국립충남대학교 CHUNGNAM NATIONAL UNIVERSITY

13. AJAX, jQuery AJAX

웹프로그래밍

2016년 1학기

충남대학교 컴퓨터공학과

목차

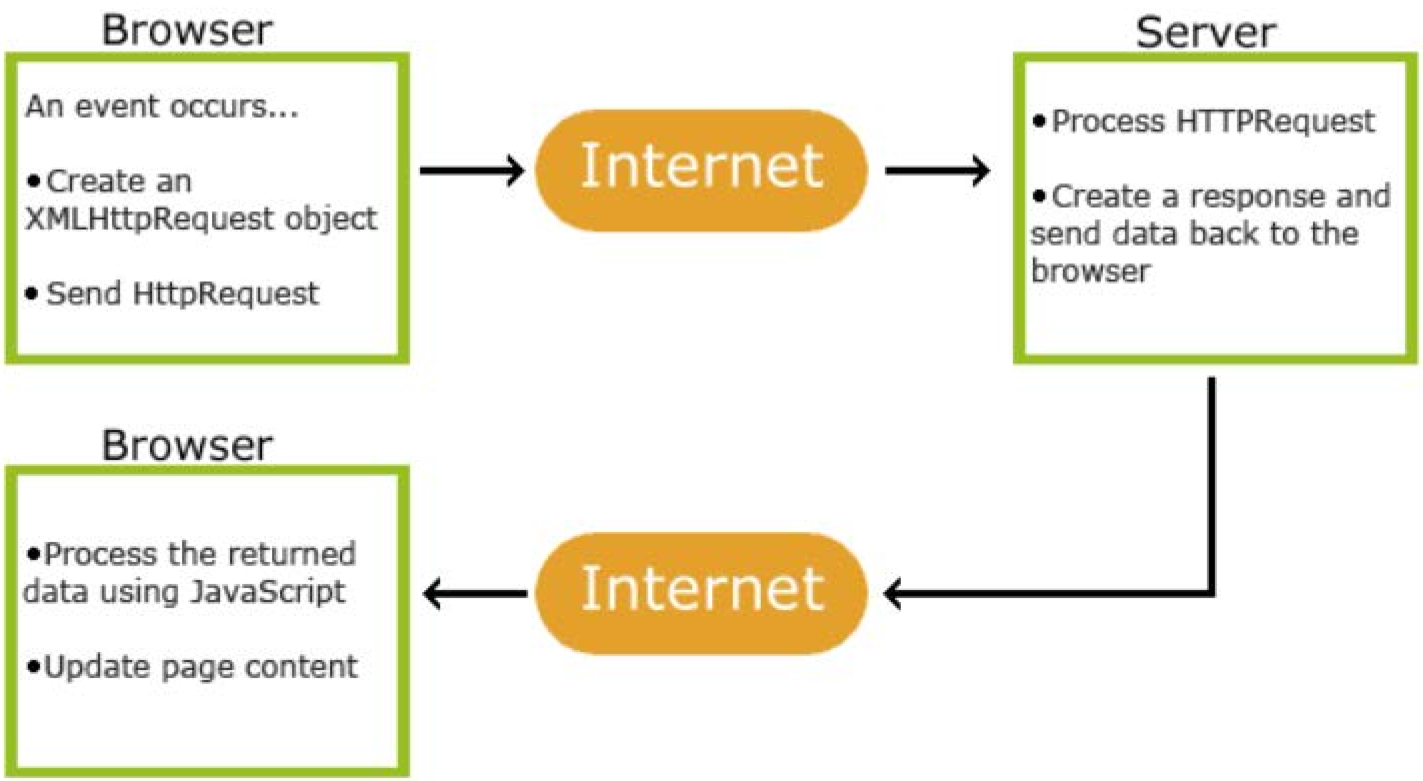
- AJAX Introduction
- AJAX XMLHttp
- AJAX Request
- AJAX Response
- AJAX Event
- AJAX Response
- jQuery AJAX Introduction
- jQuery AJAX Load
- jQuery AJAX Get/Post

AJAX Introduction

- **What is AJAX?**
  - **A**synchronous **J**ava**S**cript and **X**ML
  - A technique for creating fast and dynamic web pages
  - Allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. i.e., it is possible to update parts of a web page, without reloading the whole page.
  - Examples of applications
    - Google Maps, Gmail, YouTube, and Facebook tabs.

AJAX Introduction (cont'd)

How AJAX Works



Browser

An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

Server

- Process HTTPRequest
- Create a response and send data back to the browser

Browser

- Process the returned data using JavaScript
- Update page content

AJAX Introduction (cont'd)

- **AJAX is Based on Internet Standards, and uses a combinations of:**
  - **XMLHttpRequest** Object (to retrieve data from a web server)
  - JavaScript/DOM (to display/use the data)
  - ❖ XMLHttpRequest is a misleading name. You don't have to understand XML to use AJAX.

- **Google Suggest**
  - AJAX was made popular in 2005 by Google, with Google Suggest.
  - When you start typing in Google's search box, JavaScript sends the letters off to a server and the server returns a list of suggestions.

- **AJAX Example:** *Try it!*

AJAX - Create an XMLHttpRequest Object

The XMLHttpRequest Object

- Used to exchange data with a server behind the scenes.
- i.e., it is possible to update parts of a web page, without reloading the whole page.

Create an XMLHttpRequest Object

- Syntax:

```
variable = new XMLHttpRequest();
```

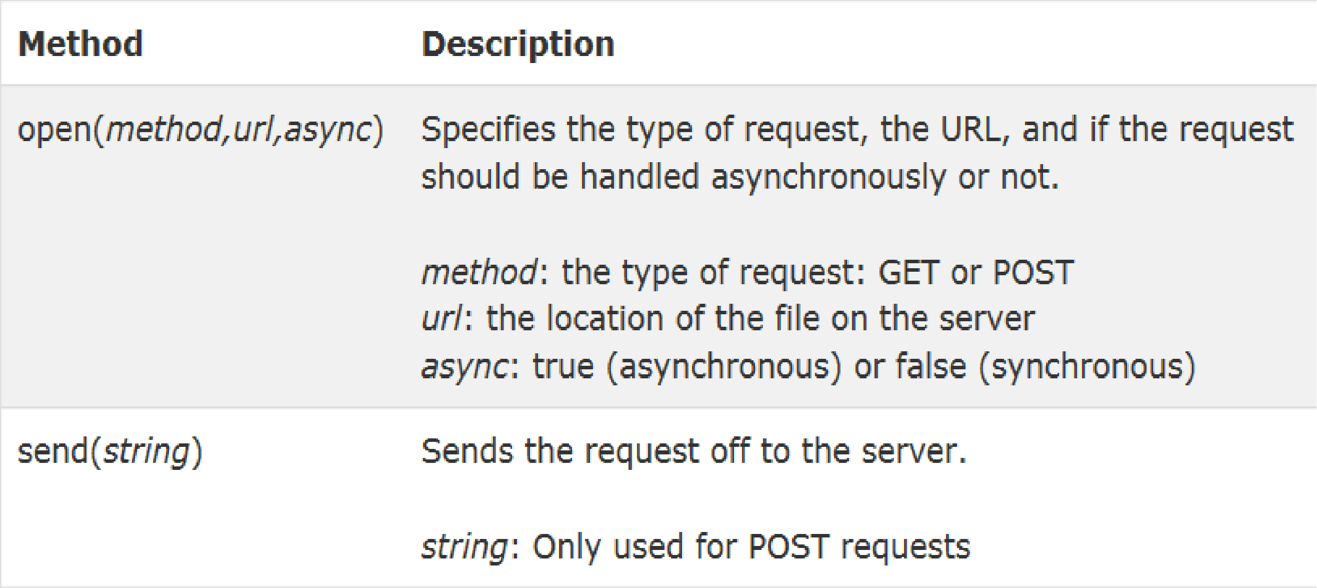
- To handle all browsers, including IE5 and IE6, check if the browser supports the XMLHttpRequest object. If it does, create an XMLHttpRequest object, if not, create an ActiveXObject.
- Example: *Try it!*

AJAX - Send a Request To a Server

Send a Request To a Server

- Use the open() and send () methods of the XMLHttpRequest object:

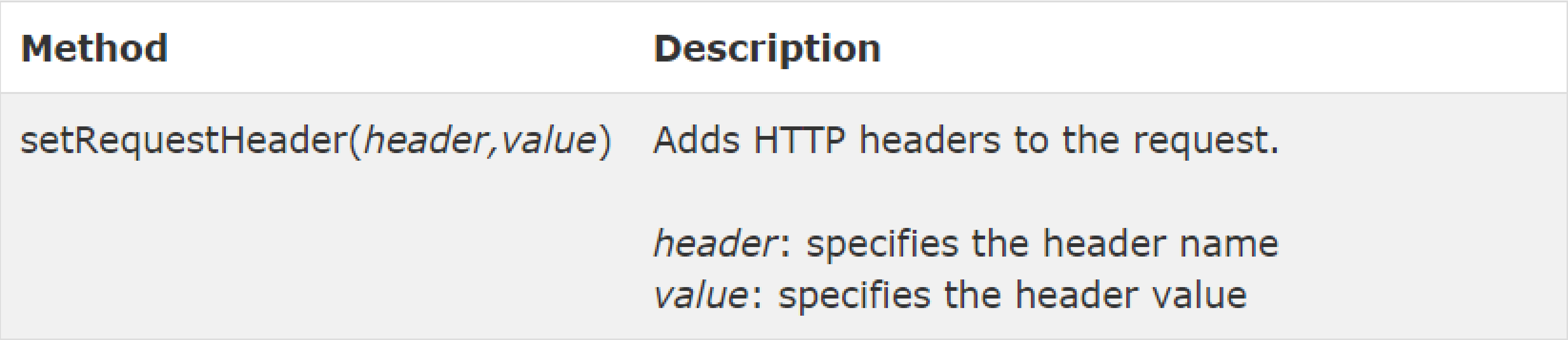
```
xmlhttp.open("GET", "ajax_info.txt", true);
xmlhttp.send( );
```



Method	Description
open(*method*,*url*,*async*)	Specifies the type of request, the URL, and if the request should be handled asynchronously or not. *method*: the type of request: GET or POST *url*: the location of the file on the server *async*: true (asynchronous) or false (synchronous)
send(*string*)	Sends the request off to the server. *string*: Only used for POST requests

AJAX - Send a Request To a Server (cont'd)

- **GET Requests**
  - Simple example: *Try it!*
  - If you want to send information with the GET method: *Try it!*
- **POST Requests**
  - Simple example: *Try it!*
  - To POST data like an HTML form: *Try it!*

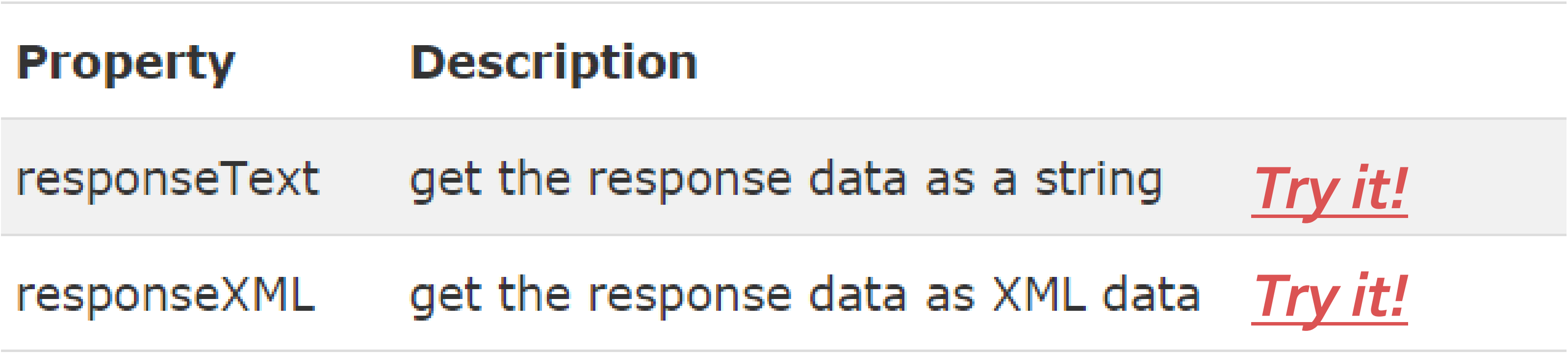


Method	Description
setRequestHeader(*header,value*)	Adds HTTP headers to the request. *header*: specifies the header name *value*: specifies the header value

AJAX - Server Response

Server Response

- To get the response from a server, use the **responseText** or **responseXML** property of the XMLHttpRequest object.

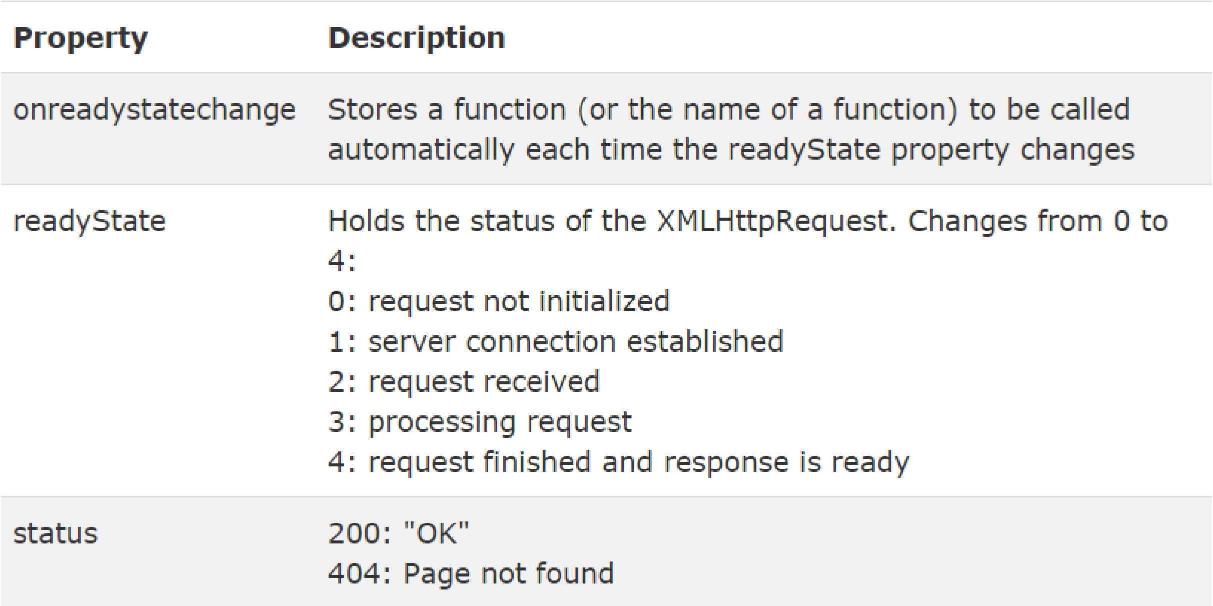


Property	Description	
responseText	get the response data as a string	*Try it!*
responseXML	get the response data as XML data	*Try it!*

AJAX - The onreadystatechange Event

The onreadystatechange event

- When a request to a server is sent, we want to perform some actions based on the response.
- Three important properties of the XMLHttpRequest object:

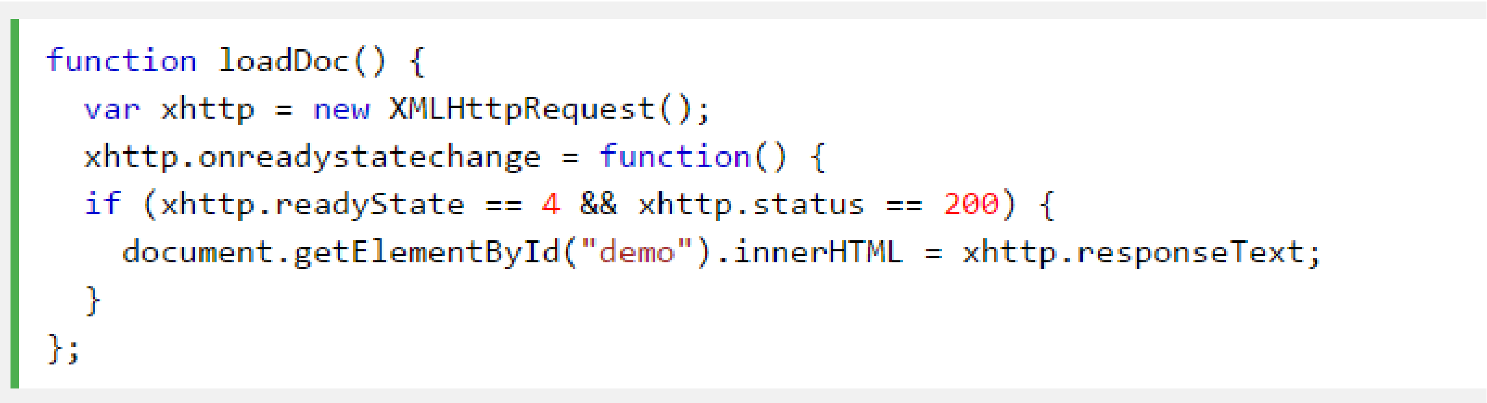


Property	Description
onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 404: Page not found

AJAX - The onreadystatechange Event (con’t)

The onreadystatechange event (con’t)

- In the onreadystatechange event, we specify what will happen when the server response is ready to be processed.
- When readyState is 4 and status is 200, the response is ready.
- Example: *Try it!*

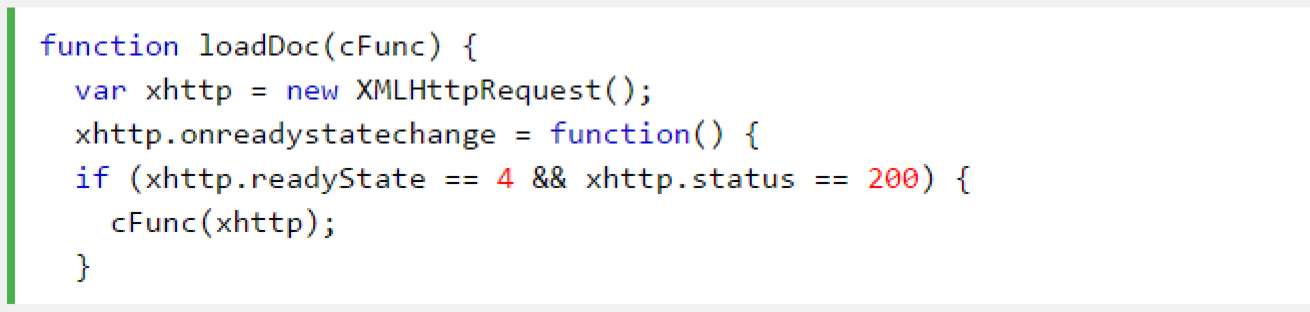


```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
  if (xhttp.readyState == 4 && xhttp.status == 200) {
    document.getElementById("demo").innerHTML = xhttp.responseText;
  }
};
```

AJAX - The onreadystatechange Event (con't)

Using a Callback Function

- If you have more than one AJAX task on your website, you should create ONE standard function for creating the XMLHttpRequest object, and call this for each AJAX task.
- The function call should contain the URL and what to do on onreadystatechange (which is probably different for each call).
- Example: *Try it!*

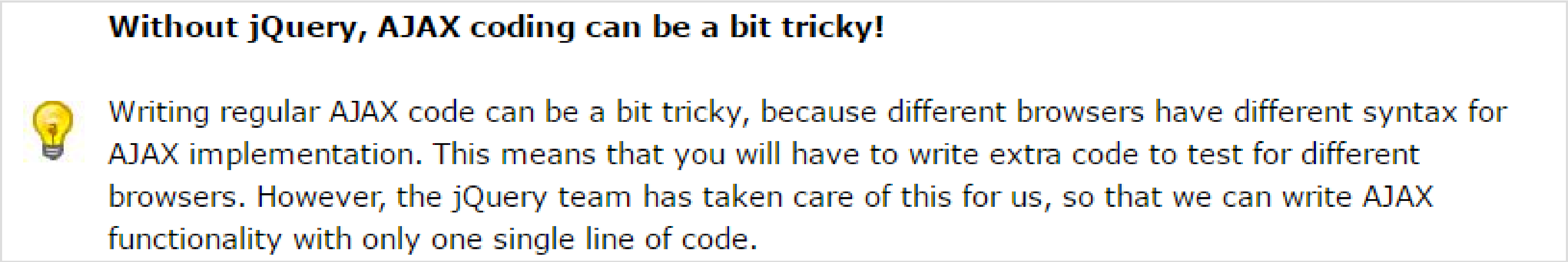


```
function loadDoc(cFunc) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
  if (xhttp.readyState == 4 && xhttp.status == 200) {
    cFunc(xhttp);
  }
```

jQuery - AJAX Introduction *Try it!*

What About jQuery and AJAX?

- jQuery provides several methods for AJAX functionality.
- With the jQuery AJAX methods:
  - You can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post.
  - You can load the external data directly into the selected HTML elements for your web page.



Without jQuery, AJAX coding can be a bit tricky!

Writing regular AJAX code can be a bit tricky, because different browsers have different syntax for AJAX implementation. This means that you will have to write extra code to test for different browsers. However, the jQuery team has taken care of this for us, so that we can write AJAX functionality with only one single line of code.

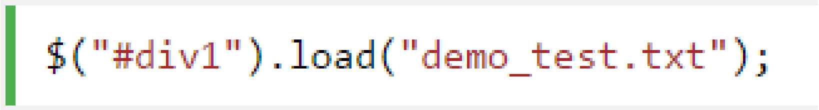
jQuery - AJAX load() Method

- **Loads data from a server and puts the returned data into the selected element.**
- **Syntax:** $(*selector*).load(*URL, data, callback*);
  - The required URL parameter specifies the URL you wish to load.
  - The optional data parameter specifies a set of querystring key/value pairs to send along with the request.
  - The optional callback parameter is the name of a function to be executed after the load() method is completed.

jQuery - AJAX load() Method (cont'd)

- **The content of "demo_test.txt"**

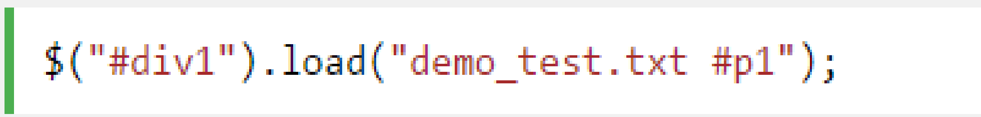
```
<h2> jQuery and AJAX is FUN!!!</h2>
<p id="p1"> This is some text in a paragraph.</p>
```

- **Example 1:** 

```
$("#div1").load("demo_test.txt");
```

Try it!

  - Loads the content of the file "demo_test.txt" into a specific <div> element.

- **Example 2:** 

```
$("#div1").load("demo_test.txt #p1");
```

Try it!

  - Loads the content of the element with id="p1", inside the file "demo_test.txt", into a specific <div> element.

jQuery - AJAX load() Method (cont'd)

- **The optional callback parameter specifies a callback function to run when the load() method is completed.**
- **The callback function can have different parameters:**
  - responseTxt: contains the resulting content if the call succeed.
  - statusTxt: contains the status of the call.
  - xhr: contains the XMLHttpRequest object.
- **Example:** *Try it!*
  - Displays an alert box after the load() method completes.
  - If the load () method has succeed, it displays "External content loaded successfully!", and if it fails, it displays an error message.
- *jQuery AJAX Reference*

jQuery - AJAX get() and post() Methods

- **jQuery get() and post() methods are used to request data from the server with an HTTP GET or POST request.**
- **jQuery $.get() Method**
  - Requests data from the server with an HTTP GET request.
  - Syntax:

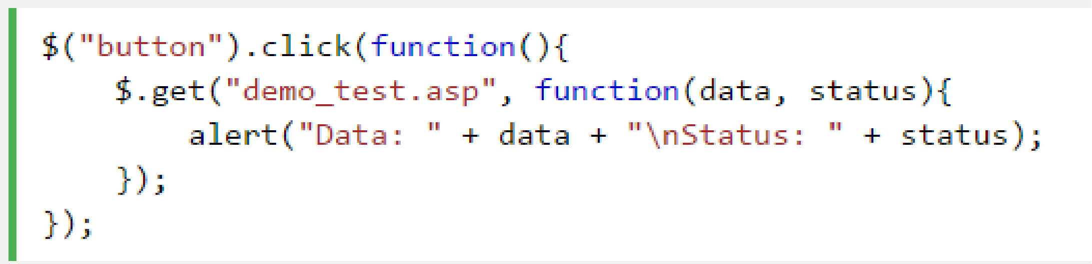
```
$.get(URL, callback);
```

    - The required URL parameter specifies the URL you wish to load.
    - The optional callback parameter is the name of a function to be executed after the load( ) method is completed.

jQuery - AJAX get() and post() Methods (cont'd)

jQuery $.get() Method (cont'd)

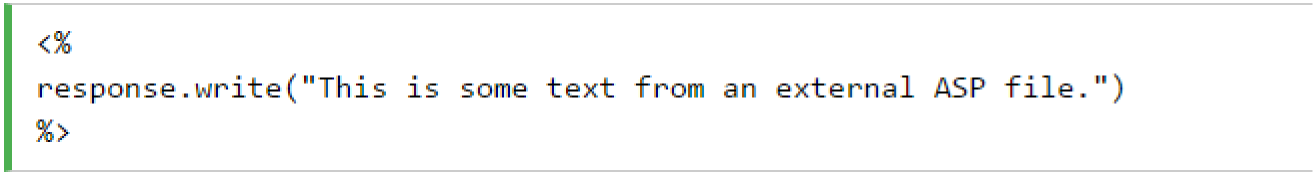
- Example:



```
$("button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

Try it!

  - The first parameter of $.get() is the URL we wish to request.
  - The second parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.
  - **Tip:** Here is how the ASP file looks like ("demo_test.asp"):



```
<%
response.write("This is some text from an external ASP file.")
%>
```

jQuery - AJAX get() and post() Methods (cont'd)

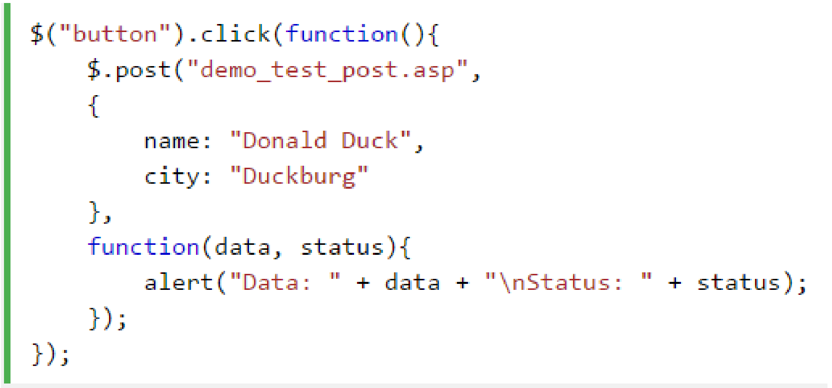
jQuery $.post() Method

- Requests data from the server with an HTTP POST request.
- Syntax: $.post (*URL*, *data*, *callback*);
  - The required URL parameter specifies the URL you wish to load.
  - The optional data parameter specifies some data to send along with the request.
  - The optional callback parameter is the name of a function to be executed after the load() method is completed.

jQuery - AJAX get() and post() Methods (cont'd)

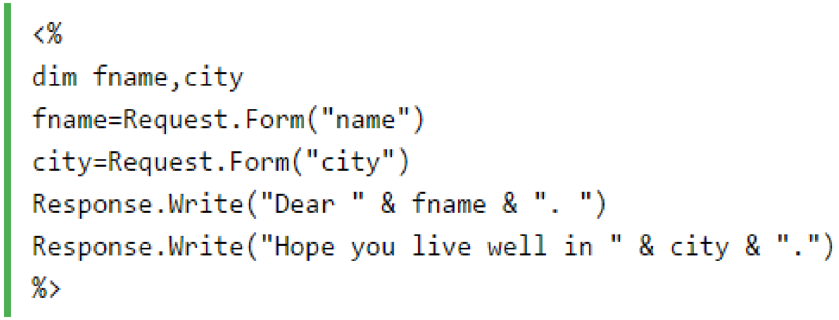
jQuery $.post() Method (cont'd)

- Example: *Try it!*
  - The first parameter of $.post() is the URL we wish to request.
  - Then pass in some data to send along with the request (name and city).
  - The ASP script in "demo_test_post.asp" reads the parameters, processes them, and returns a result.
  - The third parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.



```
$("button").click(function(){
    $.post("demo_test_post.asp",
    {
        name: "Donald Duck",
        city: "Duckburg"
    },
    function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

- Tip: Here is how the ASP file looks like ("demo_test_post.asp"):



```
<%
dim fname,city
fname=Request.Form("name")
city=Request.Form("city")
Response.Write("Dear " & fname & ". ")
Response.Write("Hope you live well in " & city & ".")
%>
```