

07. JSON Parse.

7.1. JSON.parse()

A common use of JSON is to exchange data to/from a web server.

When receiving data from a web server, the data is always a string.

Parse the data with `JSON.parse()`, and the data becomes a JavaScript object.

Example: Parsing JSON. Imagine we received this text from a web server:

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

Use the JavaScript function `JSON.parse()` to convert text into a JavaScript object:

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

Make sure the text is written in JSON format, or else you will get a syntax error.

Use the JavaScript object in your page:

Example:

```
<p id="demo"></p>  
  
<script>  
document.getElementById("demo").innerHTML = obj.name +  
", " + obj.age;  
</script>
```

7.2. JSON From the Server

You can request JSON from the server by using an AJAX request.

As long as the response from the server is written in JSON format, you can parse the string into a JavaScript object.

Example: Use the XMLHttpRequest to get data from the server:

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        var myObj = JSON.parse(this.responseText);
        document.getElementById("demo").innerHTML = myObj.name;
    }
};
xmlhttp.open("GET", "json_demo.txt", true);
xmlhttp.send();
```

Take a look at **json_demo.txt**:

```
{
  "name": "John",
  "age": 31,
  "pets": [
    {
      "animal": "dog",
      "name": "Fido"
    },
    {
      "animal": "cat",
      "name": "Felix"
    },
    {
      "animal": "hamster",
      "name": "Lightning"
    }
  ]
}
```

7.3. Array as JSON

When using the `JSON.parse()` on a JSON derived from an array, the method will return a JavaScript array, instead of a JavaScript object.

Example: The JSON returned from the server is an array:

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        var myArr = JSON.parse(this.responseText);
        document.getElementById("demo").innerHTML = myArr[0];
    }
};
xmlhttp.open("GET", "json_demo_array.txt", true);
xmlhttp.send();
```

Take a look at `json_demo_array.txt`:

```
[
  "Ford",
  "BMW",
  "Audi",
  "Fiat"
]
```

7.4. Exceptions

a) Parsing Dates:

Date objects are not allowed in JSON.

If you need to include a date, write it as a string.

You can convert it back into a date object later:

Example: Convert a string into a date:

```
var text = '{ "name":"John", "birth":"1986-12-14", "city":"New York"}';
```

```
var obj = JSON.parse(text);
obj.birth = new Date(obj.birth);

document.getElementById("demo").innerHTML = obj.name + ", " +
obj.birth;
```

Or, you can use the second parameter, of the JSON.parse() function, called *reviver*.

The *reviver* parameter is a function that checks each property, before returning the value.

Example: Convert a string into a date, using the *reviver* function:

```
var text = '{ "name":"John", "birth":"1986-12-14", "city":"New York"}';
var 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;
```

b) Parsing Functions:

Functions are not allowed in JSON.

If you need to include a function, write it as a string.

You can convert it back into a function later:

Example: Convert a string into a function:

```
var text = '{ "name":"John", "age":"function () {return 30;}", "city":"New York"}';
var obj = JSON.parse(text);
obj.age = eval("(" + obj.age + ")");

document.getElementById("demo").innerHTML = obj.name + ", " +
obj.age();
```

You should avoid using functions in JSON, the functions will lose their scope, and you would have to use **eval()** to convert them back into functions.

7.5. Browser Support

The `JSON.parse()` function is included in all major browsers and in the latest ECMAScript (JavaScript) standard:

Web Browsers Support

- Firefox 3.5
- Internet Explorer 8
- Chrome
- Opera 10
- Safari 4

For older browsers, a JavaScript library is available at <https://github.com/douglascrockford/JSON-js> .