CS418/518: Web Programming
Fall 2022

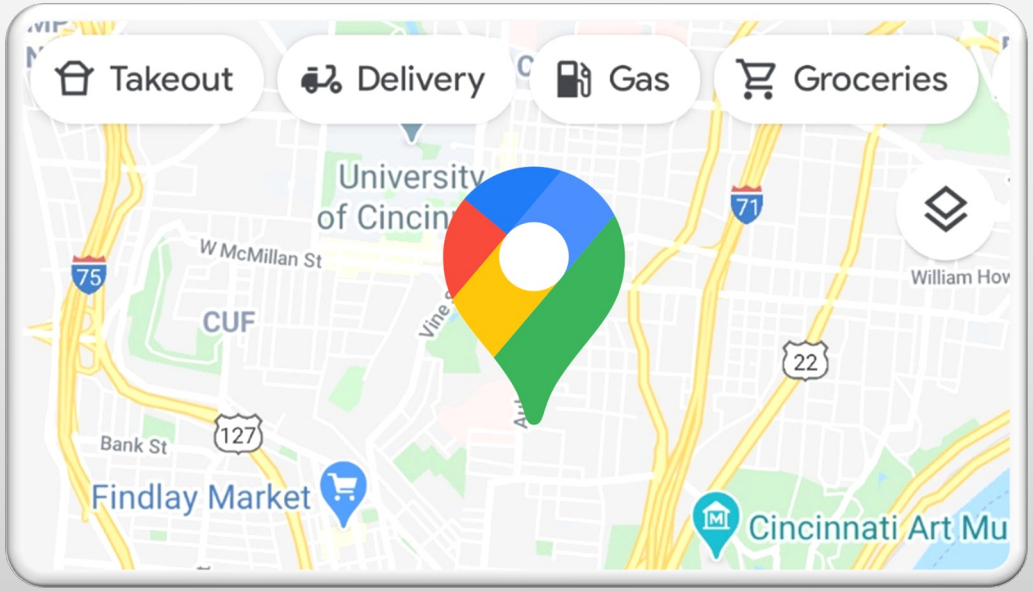
LECTURE14: ASYNCHRONOUS COMMUNICATION AND JQUERY

DR. JIAN WU

Courtesy: Dr. Justin Brunelle and Dr. Mohammed Misbhauddin

BACKGROUND - AJAX

- Joined in 2005: Asynchronous JavaScript and XML
- A set of methods built into JavaScript to transfer data between the browser and a server in the background
- Example: Google map
  - New sections of a map are downloaded from the server when needed
- Benefit:
  - Save bandwidth
  - Makes web pages seamlessly dynamic
  - Better user interaction
  - Better responsiveness



ASYNCHRONOUS COMMUNICATION

- Started since the release of Internet Explorer 5 in 1999
- ActiveX object: XMLHttpRequest
- Early adopters: Chat Rooms

XMLHttpRequest OBJECT'S PROPERTIES

Property	Description
`onreadystatechange`	Specifies an event-handling function to be called whenever the `readyState` property of an object changes.
`readyState`	An integer property that reports on the status of a request. It can have any of these values: `0` = Uninitialized, `1` = Loading, `2` = Loaded, `3` = Interactive, or `4` = Completed.
`responseText`	The data returned by the server in text format.
`responseXML`	The data returned by the server in XML format.
`status`	The HTTP status code returned by the server.
`statusText`	The HTTP status text returned by the server.

XMLHttpRequest OBJECT'S METHODS

Method	Description
`abort()`	Aborts the current request
`getAllResponseHeaders()`	Returns all headers as a string
`getResponseHeader(param)`	Returns the value of *param* as a string
`open('method', 'url', 'async')`	Specifies the HTTP method to use (GET or POST), the target URL, and whether the request should be handled asynchronously (`true` or `false`)
`send(data)`	Sends *data* to the target server using the specified HTTP method
`setRequestHeader('param', 'value')`	Sets a header with a parameter/value pair

FIRST ASYNCHRONOUS PROGRAM

- urlpost.html

```
<html> <!-- urlpost.html -->
  <head>
    <title>Asynchronous Communication Example</title>
  </head>
  <body style='text-align:center'>
    <h1>Loading a web page into a DIV</h1>
    <div id='info'>This sentence will be replaced</div>

    <script>
      params  = "url=news.com"
      request = new asyncRequest()

      request.open("POST", "urlpost.php", true)
      request.setRequestHeader("Content-type",
        "application/x-www-form-urlencoded")

      request.onreadystatechange = function()
      {
        if (this.readyState == 4)
        {
          if (this.status == 200)
          {
            if (this.responseText != null)
            {
              document.getElementById('info').innerHTML =
                this.responseText
            }
            else alert("Communication error: No data received")
          }
          else alert( "Communication error: " + this.statusText)
        }
      }

      request.send(params)
```

```
function asyncRequest()
{
  try
  {
    var request = new XMLHttpRequest()
  }
  catch(e1)
  {
    try
    {
      request = new ActiveXObject("Msxml2.XMLHTTP")
    }
    catch(e2)
    {
      try
      {
        request = new ActiveXObject("Microsoft.XMLHTTP")
      }
      catch(e3)
      {
        request = false
      }
    }
  }
  return request
}
</script>
</body>
</html>
```

creating an XMLHttpRequest object. Try three ways, to adapt to old browsers.

This creates the asynchronous object, giving you control over what data you send to the server and receive back.

This is what will be sent to the server.

Set the object to make a POST request to *urlpost.php* in asynchronous mode.

sets the onreadystatechange property to call a "callback" function of our choice each time readyState changes

Only this element of the web page changes, while everything else remains the same

readyState=4 represents a complete call

status=200 represents a successful call.

The asynchronous request is finally sent to the server. After this, all the preceding code is activated each time readyState changes.

THE SERVER HALF OF THE ASYNCHRONOUS PROCESS

load in the web page at the URL supplied to it

This is a function that should be applied to all post messages.

```
<?php // urlpost.php
  if (isset($_POST['url']))
  {
    echo file_get_contents('http://' . SanitizeString($_POST['url']));
  }

  function SanitizeString($var)
  {
    $var = strip_tags($var);
    $var = htmlentities($var);
    return stripslashes($var);
  }
?>
```

strip_tags — Strip HTML and PHP tags from a string. See https://www.php.net/manual/en/function.strip-tags

htmlentities — Convert all applicable characters to HTML entities. See https://www.php.net/manual/en/function.htmlentities.php

stripcslashes — Un-quote string quoted with addcslashes() https://www.php.net/manual/en/function.stripcslashes

LIMITATION OF NATIVE JAVASCRIPT

- Javascript is powerful and flexible: many built-in functions, but
- You still need additional layers of code for simple things
  - Animation
  - Event handling
  - Asynchronous communication
- Ensuring your website look the same on all browsers needs tedious JavaScript code
- To resolve these problems: AngularJS, jQuery, MooTools, Prototype, script.aculo.us, and YUI

WHY JQUERY?

- Most widely used: installed on more than 70% of websites (https://w3techs.com/technologies/details/js-jquery)
- High level of cross-browser compatibility
- Quick and easy access of DOM objects
- Special functions to interact directly with CSS
- Powerful tools to create professional effects and animations
- Powerful functions for conducting asynchronous communications with server
- The base for a wide range of plug-ins and other utilities

INCLUDING JQUERY

- Two ways
  - Download the version you need: https://releases.jquery.com/jquery/, upload to your web server, and reference it from a `script` tag in your HTML file
    - `<script src='http://myserver.com/jquery-3.2.1.min.js'></script>`
    - `<script src='jquery-3.2.1.min.js'></script>`
  - Free content delivery network (CDN) and link to the version you require (no local copy)
    - Recommended if you do not need to modify the jQuery source code
    - `<script src='http://code.jquery.com/jquery-3.2.1.min.js'></script>`
    - `<script src='http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.min.js'></script>`
    - `<script src='http://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js'> </script>`

DIFFERENT FLAVORS OF JQUERY

- Generally, each new version of jQuery supports these browser versions:
  - Chrome: (Current – 1) and Current
  - Edge: (Current – 1) and Current
  - Firefox: (Current – 1) and Current
  - Internet Explorer: 9+
  - Safari: (Current – 1) and Current
  - Opera: Current
- To support older browsers like Internet Explorer 6–8, Opera 12.1x, or Safari 5.1+, the jQuery developers recommend using version 1.12. (more detail: https://jquery.com/browser-support/)
- Compressed or editable: Generally, a minified version is best, but most web servers support gzip for on-the-fly compression and decompression, so this is less important.
- Slim version: The slim versions of jQuery omit the asynchronous communication functions to save on space, so you should avoid these if you use AJAX.

A SIMPLE EXAMPLE

- To change the font family of all paragraphs to monospace:
  - `$('p').css('font-family', 'monospace’)`
- To add a border to a `<code>` element, you could use this:
  - `$('code').css('border', '1px solid #aaa’)`
- Use this:
  - `$(this).css('background', '#ff0')`

A COMPLETE EXAMPLE

```
<!DOCTYPE html>
<html>
  <head>
    <title>First jQuery Example</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>
  <body>
    The jQuery library uses either the <code>$()</code>
      or <code>jQuery()</code> function names.
    <script>
      $('code').css('border', '1px solid #aaa')
    </script>
  </body>
</html>
```

Replace this with a CND link

Can you put the <script> block before the body text?

The css method allows users to dynamically alter any CSS property.

Example: simplejqueryexample.html

THE `css` METHOD

- Allows users to set the content to display with full justification.
  - `$('p').css('text-align', 'justify') // selecting an element`
  - `$('#advert').css('border', '3px dashed red') // selecting an ID`
  - `$('.new').css('text-decoration', 'underline') // selecting a class`
  - `$('blockquote, #advert, .new').css('font-weight', 'bold') // combined selector`
- Return a computed value
  - `color = $('#elem').css('color')`

Example: diffselectors.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Second jQuery Example</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>
  <body>
    <blockquote>Powerful and flexible as JavaScript is, with a plethora of
      built-in functions, it is still necessary to use additional code for
      simple things that cannot be achieved natively or with CSS, such as
      animations, event handling, and asynchronous communication.</blockquote>
    <div id='advert'>This is an ad</div>
    <p>This is my <span class='new'>new</span> website</p>
    <script>
      $('blockquote').css('background', 'lime')
      $('#advert').css('border', '3px dashed red')
      $('.new').css('text-decoration', 'underline')
      $('blockquote, #advert, .new').css('font-weight', 'bold')
    </script>
  </body>
</html>
```

Example: diffselectors.html

HANDLING EVENTS USING JQUERY

```
$('#clickme').click(function()
{
  $('#result').html('You clicked the button!')
})
```

When the element with the ID of clickme is clicked, the element with the ID of result is updated via the jQuery html function.

When accessing an event with jQuery, omit the on prefix.
onmouseover (JavaScript) -> mouseover (jQuery)
onclick (JavaScript) -> click (jQuery)

Example: processevent.html

Note: jQuery objects are NOT the same as JavaScript objects created using getElementByID().
JavaScript:

```
object = document.getElementById('result’)
object.innerHTML = 'something’
```

But

```
$('#result').innerHTML
```

won’t work, because innerHTML is NOT a property of a jQuery object.

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery Events</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>
  <body>
    <button id='clickme'>Click Me</button>
    <p id='result'>I am a paragraph</p>
    <script>
      $('#clickme').click(function()
      {
        $('#result').html('You clicked the button!')
      })
    </script>
  </body>
</html>
```

Example: processevent.html

WAITING UNTIL THE DOCUMENT IS READY

- jQuery depends on DOM
- Wait until a webpage is fully loaded
- In JavaScript, we used the "onload" event
- In jQuery, we use "ready" so you can enable it the earliest possible moment.
- Best practice: place your script at the page end and place its jQuery calls within the ready function

```
$('document').ready(function()
{
  // Your code goes here
})
```

They are equivalent

```
$(function()
{
  // Your code goes here
})
```

EVENT FUNCTIONS AND PROPERTIES

- The blue and focus events
- The click and dblclick Events
- The keypress Event
- The mouseenter and mouseleave Events
- The submit Event
- For a complete list see: https://api.jquery.com/category/events/

THE BLUR AND FOCUS EVENTS

- The blur event triggers when focus is removed from an element, causing it to blur. Similar with focus. They usually are used in pairs.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Events: blur</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>
  <body>
    <h2>Click in and out of these fields</h2>
    <input id='first'> <input> <input> <input>
    <script>
      $('#first').focus()
      $('input').focus(function() { $(this).css('background', '#ff0') } )
      $('input') .blur(function() { $(this).css('background', '#aaa') } )
    </script>
  </body>
</html>
```

this is yellow

this is grey

You are allowed to include whitespace characters before or after the period character

Example: focusblurevents.html

THE CLICK AND DBLCLICK EVENTS

If using an anonymous function:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Events: click & dblclick</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>
  <body>
    <h2>Click and double click the buttons</h2>
    <button class='myclass'>Button 1</button>
    <button class='myclass'>Button 2</button>
    <button class='myclass'>Button 3</button>
    <button class='myclass'>Button 4</button>
    <button class='myclass'>Button 5</button>
    <script>
      $('.myclass').click(    function() { $(this).slideUp() })
      $('.myclass').dblclick( function() { $(this).hide()    })
    </script>
  </body>
</html>
```

If using a named function:

```
$('.myclass').click(doslide)

function doslide()
{
  $(this).slideUp()
}
```

slideUp: disappear with an animation

hide: just vanish

THE KEYPRESS EVENT

```
<!DOCTYPE html>
<html>
  <head>
    <title>Events: keypress</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>
  <body>
    <h2>Press some keys</h2>
    <div id='result'></div>
    <script>
      $(document).keypress(function(event)
      {
        key = String.fromCharCode(event.which)

        if (key >= 'a' && key <= 'z' ||
            key >= 'A' && key <= 'Z' ||
            key >= '0' && key <= '9')
        {
          $('#result').html('You pressed: ' + key)
          event.preventDefault()
        }
      })
    </script>
  </body>
</html>
```

the which property of the event object is normalized by jQuery to return the same character codes across all browsers.

turn it into a single-letter string using fromCharCode()

Accept only the characters a–z, A–Z, and 0–9, ignoring all others. You should not assume users always input the right characters.

preventDefault keeps the event from "bubbling up" to other handlers.

Example: keypressevents.html

THE MOUSEENTER AND MOUSELEAVE EVENTS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Events: Further Mouse Handling</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>

  <body>
    <h2 id='test'>Pass the mouse over me</h2>
    <script>
      $('#test').mouseenter(function() { $(this).html('Hey, stop tickling!') } )
      $('#test').mouseleave(function() { $(this).html('Where did you go?')   } )
    </script>
  </body>
</html>
```

See more mouse events: https://api.jquery.com/category/events/mouse-events/

THE SUBMIT EVENT

- Error checking on the data entered before it gets sent to the server

```
<!DOCTYPE html>
<html>
  <head>
    <title>Events: submit</title>
    <script src='jquery-3.2.1.min.js'></script>
  </head>
  <body>
    <form id='form'>
      First name: <input id='fname' type='text' name='fname'><br>
      Last name:  <input id='lname' type='text' name='lname'><br>
      <input type='submit'>
    </form>
    <script>
      $('#form').submit(function()
      {
        if ($('#fname').val() == '' ||
            $('#lname').val() == '')
        {
          alert('Please enter both names')
          return false
        }
      })
    </script>
  </body>
</html>
```

val method is used to retrieve the value in the value property of each field