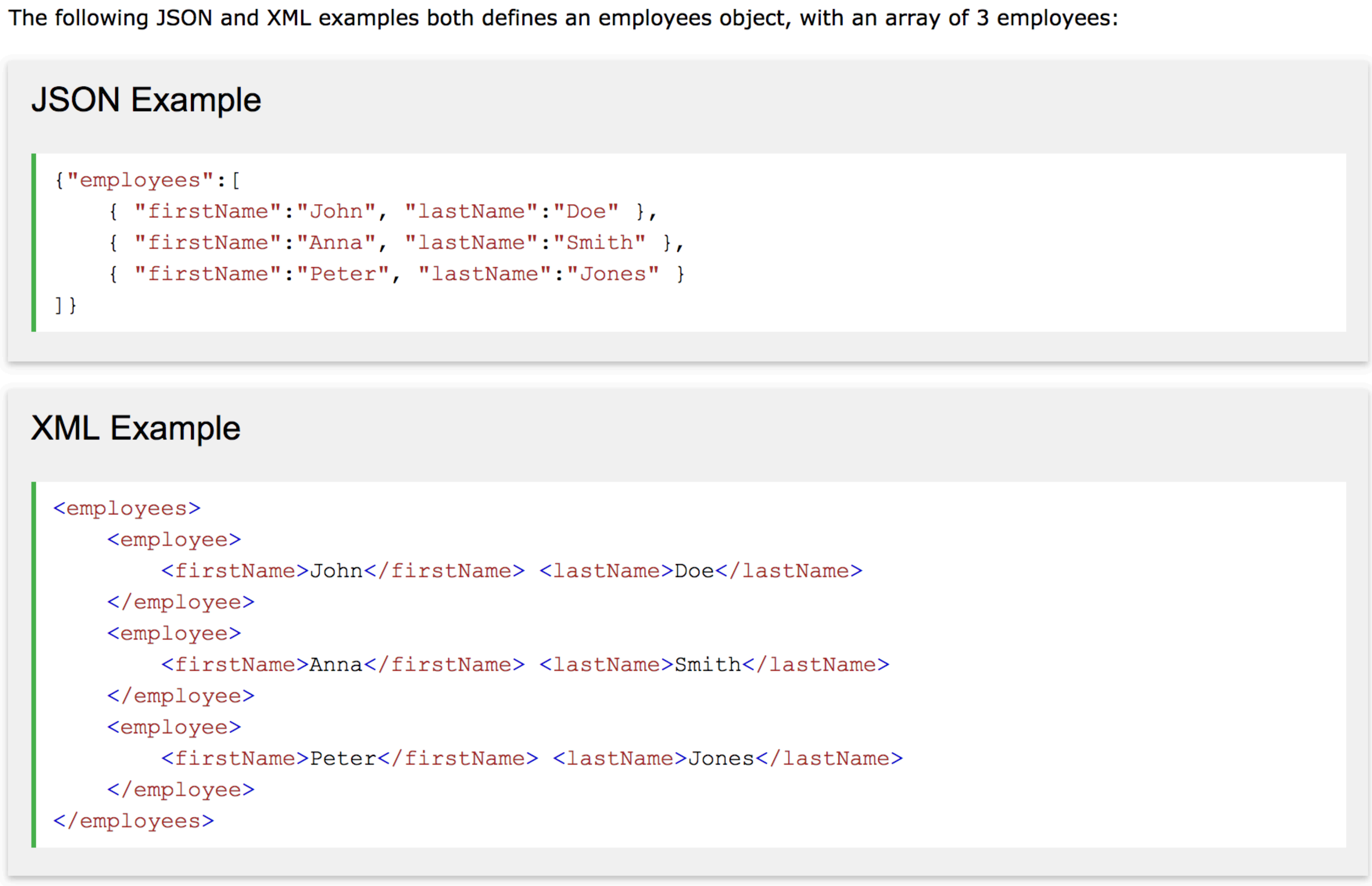
CSC 337

JavaScript Object Notation (JSON)

Rick Mercer

Why JSON over XML? JSON was built to know JS



The following JSON and XML examples both defines an employees object, with an array of 3 employees:

JSON Example

```
{"employees":[
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML Example

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

JSON

- **J**ava**S**cript **O**bject **N**otation
- Data-interchange format
- Lightweight
- Replacement for XML
- It's just a string with some special formatting
- Based on JavaScript
    - but JSON has nothing to do with JavaScript

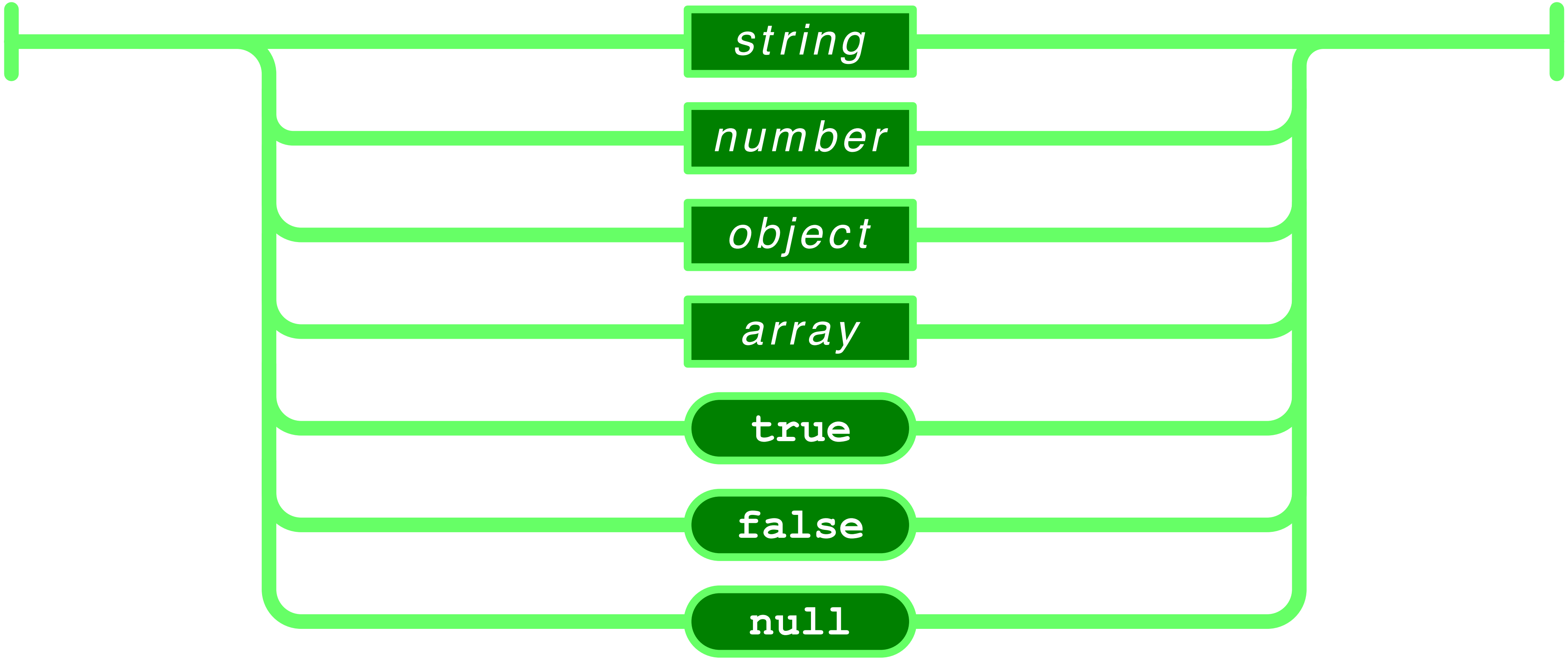
Characteristics

- Language Independent
  - Again, just a big string with special chars like `{ } , ' " : [ ]`
- Text-based
- Light-weight
- Easy to encode inside a `<?php` block
  - PHP has a method for this: `json_encode($array);`
- Easy to parse inside a <script> of .js file
  - JS method: `var array = JSON.parse(xhttp.responseText);`

JSON is not

- JSON is not a style like CSS
- JSON is not a markup language like HTML
- JSON is not a programming language like JavaScript or PHP
- JSON is not a general serialization format as seen in Java

JSON Values



Literal Values in JSON

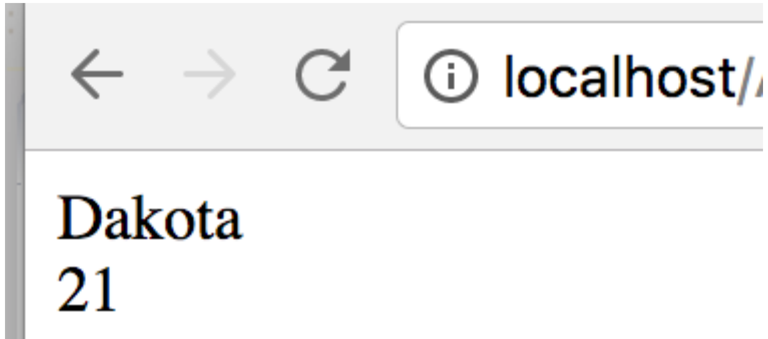
- number: `1   -2   4.56`
- string: `"need double quotes"`
- object: `{ "name":"Dakota",  "age": 20 }`
- null: `null`
- boolean: `true   false`
- arrays: `[1, 2, 3]`
- arrays of objects

```
{ "students" :
     [
       {"name":"Li", "age": 21},
       {"name":"Ty", "age": 19}
     ]
}
```

Important usage! Send a JSON string from server to client. Needed: JSON.parse

```
<?php
  echo '{ "name":"Dakota",  "age":21 }';
?>
```

```
<script>

     . . .
    ajax.onreadystatechange = function() {
       if (ajax.readyState == 4 && ajax.status == 200) {
           var student = JSON.parse(ajax.responseText);
           document.write(student.name + '<br>');
           document.write(student.age + '<br>');
     }
   . . .
</script>
```



```
Dakota
21
```

JSON Arrays

- Square brackets hold **arrays** (JSON has no indexed arrays)

  [1, 2, 3]

- JSON array are very much like JavaScript arrays
  - JSON arrays can't store as many different types as JS array

```
<?php
$array = [123, 'a string', true && false];
echo json_encode($array);
?>

<input type="button" value="Click me" onclick="aFunction()"> <br>
<div id="tochange"> Should see an array here on click </div>
<script>
function aFunction() {
var ajax = new XMLHttpRequest();
  ajax.open("GET", "array.php", true);
  ajax.send();
  ajax.onreadystatechange = function () {
    if (ajax.readyState == 4 && ajax.status == 200) {
      var array = JSON.parse(ajax.responseText);
      for(i = 0; i < array.length; i++)
          document.write(array[i] + "<br>");
     }
  }; // End anonymous function
}
```

Code demo

- Write a service that send back an array of string or and array of integers depending
  - Need isset and $_GET[]

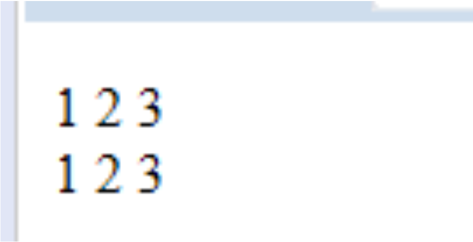
JSON Objects

- A JSON object is an unordered set of name/value pairs
- A colon exist between the name and the value
- Data is separated by commas
- Curly braces hold **objects**

```
{ "name": "Chris Miller",  "age":20,  "year": "Sophomore" }
```

JSON objects look like JavaScript objects

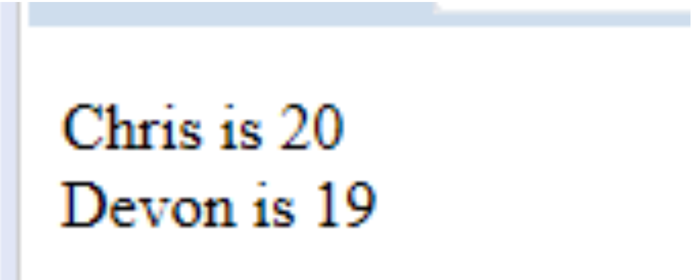
```
<script>
var array = [1,2,3];
for (i = 0; i < 3; i++ )
   document.write(array[i] + " ");

document.write("<br>");

var string = '[1,2,3]';
// Convert a JSON string to JavaScript
var JSarr = JSON.parse(string);
for (i = 0; i < 3; i++ )
   document.write(JSarr[i] + " ");
</script>
```



```
1 2 3
1 2 3
```

Array of objects; the same code in JS and JSON

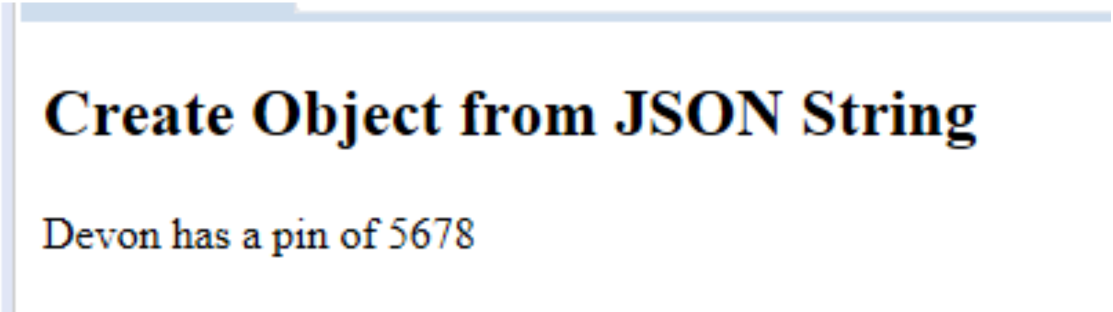
```
<script>
// Strings like this will be echoed from a PHP script on the server
var string = '{ "student": [ ' +
                  '{"name": "Chris",  "age":20},' +
                  '{"name": "Devon", "age":19}' +
                           ']
              }'
var JSref = JSON.parse(string);
for (i = 0; i < 3; i++ ) {
  document.write(JSref.student[i].name + " is " + JSref.student[i].age );
  document.write("<br>");
}
</script>
```



```
Chris is 20
Devon is 19
```

JSON data converted to JS to change the DOM

```
<h2>Create Object from JSON String</h2>
<p id="demo"></p>
<script>
// Store an array of objects:
var JSON_arrayOfObjects = '{"accounts":[' +
                                          '{"ID":"Casey", "pin":"1234" },' +
                                          '{"ID":"Devon", "pin":"5678" },' +
                                          '{"ID":"Lindsay", "pin":"9999"}' +
                                      ']' +
                          '}';
ref = JSON.parse(JSON_arrayOfObjects);
document.getElementById("demo").innerHTML =
    ref.accounts[1].ID + " has a pin of " + ref.accounts[1].pin;
</script>
```



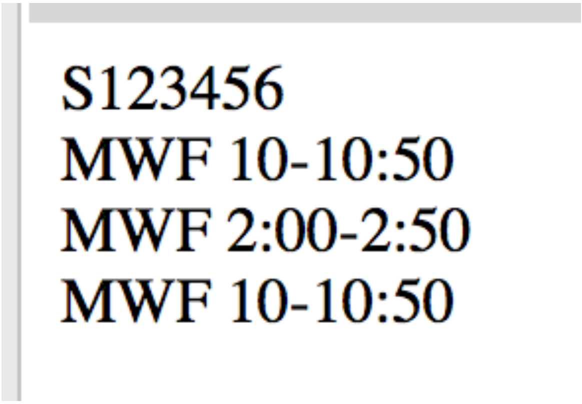
Create Object from JSON String

Devon has a pin of 5678

Javascript objects look like JSON objects

- One way to create a JavaScript Object, use { }

```
// Store an JS object with a JS object inside it:
var JSObject = { "id":'S123456',
            "inner": {
                            "CSC127A": 'MWF 10-10:50',
                     "CSC337": 'MWF 2:00-2:50',
                    "CSC335": 'MWF 10-10:50'
                  }
         };

document.write(JSObject.id + '<br>');
document.write(JSObject.inner.CSC127A + '<br>');
document.write(JSObject.inner.CSC337 + '<br>');
document.write(JSObject.inner.CSC335 + '<br>');
```



```
S123456
MWF 10-10:50
MWF 2:00-2:50
MWF 10-10:50
```

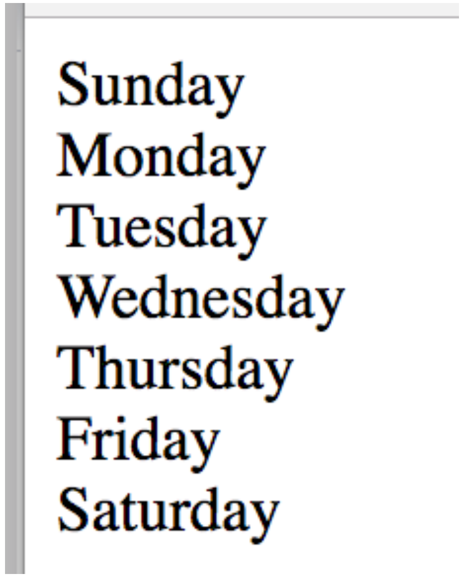
JSON Arrays

- Arrays are ordered sequences of values
- Arrays are wrapped in `[ ]`
- separates values with `,`
- JSON does not have indexed arrays

JSON arrays in PHP as a string sent to AJAX

```
<?php
echo '["Sunday", "Monday", "Tuesday", "Wednesday",
       "Thursday", "Friday", "Saturday"]';
?>
```

```
ajax.onreadystatechange = function() {
  if (ajax.readyState == 4 && ajax.status == 200) {
    var array = JSON.parse(ajax.responseText);
    for(index = 0; index < array.length; index++) {
       document.write(array[index] + '<br>');
    }
  }
}
```



```
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

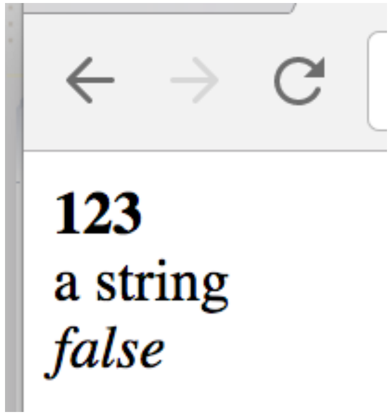
json_encode

- When exchanging data between a browser and a server, the data can only be echoed as text, JSON for example. We will use json_encode

```
<?php
  $array = [123, 'a string', true && false];
  echo json_encode($array);
?>
```

- We convert any JSON received from the server into JS objects

```
 ajax.onreadystatechange = function() {
   if (ajax.readyState == 4 && ajax.status == 200) {
      var array = JSON.parse(ajax.responseText);
      document.write('<b>' + array[0] + '</b>' + '<br>');
      document.write(array[1] + '<br>');
      document.write('<i>' + array[2] + '</i>');
   }
}
```



123
a string
false