

Serializing and Parsing JSON

In this lesson, we'll get into the details of serializing and parsing JSONs.
Let's begin!

WE'LL COVER THE FOLLOWING



- The `stringify()` method
- The `parse()` method
- Listing 8-2: Exercise-08-02/index.html



Serializing and Parsing JSON



When transferring data from one component to another in a distributed system, JSON is a perfect format and fits JavaScript as well.

The fifth edition of the ECMAScript specification formalized the serialization (converting an object to a string) and parsing of JSON into a native global object, not surprisingly named JSON.

This object is very simple for it has only two methods, `stringify()` and `parse()`.


The `stringify()` method

The `stringify()` method converts the specified value (**first argument**) into its JSON representation. The method accepts two other optional arguments, a replacer object (**second argument**), and a spacing value (**third argument**).

If you append the following code line to **Listing 8-1**, it will display the JSON representation indented similarly, as it is written in the code snippet below:

```
var orderJson = JSON.stringify(order, null, "  ");
console.log(orderJson);
```



 **NOTE:** For more details, see the `stringify()` method reference on [MDN](#). You can also customize how a concrete object is serialized by defining a `toJSON()` method for the object instance, which returns the string representation of the object to be used for JSON serialization.

The `parse()` method

The `parse()` method converts a JSON string into an object. The first argument of this method is the JSON string to be converted into an object, and it has a second, optional argument, which prescribes how the value originally produced by parsing is transformed, before being returned. Listing 8-2 shows an example of using `parse()`.

It uses the same order object as Listing 8-1, however, it emulates that the order has been received through a network connection as a JSON string.

Listing 8-2: Exercise-08-02/index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Parsing JSON</title>
  <script>
    var remoteOrder = {
      "date": "11/20/2013",
      "customerId": 116,
      "items": [
        {
          "product": "Surface 4 Pro",
          "unitprice": "799",
```

```

        "amount": 1
    },
    {
        "product": "Type Cover 4",
        "unitprice": "129",
        "amount": 1
    },
    {
        "product": "Docking station",
        "unitprice": "199",
        "amount": 1
    }
]
}
var orderString = JSON.stringify(remoteOrder);

// Assume the order is pushed through the network
var order = JSON.parse(orderString);
console.log("Date: " + order.date);
for (var i = 0; i < order.items.length; i++) {
    console.log("Product: "
        + order.items[i].product);
    console.log("Unit price: "
        + order.items[i].unitprice);
    console.log("Amount: "
        + order.items[i].amount);
}
</script>
</head>
<body>
    Listing 8-2: View the console output
</body>
</html>

```

 **NOTE:** For more details, see the `parse()` method reference on [MDN](#).

In the *next lesson*, we'll make things interesting with JavaScript functions.

Stay tuned! :)