

## Chapter 15.5: JSON

CS 80: Internet Programming

Instructor: Mark Edmonds

### JSON describes data

- XML let us specify a data format
- There's another extremely relevant data format: JSON
- JSON stands for **J**ava**S**cript **O**bject **N**otation

### JSON describes data

- In Javascript, basically everything is an object, so we are literally defining an object with JSON
- A simple example:
  - Suppose you wanted a list of employees, each with a first and last name. What would the XML look like?

### JSON describes data

- Possibly something like this:

```
1 <!-- XML to represent a list of employees-->
2 <employees>
3   <employee>
4     <firstName>John</firstName>
5     <lastName>Doe</lastName>
6   </employee>
7   <employee>
8     <firstName>Anna</firstName>
9     <lastName>Smith</lastName>
10  </employee>
11  <employee>
12    <firstName>Peter</firstName>
13    <lastName>Jones</lastName>
14  </employee>
15 </employees>
```

### JSON describes data

- JSON follows a `{"attribute": "value", ... }` model for representing data
  - `[]` can be the value, indicating an array
  - Let's see an example of what we did above

### JSON describes data

```
1 // json to represent a list of employees
2 {"employees":[
3   {"firstName":"John", "lastName":"Doe"},
4   {"firstName":"Anna", "lastName":"Smith"},
5   {"firstName":"Peter", "lastName":"Jones"}
6 ]}
```

### Why is JSON good?

- The web runs on Javascript
- There's a magical parsing function that converts JSON into an actual Javascript object:
  - `JSON.parse(json_text)` where `json_text` is valid JSON text (e.g. the text above)

### Why is JSON good?

- Very similar to XML: human readable, hierarchical, widely usable, can be used with Ajax (soon to be covered)
- Dissimilar from XML: no end tag, shorter, quicker to read and write, can use arrays
- **The biggest difference: XML has to be parsed by a specialized XML parser. JSON is parsed with one simple Javascript function call**
- Here's an example of loading JSON into an object:

### Example: `simple_json.html`

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <title>Simple JSON example</title>
```

```
7   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/
    jquery.min.js"></script>
8   <script>
9     // run when the document is ready
10    $(document).ready(function(){
11      // create JSON data
12      var text = '{"name":"John Johnson","street":"Oslo West 16","phone"
                  "":"555 1234567"}';
13
14      // parse it into an object
15      var obj = JSON.parse(text);
16
17      // access members/attributes of the new object
18      document.getElementById("demo").innerHTML =
19        obj.name + "<br>" +
20        obj.street + "<br>" +
21        obj.phone;
22    });
23
24  </script>
25  </head>
26
27  <body>
28
29    <h2>JSON Object Creation in JavaScript</h2>
30
31    <p id="demo"></p>
32
33  </body>
34
35  </html>
```

## JSON Values

- number (int or float)
- string (double-quoted)
- boolean
- array (indicated with square brackets)
- object (another pair of curly braces)
- null (no content)

## JSON Values

- How would you load and print `employees` JSON object above?

### Example: `employee_json.html`

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <title>Employee JSON example</title>
7   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/
8     jquery.min.js"></script>
9   <script>
10     // run when the document is ready
11     $(document).ready(function() {
12       // create JSON text
13       var data = '{"employees": [ {"firstName": "John", "lastName": "Doe"
14         }, {"firstName": "Anna", "lastName": "Smith"}, {"firstName":
15         "Peter", "lastName": "Jones" } ]}';
16
17       // parse JSON text into object
18       var employees_obj = JSON.parse(data);
19
20       // setup document
21       document.writeln("<h2>Employees list</h2>");
22       document.writeln("<p>");
23       // loop over every employee
24       for (var i = 0; i < employees_obj.employees.length; i++) {
25         // print employee info
26         document.writeln(employees_obj.employees[i].firstName + " " +
27           employees_obj.employees[i].lastName + "<br>");
28       }
29       document.writeln("</p>");
30     });
31   </script>
32 </head>
33
34 <body>
35 </body>
36
```

```
33 </html>
```

## JSON & JavaScript

- Because of JSON's deep connection with Javascript objects, we can actually declare Javascript objects using JSON syntax without needing to parse
  - Javascript natively supports the ability to create objects using JSON syntax
  - Example for our employees:

### Example: employee\_js.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <title>Employee JSON example</title>
7   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/
8     jquery.min.js"></script>
9   <script>
10     // run when the document is ready
11     $(document).ready(function() {
12       // create JS object using JSON syntax
13       var employees = [{ "firstName": "John", "lastName": "Doe" }, {
14         "firstName": "Anna", "lastName": "Smith"}, {"firstName": "Peter
15         ", "lastName": "Jones" }]];
16
17     // setup document
18     document.writeln("<h2>Employees list</h2>");
19     document.writeln("<p>");
20     // loop over every employee
21     for (var i = 0; i < employees.length; i++) {
22       // print employee info
23       document.writeln(employees[i].firstName + " " + employees[i].
24         lastName + "<br>");
25     }
26     document.writeln("</p>");
27   });
28 </script>
29 </head>
30
```

```
27 <body>
28 </body>
29
30 </html>
```