JavaScript strings clarification

- Non-standard syntax used in an example
  - Using `s[i]` to access the $i^{th}$ character of string `s`
  - Works in Firefox, does not work in IE
- Standard functions for a working with strings
  - Use `s.charAt(i)` to access the $i^{th}$ character
  - Use `s.length` to find its length
  - Use `s.substring(i,j)` to get substring from $i^{th}$ to $j^{th}$ character
  - Use `s.indexOf('a')` to find index of first occurrence of character `a`

Web services

Lecture 21

Lecture outline

- Web Services
  - Remote procedure calls
  - AJAX (asynchronous requests)
What are web services?
- A form of remote procedure call: your program (the client) asks another computer (the server) to run a procedure for you
  - Parameters sent over the network from client to server
  - Results sent over network from server to client
- Why would you ever want to do a remote procedure call?
  - Data needed for answer not (easily) accessible to your computer
  - You want to re-use existing procedures that run in a different environment than your program
  - Your computer lacks the resources (i.e. processor capacity, memory, network connection speed) to compute the result
- There are many other forms of RPC older than web services
  - CORBA, DCOM, SunRPC, RMI

Internals of an RPC framework
- Code for marshalling/unmarshalling – encoding and decoding parameters/results
  - A.k.a. serializing objects
- Description of the available procedures (methods)
  - Using an interface description language (IDL)
- Framework that turns these descriptions into “stubs”
  - On the client the stub makes it look to your program like the stub is executing the procedure locally
  - On the server the stub invokes the procedure
  - The client and server stub interact over the network

Specific to web services
- They run over http
  - Procedure call is in an http request
  - Result is in an http response
- They use XML to
  - Encode responses
  - Encode requests (sometimes)
  - Describe the procedures (incl. arguments and results)
- Client and server often use different languages
  - Client may be JavaScript code in browser – AJAX
- Client and server are often in different organizations
More on web services

- Many companies allow access to their data / services through web services (some free)
- Examples: Amazon, ebay, Google, Yahoo, USPS
- Two popular flavors
  - SOAP (Simple Object Access Protocol) – requests in XML
  - REST (REpresentational State Transfer) – requests are like query strings of HTML forms (name-value pairs)
- SOAP more powerful, REST has lower overhead
- WSDL (web service description language) used for XML file describing method names, parameters and results for a given web service

ASP.NET and web services

- Writing web services
  - Create new web site of type “web service”
  - Write C# methods exposed to clients
  - The system does the rest (incl. generating WSDL)
- Using web services
  - Add “web reference” to your site/project
    - Give URL of WSDL file
  - Use web service in your code
    - System adds “glue” – class definitions, stubs for building requests and parsing responses, etc.
Lecture outline

- Web Services
  - Remote procedure calls
- AJAX (asynchronous requests)

What is AJAX?

- Dynamic, interactive web pages that feel more like a desktop application than a static page
- Key technologies – Asynchronous requests, JavaScript, DOM manipulation, CSS (especially positioning of elements), web services using XML
- Core tenet – in response to user actions, do not refresh full page from server, send small requests as needed and update individual elements of web page
- Examples: dragging/zooming maps at Google and Yahoo, “tooltip details” at Netflix, etc.

Asynchronous requests

- Requests from browser other than those triggered by the user following a link or submitting a form
  - Not postbacks
- Explicitly triggered by JavaScript code
  - Building a request object
  - Registering a handler with the browser
  - Browser calls handler when various events occur (e.g. the reply arrives)
- May not be a direct result of an user action
  - E.g. pre-fetching neighboring tiles of a map
AJAX vs. postbacks

- Replies to asynchronous requests may be smaller
  - Typically contain less information than full page
- When coupled with animations, asynchronous requests can achieve effects that are not possible with postbacks
- Use postbacks triggered by a submit button when you want to be sure that the user understands that she is submitting information to the server

The XMLHttpRequest object

```javascript
var xmlHttp;
function createXMLHttpRequest() {
    if (window.ActiveXObject) {
        return new ActiveXObject("Microsoft.XMLHTTP");
    } else if (window.XMLHttpRequest) {
        return new XMLHttpRequest();
    }
}
function do_liveSearch(pattern) {
    xmlHttp = createXMLHttpRequest();
    xmlHttp.onreadystatechange = handleStateChange;
    xmlHttp.open("GET", document.URL.substring(0, document.URL.lastIndexOf("/")) + "+/SearchService.aspx?pattern="+pattern, true);
    xmlHttp.send(null);
}
function handleStateChange() {
    if(xmlHttp.readyState == 4) {
        if(xmlHttp.status == 200) {
            updatePage();
        } else {alert("Status "+xmlHttp.status);}
    }
}
... // var xmlDoc = xmlHttp.responseXML;
```