# JSON.stringify()

- description : JSON.stringify()
- author : 오션
- email : shlim@repia.com
- lastupdate : 2022-04-06 Wed

## The source of this article

JSON.stringify()

JSON의 일반적인 용도는 웹 서버와 데이터를 교환하는 것입니다.

웹 서버에 데이터를 보낼 때, 데이터는 문자열이어야 합니다.

**`JSON.stringify()`**를 사용하여 JavaScript 객체를 문자열로 변환합니다.

## Stringify a JavaScript Object

아래와 같이 자바스크립트로 작성된 객체를 가지고 있다고 가정합니다.

```
const obj = {name: "John", age: 30, city: "New York"};
```

JavaScript 함수 **`JSON.stringify()`**를 사용하여 객체를 문자열로 변환합니다.

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

```
결과는 JSON 표기법을 따르는 문자열입니다.
```

**`myJSON`**은 이제 문자열이며 서버로 보낼 준비가 되었습니다:

### Example

```
<body>
```

```
    <h2>Create a JSON string form a JavaScript object.</h2>
    <hr>
    <p id="demo"></p>

    <script>
        const obj = {name: "Yejin", age: 34, city: "Pohang"};
        const myJSON = JSON.stringify(obj);
        document.getElementById("demo").innerHTML = myJSON;
        console.log(myJSON);           //
{"name":"Yejin","age":34,"city":"Pohang"}
        console.log(typeof myJSON);    // string
    </script>

</body>
```

## Stringify a JavaScript Array

JavaScript 배열을 문자열화하는 것도 가능합니다:

JavaScript에 다음 배열이 있다고 가정합니다.

```
const arr = ["John", "Peter", "Sally", "Jane"];
```

자바사크립트 함수 **JSON.stringify()**를 사용하여 배열을 문자열로 변환합니다.

```
const myJSON = JSON.stringify(arr);
```

```
결과는 JSON 표기법을 따르는 문자열입니다.
```

**myJSON**은 이제 문자열이고, 서버에 보내질 준비가 되었습니다:

### Example

```
<body>

    <h2>Create a JSON string from a JavaScript object.</h2>
    <hr>
    <p id="demo"></p>
```

```
    <script>
        const arr = ["John", "Peter", "Sally", "Jane"];
        const myJSON = JSON.stringify(arr);
        document.getElementById("demo").innerHTML = myJSON;
        console.log(arr); // ['John', 'Peter', 'Sally', 'Jane']
        console.log(typeof arr);    // object
        console.log(myJSON); // ["John","Peter","Sally","Jane"]
        console.log(typeof myJSON); // string
    </script>

</body>
```

## Storing Data

데이터를 저장할 때, 데이터는 특정 형식이어야 하며
저장 위치에 관계없이 텍스트는 항상 유효한 형식 중 하나입니다.

JSON을 사용하면 JavaScript 객체를 텍스트로 저장할 수 있습니다.

### Example

로컬 저장소에 데이터 저장하기

```
<body>

    <h2>Store and retrieve data from local storage.</h2>
    <hr>
    <p id="demo"></p>

    <script>
        // Storing data:
        const myObj = {name: "John", age: 31, city: "New York"};
        const myJSON = JSON.stringify(myObj);
        localStorage.setItem("testJSON", myJSON);

        // Retrieving data:
        let text = localStorage.getItem("testJSON");
        let obj = JSON.parse(text);
        document.getElementById("demo").innerHTML = obj.name;
        console.log(localStorage);
        /* Storage {testJSON: '{"name":"John","age":31,"city":"New York"}',
length: 1}
        testJSON: "{\"name\":\"John\",\"age\":31,\"city\":\"New York\"}"
        length: 1
```

```
        [[Prototype]]: Storage */
    </script>

</body>
```

# Exceptions

## Stringify Dates

JSON에서는 date 객체가 허용되지 않습니다. **JSON.stringify()** 함수는 모든 날짜를 문자열로 변환합니다.

## Example

```
<body>

    <h2>JSON.stringify() converts date objects into strings.</h2>
    <hr>
    <p id="demo"></p>

    <script>
        const obj = {name:"John", today: new Date(), city: "New York"};
        const myJSON = JSON.stringify(obj);
        document.getElementById("demo").innerHTML = myJSON;
        console.log(obj); // {name: 'John', today: Wed Apr 06 2022 14:21:07
GMT+0900 (한국 표준시), city: 'New York'}
        console.log(typeof obj); // object
        console.log(myJSON); //
{"name":"John","today":"2022-04-06T05:21:31.814Z","city":"New York"}
        console.log(typeof myJSON); // string
    </script>

</body>
```

수신측에서 문자열을 다시 date 객체로 변환할 수 있습니다.

## Stringify Functions

JSON에서, 함수는 객체의 value로 허용되지 않습니다.

**JSON.stringify()** 함수는 자바스크립트 객체에서 모든 함수, 즉 key와 value 모두를 제거합니다.

## Example

```
<body>

    <h2>JSON.stringify() will remove any function from an object.</h2>
    <hr>
    <p id="demo"></p>

    <script>
        const obj = {name:"John", age:function() {return 31;}, city: "New
York"};
        const myJSON = JSON.stringify(obj);
        document.getElementById("demo").innerHTML = myJSON; //
{"name":"John","city":"New York"}
        console.log(obj); // {name: 'John', city: 'New York', age: f}
        console.log(myJSON); // {"name":"John","city":"New York"}
    </script>

</body>
```

**JSON.stringify()** 함수를 실행하기 전에 함수를 문자열로 변환하는 경우 생략할 수 있습니다.

```
<body>

    <h2>JSON.stringify() will remove any function from an object.</h2>
    <hr>
    <p id="demo"></p>

    <script>
        const obj = {name:"John", age:function() {return 31;}, city: "New
York"};
        obj.age = obj.age.toString();
        console.log(obj.age); // function() {return 31;}
        console.log(typeof obj.age);    // string
        console.log(obj); // {name: 'John', age: 'function() {return 31;}',
city: 'New York'}
        const myJSON = JSON.stringify(obj);
        document.getElementById("demo").innerHTML = myJSON; //
{"name":"John","age":"function() {return 31;}","city":"New York"}
    </script>

</body>
```

JSON을 사용하여 함수를 보내면, 함수는 범위를 잃게 되며 수신자는 `eval()`을 사용하여 함수로 다시 변환해야 합니다.\\

오션,, JSON.stringify()

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
**http://125.132.25.164/dokuwiki/doku.php?id=wiki:javascript:javascript_note:json_stringfy&rev=1649226704**

Last update: **2022/04/06 15:31**