# 30 Days Of JavaScript: JSON

LinkedIn Follow @asabeneh

Author: Asabeneh Yetayeh

January, 2020

&lt;&lt; Day 15 | Day 17 &gt;&gt;

Thirty Days Of JavaScript

# Day 16

## JSON

JSON stands for JavaScript Object Notation. The JSON syntax is derived from JavaScript object notation syntax, but the JSON format is text or string only. JSON is a light weight data format for storing and transporting. JSON is mostly used when data is sent from a server to a client. JSON is an easier-to-use alternative to XML.

**Example:**

```
{
"users":[
  {
    "firstName":"Asabeneh",
    "lastName":"Yetayeh",
    "age":250,
    "email":"asab@asb.com"
  },
  {
    "firstName":"Alex",
    "lastName":"James",
    "age":25,
    "email":"alex@alex.com"
  },
  {
  "firstName":"Lidiya",
  "lastName":"Tekle",
  "age":28,
  "email":"lidiya@lidiya.com"
  }
]
}
```

The above JSON example is not much different from a normal object. Then, what is the difference ? The difference is the key of a JSON object should be with double quotes or it should be a string. JavaScript Object and JSON are very similar that we can change JSON to Object and Object to JSON.

Let us see the above example in more detail, it starts with a curly bracket. Inside the curly bracket, there is "users" key which has a value array. Inside the array we have different objects and each objects has keys, each keys has to have double quotes. For instance, we use "firstNaMe" instead of just firstName, however in object we use keys without double quotes. This is the major difference between an object and a JSON. Let's see more examples about JSON.

**Example:**

```
  {
      "Alex": {
          "email": "alex@alex.com",
          "skills": [
              "HTML",
              "CSS",
              "JavaScript"
          ],
          "age": 20,
          "isLoggedIn": false,
          "points": 30
      },
      "Asab": {
          "email": "asab@asab.com",
          "skills": [
              "HTML",
              "CSS",
              "JavaScript",
              "Redux",
              "MongoDB",
              "Express",
              "React",
              "Node"
          ],
            "age": 25,
            "isLoggedIn": false,
            "points": 50
        },
        "Brook": {
            "email": "daniel@daniel.com",
            "skills": [
                "HTML",
                "CSS",
                "JavaScript",
                "React",
                "Redux"
            ],
            "age": 30,
            "isLoggedIn": true,
            "points": 50
        },
        "Daniel": {
            "email": "daniel@alex.com",
            "skills": [
                "HTML",
                "CSS",
                "JavaScript",
                "Python"
            ],
            "age": 20,
            "isLoggedIn": false,
            "points": 40
        },
        "John": {
            "email": "john@john.com",
            "skills": [
                "HTML",
                "CSS",
                "JavaScript",
                "React",
                "Redux",
                "Node.js"
            ],
            "age": 20,
            "isLoggedIn": true,
            "points": 50
        },
        "Thomas": {
            "email": "thomas@thomas.com",
            "skills": [
                "HTML",
                "CSS",
                "JavaScript",
                "React"
            ],
            "age": 20,
            "isLoggedIn": false,
        "points": 40
    },
    "Paul": {
        "email": "paul@paul.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "MongoDB",
            "Express",
            "React",
            "Node"
        ],
        "age": 20,
        "isLoggedIn": false,
        "points": 40
    }
}
```

## Converting JSON to JavaScript Object

Mostly we fetch JSON data from HTTP response or from a file, but we can store the JSON as a string and we can change to Object for sake of demonstration. In JavaScript the keyword *JSON* has *parse()* and *stringify()* methods. When we want to change the JSON to an object we parse the JSON using *JSON.parse()*. When we want to change the object to JSON we use *JSON.stringify()*.

### JSON.parse()

```
JSON.parse(json[, reviver])
// json or text , the data
// reviver is an optional callback function
/* JSON.parse(json, (key, value) => {

})
*/
```

```
const usersText = `{
"users":[
  {
    "firstName":"Asabeneh",
    "lastName":"Yetayeh",
    "age":250,
    "email":"asab@asb.com"
  },
  {
    "firstName":"Alex",
    "lastName":"James",
    "age":25,
    "email":"alex@alex.com"
  },
  {
  "firstName":"Lidiya",
  "lastName":"Tekle",
  "age":28,
  "email":"lidiya@lidiya.com"
  }
]
}`

const usersObj = JSON.parse(usersText, undefined, 4)
console.log(usersObj)
```

## Using a reviver function with JSON.parse()

To use the reviver function as a formatter, we put the keys we want to format first name and last name value. Let us say, we are interested to format the firstName and lastName of the JSON data .

```
const usersText = `{
"users":[
  {
    "firstName":"Asabeneh",
    "lastName":"Yetayeh",
    "age":250,
    "email":"asab@asb.com"
  },
  {
    "firstName":"Alex",
    "lastName":"James",
    "age":25,
    "email":"alex@alex.com"
  },
  {
  "firstName":"Lidiya",
  "lastName":"Tekle",
  "age":28,
  "email":"lidiya@lidiya.com"
  }
]
}`

const usersObj = JSON.parse(usersText, (key, value) => {
  let newValue =
    typeof value == 'string' && key != 'email' ? value.toUpperCase() : value
  return newValue
})
console.log(usersObj)
```

The *JSON.parse()* is very handy to use. You do not have to pass optional parameter, you can just use it with the required parameter and you will achieve quite a lot.

## Converting Object to JSON

When we want to change the object to JSON we use *JSON.stringify()*. The stringify method takes one required parameter and two optional parameters. The replacer is used as filter and the space is an indentations. If we do not want to filter out any of the keys from the object we can just pass undefined.

```
JSON.stringify(obj, replacer, space)
// json or text , the data
// reviver is an optional callback function
```

Let us convert the following object to a string. First let use keep all the keys and also let us have 4 space indentation.

```
const users = {
  Alex: {
    email: 'alex@alex.com',
    skills: ['HTML', 'CSS', 'JavaScript'],
    age: 20,
    isLoggedIn: false,
    points: 30
  },
  Asab: {
    email: 'asab@asab.com',
    skills: [
      'HTML',
      'CSS',
      'JavaScript',
      'Redux',
      'MongoDB',
      'Express',
      'React',
      'Node'
    ],
    age: 25,
    isLoggedIn: false,
    points: 50
  },
  Brook: {
    email: 'daniel@daniel.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'React', 'Redux'],
    age: 30,
    isLoggedIn: true,
    points: 50
  },
  Daniel: {
    email: 'daniel@alex.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'Python'],
    age: 20,
    isLoggedIn: false,
    points: 40
  },
  John: {
    email: 'john@john.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'React', 'Redux', 'Node.js'],
    age: 20,
    isLoggedIn: true,
    points: 50
  },
  Thomas: {
    email: 'thomas@thomas.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'React'],
    age: 20,
    isLoggedIn: false,
    points: 40
  },
  Paul: {
    email: 'paul@paul.com',
    skills: [
      'HTML',
      'CSS',
      'JavaScript',
      'MongoDB',
      'Express',
      'React',
      'Node'
    ],
    age: 20,
    isLoggedIn: false,
    points: 40
  }
}

const txt = JSON.stringify(users, undefined, 4)
console.log(txt) // text means JSON- because json is a string form of an object.
```

```
{
    "Alex": {
        "email": "alex@alex.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript"
        ],
        "age": 20,
        "isLoggedIn": false,
        "points": 30
    },
    "Asab": {
        "email": "asab@asab.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "Redux",
            "MongoDB",
            "Express",
            "React",
            "Node"
        ],
        "age": 25,
        "isLoggedIn": false,
        "points": 50
    },
    "Brook": {
        "email": "daniel@daniel.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "React",
            "Redux"
        ],
        "age": 30,
        "isLoggedIn": true,
        "points": 50
    },
    "Daniel": {
        "email": "daniel@alex.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "Python"
        ],
        "age": 20,
        "isLoggedIn": false,
        "points": 40
    },
    "John": {
        "email": "john@john.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "React",
            "Redux",
            "Node.js"
        ],
        "age": 20,
            "isLoggedIn": true,
            "points": 50
        },
        "Thomas": {
            "email": "thomas@thomas.com",
            "skills": [
                "HTML",
                "CSS",
                "JavaScript",
                "React"
            ],
            "age": 20,
            "isLoggedIn": false,
            "points": 40
        },
        "Paul": {
            "email": "paul@paul.com",
            "skills": [
                "HTML",
                "CSS",
                "JavaScript",
                "MongoDB",
                "Express",
                "React",
                "Node"
            ],
            "age": 20,
            "isLoggedIn": false,
            "points": 40
        }
    }
```

## Using a Filter Array with JSON.stringify

Now, lets use the replacer as a filter. The user object has long list of keys but we are interested only in few of them. We put the keys we want to keep in array as show in the example and use it the place of the replacer.

```
const user = {
  firstName: 'Asabeneh',
  lastName: 'Yetayeh',
  country: 'Finland',
  city: 'Helsinki',
  email: 'alex@alex.com',
  skills: ['HTML', 'CSS', 'JavaScript', 'React', 'Python'],
  age: 250,
  isLoggedIn: false,
  points: 30
}
```

```
const txt = JSON.stringify(user,['firstName', 'lastName', 'country', 'city', 'age'],4)
console.log(txt)
```

```
{
    "firstName": "Asabeneh",
    "lastName": "Yetayeh",
    "country": "Finland",
    "city": "Helsinki",
    "age": 250
}
```

🌕 You are extraordinary. Now, you knew a light-weight data format which you may use to store data or to send it an HTTP server. You are 16 steps a head to your way to greatness. Now do some exercises for your brain and for your muscle.

## Exercises

```
const skills = ['HTML', 'CSS', 'JS', 'React','Node', 'Python']
let age = 250;
let isMarried = true
const student = {
  firstName:'Asabeneh',
  lastName:'Yetayehe',
  age:250,
  isMarried:true,
  skills:['HTML', 'CSS', 'JS', 'React','Node', 'Python', ]
}
const txt = `{
    "Alex": {
        "email": "alex@alex.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript"
        ],
        "age": 20,
        "isLoggedIn": false,
        "points": 30
    },
    "Asab": {
        "email": "asab@asab.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "Redux",
            "MongoDB",
            "Express",
            "React",
            "Node"
        ],
        "age": 25,
        "isLoggedIn": false,
        "points": 50
    },
    "Brook": {
        "email": "daniel@daniel.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "React",
            "Redux"
        ],
        "age": 30,
        "isLoggedIn": true,
        "points": 50
    },
    "Daniel": {
        "email": "daniel@alex.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "Python"
        ],
        "age": 20,
        "isLoggedIn": false,
        "points": 40
    },
    "John": {
        "email": "john@john.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "React",
            "Redux",
            "Node.js"
        ],
        "age": 20,
        "isLoggedIn": true,
        "points": 50
    },
    "Thomas": {
        "email": "thomas@thomas.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "React"
        ],
        "age": 20,
        "isLoggedIn": false,
        "points": 40
    },
    "Paul": {
        "email": "paul@paul.com",
        "skills": [
            "HTML",
            "CSS",
            "JavaScript",
            "MongoDB",
            "Express",
            "React",
            "Node"
        ],
        "age": 20,
        "isLoggedIn": false,
        "points": 40
    }
}
`
```

## Exercises Level 1

1. Change skills array to JSON using JSON.stringify()
2. Stringify the age variable
3. Stringify the isMarried variable
4. Stringify the student object

## Exercises Level 2

1. Stringify the students object with only firstName, lastName and skills properties

## Exercises Level 3

1. Parse the *txt* JSON to object.
2. Find the user who has many skills from the variable stored in *txt*.

🎉 CONGRATULATIONS ! 🎉

&lt;&lt; Day 15 | Day 17 &gt;&gt;