

LECTURE-3

- JS Events Review (from Lecture-2)
- XML
- JavaScript Object Notation – JSON
- Cookies

XML – EXTENDED MARKUP LANGUAGE

- XML is a markup language, like HTML
- Designed to carry data
 - Not to display data
- XML tags are **NOT** predefined.
 - Unlike HTML
 - You must define your own tags
- Self-descriptive
- Represented in plain text.

A SIMPLE EXAMPLE

<note>

<to>Tove</to>

<from>Jani</from>

<heading>Reminder</heading>

<body>Let's meet tomorrow!</body>

</note>

Note:

1. User defined tags.
2. Self descriptive

JSON

- Text based
 - Very useful in transferring text data over the web
 - Language independent
 - Used in JS, Java, PHP, etc.
- Provides easy means to
 - Define JS objects
 - Can convert JS objects to strings and vice-versa
 - Different languages have functions for conversion.

JSON EXAMPLE

- `var person =`

```
{  
  "firstname": "John",  
  "lastname": "Doe",  
  "age": 50,  
  "address": {  
    "street": "11 Broadway",  
    "city": "New York City"  
  }  
};
```
- Can access data of individual fields
 - `person.firstName` (or) `person[firstName]`
 - `person.lastName` (or) `person[lastName]`
 - `person.address.street`(or) `person.address[street]`

JSON DATA TYPES

- A JSON object member can be of type
 - Number
 - String
 - Another JSON object
 - Nested JSON objects
 - Boolean
 - Array
 - Null
- Values of objects' members can be
 - Modified.
 - E.g., `person.address[street] = "2 Coumbia Way"`
 - Deleted
 - E.g., `delete person.age;`

JSON DATA CONVERSION TO STRING

- JSON object to string conversion
 - `var personString = JSON.stringify(person)`
- JSON string to an JSON object
 - `var person = JSON.parse(personString);`
- Useful in sending JS objects over HTTP as strings.

JSON VS. XML

- Similarities

- Self describing and text based.
- Have user defined “tags” (unlike HTML)
- Nested
- Can be parsed in many languages
- Can be fetched using XMLHttpRequest (AJAX).

- Differences

- JSON can be parsed by JS, XML can be parsed by XML parser
- JSON does not have an end tag (e.g., **NO** </firstName>)
- JSON can use arrays
- JSON is less verbose

COOKIES

- Small amount of information a web server stores on a browser.
- Cookie structure – `<name, value>` pairs
- Typically used to
 - Remember login and password
 - User preferences
 - Web sites visited
 - Personalization
- Location where cookies are stored –
 - Different for each browser.
- Cookies have an expiration time
- Cookies can be removed

COOKIES ... CONTD.

- Cookies $\langle \text{name}, \text{value} \rangle$ pairs store
 - Name of the cookie
 - Server name and path
 - If the path is “/”, cookie is valid in the entire domain
 - Expiry Thursday, September 21, 2017
- Each web server
 - Can read its **OWN** cookies when the web page is loaded.
 - **NOT** cookies of some other web server
 - Can load multiple (up to a finite limit) cookies on each browser.

COOKIES ... CONTD.

- Cookies
 - are plain text files.
 - can't be used to read other data on the computer.
 - are not executable files
 - Cannot erase data on computer
- A site can open **ONLY** cookies it owns
- Cookies are set using “Set-Cookie” attribute in HTTP.