JSON.parse()

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

The source of this article

JSON.parse()

Valid Data Types

JSON의 일반적인 용도는 웹 서버와 데이터를 교환하는 것입니다.

웹 서버에서 데이터를 수신할 때, 데이터는 항상 문자열입니다.

JSON.parse()로 데이터를 구문분석하면, 데이터는 JavaScript 객체가 됩니다.

Example - Parsing JSON

웹 서버에서 아래의 텍스트를 받았다고 가정합니다.

```
'{"name":"John", "age":30, "city":"New York"}'
```

자바스크립트 함수 **JSON.parse()**를 사용하여 텍스트를 자바스크립트 객체로 변환합니다.

```
const obj = JSON.parse('{"name":"John", "age":30, "city":"New York"}');
```

```
텍스트가 JSON 형식인지 확인하십시오. 그렇지 않으면 구문 오류가 발생합니다.
```

웹 페이지에서 자바스크립트 객체를 사용합니다.

Example

```
<body>

    <h2>Creating an object from a JSON String</h2>

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

    <script>
        const txt = '{"name":"Dooli", "age": 350, "city": "Seoul"}';
        const parseTxt = JSON.parse(txt);
        document.getElementById("demo").innerHTML = parseTxt.name + ", " +
parseTxt.age;   // Dooli, 350
        console.log(parseTxt);   // {name: 'Dooli', age: 350, city: 'Seoul'}
    </script>

</body>
```

Array as JSON

배열에서 파생된 JSON에서 **JSON.parse()**를 사용할 때,
메서드는 JavaScript 객체 대신 JavaScript 배열을 반환합니다.

Example

```
<body>

    <h2>Parsing a JSON Array</h2>
    <p>Data written as an JSON array will be parsed into a JavaScript
array.</p>
    <p id="demo"></p>

    <script>
        const text = '["Ford", "BMW", "Audi", "Fiat"]';
        const textArray = JSON.parse(text);
        document.getElementById("demo").innerHTML = textArray[2];   // Audi
        console.log(textArray); //  ['Ford', 'BMW', 'Audi', 'Fiat']

    </script>

</body>
```

Exceptions

Parsing Dates

Date 객체는 JSON에서 허용되지 않습니다.

date를 포함시켜야 할 경우, date를 문자열로 작성합니다.

이후 해당 문자열을 date 객체로 변환할 수 있습니다.

Example

문자열을 date로 변환합니다.

```
<body>

    <h2> Convert a string into a date obejct</h2>
    <p id="demo"></p>

    <script>

        const text = '{"name":"John", "birth":"1986-12-14", "city":"New
York"}';
        const textObj = JSON.parse(text);
        textObj.birth = new Date(textObj.birth);
        document.getElementById("demo").innerHTML = textObj.name + " ," +
textObj.birth;
        // John ,Sun Dec 14 1986 09:00:00 GMT+0900 (한국 표준시)
        console.log(typeof textObj.birth); // object

    </script>

</body>
```

또는 *reviver*라고 하는 **JSON.parse()** 함수의 두 번째 매개변수를 사용할 수 있습니다.

reviver 매개변수는 값을 반환하기 전에 각 속성을 확인하는 함수입니다.

Example

reviver 함수를 사용하여 문자열을 date로 변환합니다.

```
<body>

    <h2> Convert a string into a date obejct</h2>
```

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

    <script>

        const text = '{"name":"John", "birth":"1986-12-14", "city":"New
York"}';
        const obj = JSON.parse(text, function (key, value) {
            if (key == "birth") {
                return new Date(value);
            } else {
                return value;
            }
        });
        document.getElementById("demo").innerHTML = obj.name + " ," +
obj.birth;
        // John ,Sun Dec 14 1986 09:00:00 GMT+0900 (한국 표준시)
        console.log(obj); // {name: 'John', birth: Sun Dec 14 1986 09:00:00
GMT+0900 (한국 표준시), city: 'New York'}
    </script>

</body>
```

Parsing Functions

JSON에서 함수는 허용되지 않습니다.

함수를 포함시켜야 할 경우, 함수를 문자열로 작성합니다.

이후 문자열로 작성된 함수를 함수로 변환할 수 있습니다.

Example

문자열을 함수로 변환합니다.

```
<body>

    <h2> Convert a string into a function</h2>
    <p id="demo"></p>

    <script>
        const text = '{"name":"John", "age":"function() {return 30;}",
"city":"New York"}';
        const obj = JSON.parse(text);
        obj.age = eval("(" + obj.age + ")");
        document.getElementById("demo").innerHTML = obj.name + ", " +
obj.age(); // John, 30
```

```
        console.log(obj.age); // f () {return 30;}
    </script>

</body>
```

JSON에서 함수 사용을 피해야 합니다.
함수는 자신의 범위를 잃게되며, eval()을 사용하여 다시 함수로 변환해야 합니다.

오션,, JSON.parse()

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_parse&rev=1649217074

Last update: **2022/04/06 12:51**