JSON: JavaScript Object Notation

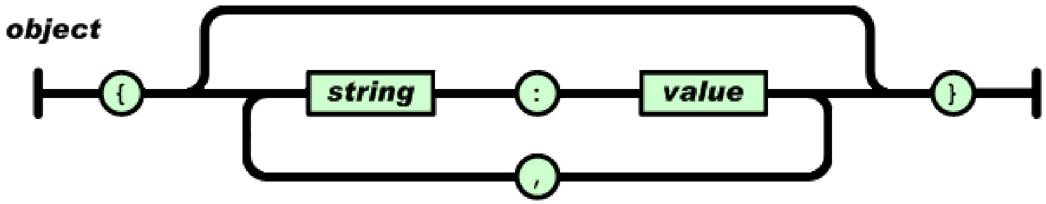
JSON as an XML Alternative

- JSON = JavaScript Object Notation
  - It’s really language independent
  - most programming languages can easily read it and instantiate objects or some other data structure
- JSON is a light-weight alternative to XML for data-interchange
- Started gaining tracking ~2006 and now widely used
- http://json.org/ has more information

JSON Data – A name and a value

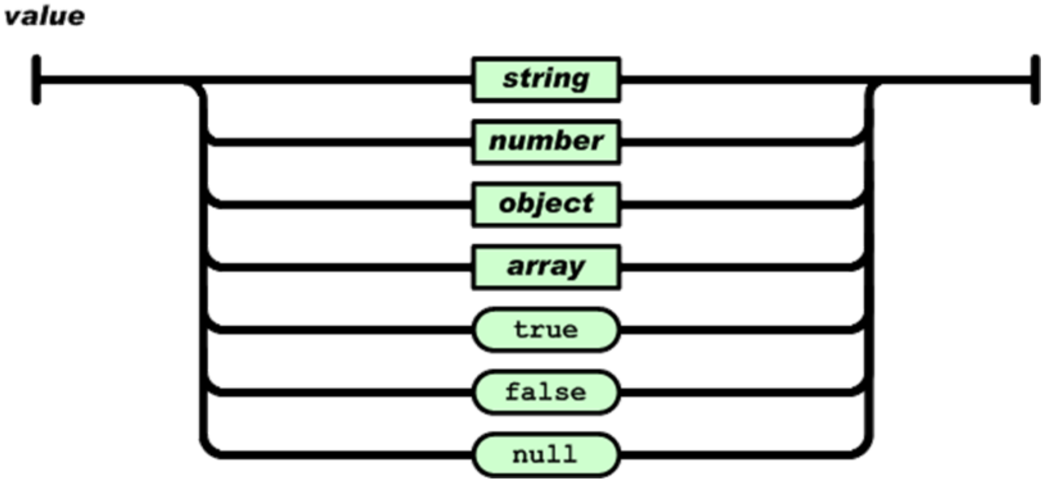
- A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value
- Unordered sets of name/value pairs
- Begins with **{** (left brace)
- Ends with **}** (right brace)
- Each name is followed by **:** (colon)
- Name/value pairs are separated by **,** (comma)

```
{
"employee_id": 1234567,
"name": "Jeff Fox",
"hire_date": "1/1/2013",
"location": "Norwalk, CT",
"consultant": false
}
```



JSON Data – A name and a value

- In JSON, *values* must be one of the following data types:
- a string
- a number
- an object (JSON object)
- an array
- a boolean
- null



```
{
"employee_id": 1234567,
"name": "Jeff Fox",
"hire_date": "1/1/2013",
"location": "Norwalk, CT",
"consultant": false
}
```

JSON Data – A name and a value

- Strings in JSON must be written in double quotes.

```
{ "name":"John" }
```

- Numbers in JSON must be an integer or a floating point.

```
{ "age":30 }
```

- Values in JSON can be objects.

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

- Values in JSON can be arrays.

```
{
"employees":[ "John", "Anna", "Peter" ]
}
```

Another example: XML vs JSON

```
<?xml version="1.0"?>
<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>
```

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

XML vs JSON

- JSON is Like XML Because
  - Both JSON and XML are "self describing" (human readable)
  - Both JSON and XML are hierarchical (values within values)
  - Both JSON and XML can be parsed and used by lots of programming languages
- JSON is Unlike XML Because
  - JSON doesn't use end tag
  - JSON is shorter
  - JSON is quicker to read and write
  - JSON can use arrays
  - JSON has a better fit for OO systems than XML
- The biggest difference is:
  - XML has to be parsed with an XML parser. JSON can be parsed by a standard JavaScript function.

Why JSON?

Steps involved in exchanging data from web server to browser involves:

Using XML

1. Fetch an XML document from web server.
2. Use the XML DOM to loop through the document.
3. Extract values and store in variables.
4. It also involves type conversions.

Using JSON

1. Fetch a JSON string.
2. Parse the JSON using JavaScript functions.

Class Activity

- Convert the following bookstore.xml to bookstore.json

```
<?xml version="1.0"?>
<bookstore>
    <book category="sci-fi">
            <title lang="en"> 2001</title>
            <author>Arthur C. Clarke</author>
            <price>$30.0</price>
            <year>1968</year>
    </book>
    <book>
             <title lang="rs">Story about a True Man</title>
             <author>Boris Polevoy</author>
            <price>$20.00</price>
            <year>1952</year>
     </book>
</bookstore>
```