COMP335
Web Application Development

JQuery

Tentative Schedule

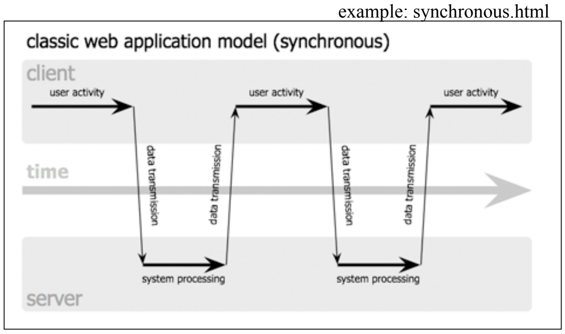
- Group Project
  - group name and short (one paragraph) description
  - **http://goo.gl/forms/eo5wnbMtXf**
- Today
  - AJAX
- Next classes
  - Lab2: Group project prototyping and presentation

- In-class programming exam (tentative)
  - **3/14 (M)**

AJAX

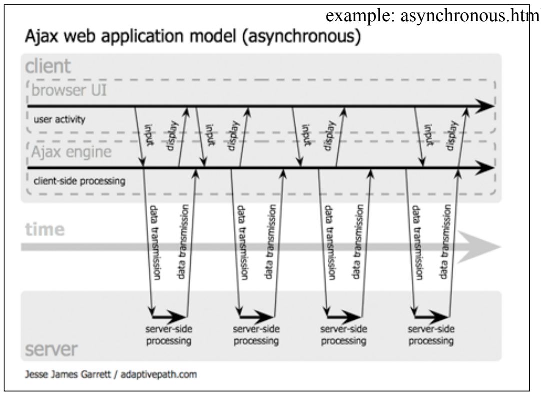
- Asynchronous JavaScript and XML (AJAX) is a term used to describe a paradigm that allows a web browser to send messages back to the server without interrupting the flow of what's being shown in the browser.
- With AJAX, you can
  - update a web page without reloading the page
  - request data from a server – after the page has loaded
  - receive data from a server – after the page has loaded
  - send data to a server – in the background
- Example
  - Google suggest

example: synchronous.html



source: http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/

example: asynchronous.html



source: http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/

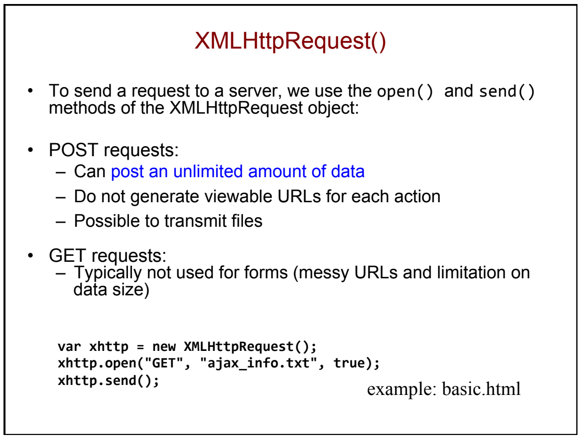
`XMLHttpRequest()`

- `XMLHttpRequest` Object
  - The `XMLHttpRequest` object is used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();

  xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
      document.getElementById("demo").innerHTML = xhttp.responseText;
    }
  };
}
```

example: basic.html

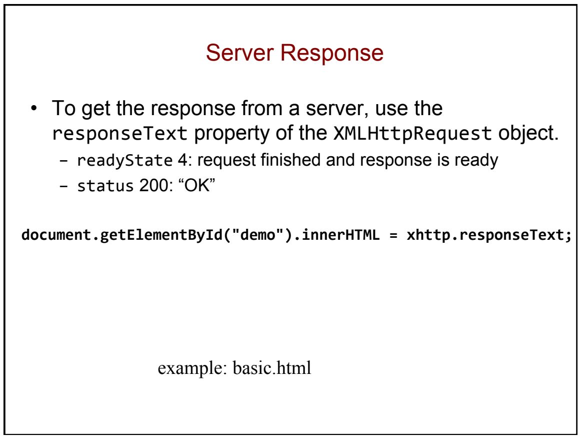


XMLHttpRequest()

- To send a request to a server, we use the `open()` and `send()` methods of the XMLHttpRequest object:
- POST requests:
  - Can post an unlimited amount of data
  - Do not generate viewable URLs for each action
  - Possible to transmit files
- GET requests:
  - Typically not used for forms (messy URLs and limitation on data size)

```
var xhttp = new XMLHttpRequest();
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

example: basic.html

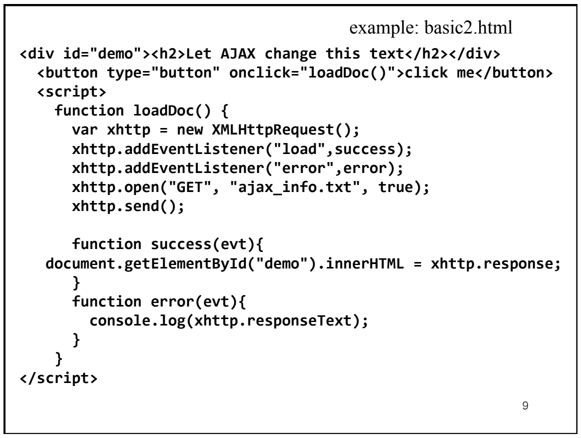


Server Response

- To get the response from a server, use the `responseText` property of the `XMLHttpRequest` object.
  - `readyState` 4: request finished and response is ready
  - `status` 200: “OK”

```
document.getElementById("demo").innerHTML = xhttp.responseText;
```

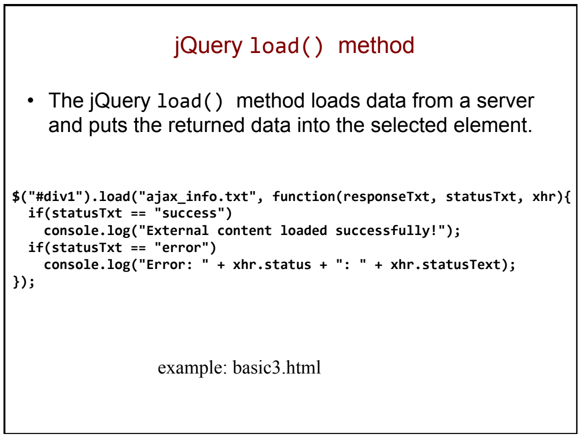
example: basic.html



example: basic2.html

```
<div id="demo"><h2>Let AJAX change this text</h2></div>
  <button type="button" onclick="loadDoc()">click me</button>
  <script>
    function loadDoc() {
      var xhttp = new XMLHttpRequest();
      xhttp.addEventListener("load",success);
      xhttp.addEventListener("error",error);
      xhttp.open("GET", "ajax_info.txt", true);
      xhttp.send();

      function success(evt){
   document.getElementById("demo").innerHTML = xhttp.response;
      }
      function error(evt){
        console.log(xhttp.responseText);
      }
    }
</script>
```

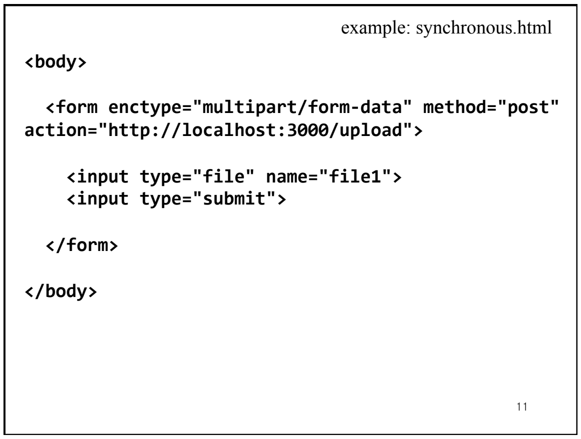


jQuery `load()` method

- The jQuery `load()` method loads data from a server and puts the returned data into the selected element.

```
$("#div1").load("ajax_info.txt", function(responseTxt, statusTxt, xhr){
  if(statusTxt == "success")
    console.log("External content loaded successfully!");
  if(statusTxt == "error")
    console.log("Error: " + xhr.status + ": " + xhr.statusText);
});
```

example: basic3.html



example: synchronous.html

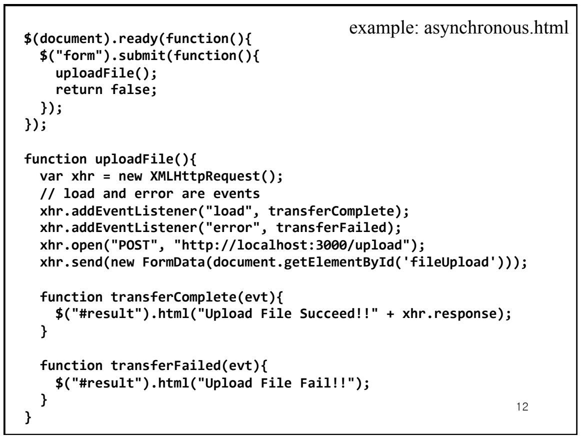
```
<body>

  <form enctype="multipart/form-data" method="post"
action="http://localhost:3000/upload">

    <input type="file" name="file1">
    <input type="submit">

  </form>

</body>
```



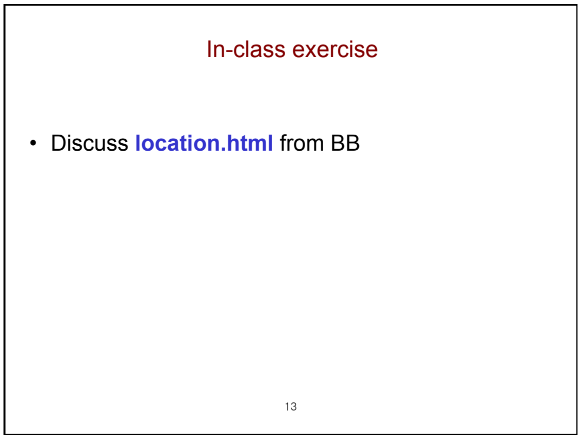
example: asynchronous.html

```
$(document).ready(function(){
  $("form").submit(function(){
    uploadFile();
    return false;
  });
});

function uploadFile(){
  var xhr = new XMLHttpRequest();
  // load and error are events
  xhr.addEventListener("load", transferComplete);
  xhr.addEventListener("error", transferFailed);
  xhr.open("POST", "http://localhost:3000/upload");
  xhr.send(new FormData(document.getElementById('fileUpload')));

  function transferComplete(evt){
    $("#result").html("Upload File Succeed!!" + xhr.response);
  }

  function transferFailed(evt){
    $("#result").html("Upload File Fail!!");
  }
}
```



In-class exercise

- Discuss **location.html** from BB