×

JavaScript > Lecture 27 > JS FirstWritingJSON.js > [🔍] teams > [🔗] matches > [🔗] result

```
25 },
26 {
27   name: "Australia",
28   rank: 3,
29   matches: [
30     {
31       vs: "India",
32       result: "Loss"
33     },
34     {
35       vs: "England",
36       result: "Win"
37     }
38   ]
39 },
40 ],
41 {
42   name: "England",
43   rank: 2,
44   matches: [
45     {
46       vs: "Australia",
47       result: "Loss"
48     },
49     {
50       vs: "India",
51       result: "Loss"
52     }
53   ]
54 },
55 ],
56 ];
57
58 let json = JSON.stringify(teams); // JSO → JSON
59 fs.writeFileSync(args.dest, json, "utf-8");
```

→ JSO

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE zsh +

JS FirstWritingJSON.js U ×

JavaScript > Lecture 27 > JS FirstWritingJSON.js > [🔍] teams > [🔗] matches > [🔗] result

```
7
8 let teams = [
9   {
10     name: "India",
11     rank: 1,
12     matches: [
13       {
14         vs: "England",
15         result: "Win"
16       },
17     ],
18     vs: "Australia",
19     result: "Win"
20   },
21   {
22     name: "Australia",
23     rank: 3,
24     matches: [
25       {
26         vs: "India",
27         result: "Loss"
28       },
29       {
30         vs: "England",
31         result: "Win"
32       }
33     ]
34   }
35 ];
```

JSO

{ } teams.json U ×

JavaScript > Lecture 27 > { } teams.json > ...

```
1 [
2   {
3     "name": "India",
4     "rank": 1,
5     "matches": [
6       {
7         "vs": "England",
8         "result": "Win"
9       },
10      {
11        "vs": "Australia",
12        "result": "Win"
13      }
14    ]
15  },
16  {
17    "name": "Australia",
18    "rank": 3,
19    "matches": [
20      {
21        "vs": "India",
22        "result": "Loss"
23      },
24      {
25        "vs": "England",
26        "result": "Win"
27      }
28    ]
29  },
30 ]
```

JSON

JSON.stringify →

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json

JS FirstWritingJSON.js U ×

JavaScript > Lecture 27 > JS FirstWritingJSON.js > ...
1 // node FirstWritingJSON.js --dest=teams.json
2
3 let minimist = require("minimist");
4 let fs = require("fs");
5
6 let args = minimist(process.argv);
7
8 let teams = [
9 {
10 name: "India",
11 rank: 1,
12 matches: [
13 {
14 vs: "England",
15 result: "Win"
16 },
17 {
18 vs: "Australia",
19 result: "Win"
20 }
21]
22 },
23 {
24 name: "Australia",
25 rank: 3,
26 matches: [
27 {
28 vs: "India",
29 result: "Loss"
30 },
31 {
32 vs: "England",
33 result: "Win"
34 }
35]
36 }
37]
38
39 let json = JSON.stringify(teams); // JSO -> JSON
40 fs.writeFileSync(args.dest, json, "utf-8");
41 console.log(teams[2].matches[1].vs);
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
→ Lecture 27 git:(main) x node FirstWritingJSON.js --dest=teams.json
India
→ Lecture 27 git:(main) x

JS FirstWritingJSON.js U ×

JavaScript > Lecture 27 > JS FirstWritingJSON.js > ...
29 {
30 matches: [
31 {
32 vs: "India",
33 result: "Loss"
34 }
35 },
36 {
37 vs: "England",
38 result: "Win"
39 }
40]
41 }
42 },
43 {
44 name: "England",
45 rank: 2,
46 matches: [
47 {
48 vs: "Australia",
49 result: "Loss"
50 },
51 {
52 vs: "India",
53 result: "Loss"
54 }
55]
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
→ Lecture 27 git:(main) x node FirstJSONRead.js --source=team.json
Reading of File failed :Error: ENOENT: no such file or directory, open 'team.json'
→ Lecture 27 git:(main) x

we want to access this property

READ JSON

JavaScript > Lecture 27 > JS FirstJSONRead.js > fs.readFile("utf-8") callback

```
1 // node FirstJSONRead.js --source=teams.json
2
3 let minimist = require("minimist");
4 let fs = require("fs");
5
6 let args = minimist(process.argv);
7
8 fs.readFile(args.source, "utf-8", function(err, json){
9   if(err){
10     console.log("Reading of File failed :"+ err);
11   }else{
12     console.log(json);
13   }
14 })
15
```

→ err

```
JS FirstJSONRead.js U X
Javascript > Lecture 27 > JS FirstJSONRead.js > fs.readFile("utf-8") callback
1 // node FirstJSONRead.js --source=teams.json
2
3 let minimist = require("minimist");
4 let fs = require("fs");
5
6 let args = minimist(process.argv);
7
8 fs.readFile(args.source, "utf-8", function(err, json){
9   if(err){
10     console.log("Reading of File failed :"+ err);
11   }else{
12     // Conversion of JSON back to JSO
13     let teams = JSON.parse(json); // JSON to JSO so you we can manipulate the object
14     console.log(teams[2].matches[1].vs);
15   }
16 })
17
```

```
{ teams.json U X
Javascript > Lecture 27 > {} teams.json > {} 2 > [ ] n
29
30 {
31   "name": "England",
32   "rank": 2,
33   "matches": [
34     {
35       "vs": "Australia",
36       "result": "Loss"
37     },
38     {
39       "vs": "India",
40       "result": "Loss"
41     }
42   ]
43 }
44
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture 27 git:(main) x node FirstJSONRead.js --source=teams.json
India
→ Lecture 27 git:(main) x

```
JS FirstJSONRead.js U X
Javascript > Lecture 27 > JS FirstJSONRead.js > ...
1 // node FirstJSONRead.js --source=teams.json
2
3 let minimist = require("minimist");
4 let fs = require("fs");
5
6 let args = minimist(process.argv);
7
8 fs.readFile(args.source, "utf-8", function(err, json){
9   if(err){
10     console.log("Reading of File failed :"+ err);
11   }else{
12     // Conversion of JSON back to JSO
13     let teams = JSON.parse(json); // JSON to JSO so you we can manipulate the object
14     console.log(teams[2].matches[1].vs);
15   }
16 })
17
18 // JSO
19 // if you want to print or save a JSO, convert the JSO to JSON using JSON.stringify()
20 // if you want to manipulate a JSON, convert the JSON to JSO using JSON.parse()
21
```

JSON.parse() method
parses string and returns
a JavaScript Object. The
string has to be written
in JSON format

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture 27 git:(main) x node FirstJSONRead.js --source=teams.json
India
→ Lecture 27 git:(main) x

- If you want to print or save a JSO, convert the JSO to JSON using JSON.stringify().
- If you want to manipulate a JSON, convert the JSON to JSO using JSON.parse().