08. JSON Stringify.

8.1. JSON.stringify()

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

When sending data to a web server, the data has to be a string.

Convert a JavaScript object into a string with JSON.**stringify**().

8.2. Stringify a JavaScript Object

Imagine we have this object in JavaScript:

Example:

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

Use the JavaScript function JSON.stringify() to convert it into a string.

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

The result will be a string following the JSON notation.

myJSON is now a string, and ready to be sent to a server:

Example:

```
var obj = { "name":"John", "age":30, "city":"New York"};
var myJSON = JSON.stringify(obj);
document.getElementById("demo").innerHTML = myJSON;
```

You will learn how to send JSON to the server in the next chapter.

8.3. Stringify a JavaScript Array

It is also possible to stringify JavaScript arrays:

Imagine we have this array in JavaScript:

Example:

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

Use the JavaScript function JSON.**stringify**() to convert it into a string.

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

The result will be a string following the JSON notation.

myJSON is now a string, and ready to be sent to a server:

Example:

```
var arr = [ "John", "Peter", "Sally", "Jane" ];
var myJSON = JSON.stringify(arr);
document.getElementById("demo").innerHTML = myJSON;
```

You will learn how to send JSON to the server in the next chapter.

8.4. Exceptions

a) Stringify Dates:

In JSON, date objects are not allowed. The JSON.**stringify**() function will convert any dates into strings.

Example:

```
var obj = { "name":"John", "today":new Date(), "city":"New York"};
var myJSON = JSON.stringify(obj);

document.getElementById("demo").innerHTML = myJSON;
```

You can convert the string back into a date object at the receiver.

b) Stringify Functions:

In JSON, functions are not allowed as object values.

The JSON.stringify() function will remove any functions from a JavaScript object, both the key and the value:

Example:

```
var obj = { "name":"John", "age":function () {return 30;},
"city":"New York"};
var myJSON = JSON.stringify(obj);

document.getElementById("demo").innerHTML = myJSON;
```

This can be omitted if you convert your functions into strings before running the JSON.stringify() function.

Example:

```
var obj = { "name":"John", "age":function () {return 30;},
"city":"New York"};
obj.age = obj.age.toString();
var myJSON = JSON.stringify(obj);

document.getElementById("demo").innerHTML = myJSON;
```

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.

8.5. Browser Support

The JSON.stringify() 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