

LEARN. DO. EARN

ACADGILD



FRONT END DEVELOPMENT (WITH ANGULARJS)

Website : <http://www.acadgild.com>
LinkedIn : <https://www.linkedin.com/company/acadgild>
Facebook : <https://www.facebook.com/acadgild>

© copyright ACADGILD



Session 6 – jQuery





Agenda – jQuery

1. **jQuery Events**
2. **Events – Mouse, Keyboard, Form**
3. **jQuery Forms**
4. **Form Event – Submit()**
5. **Event Delegation**
6. **jQuery – AJAX Introduction**
7. **jQuery AJAX Method – load()**
8. **jQuery AJAX Method – get(), post()**
9. **jQuery – noConflict()**
10. **jQuery Template**
11. **jQuery Promise**





jQuery Events

- jQuery provides methods for assigning event handlers to elements in a cross-browser way
- All the actions that a web page can respond to are called events
- An event represents the moment when something happens
- Examples:
 - moving a mouse over an element
 - selecting a radio button
 - clicking on an element

```
$('#a').click(function(ev) {  
  $(this).css({backgroundColor: 'orange'});  
  ev.preventDefault(); });
```





Event – Mouse, Keyboard, Form

- Here are some common DOM events:
 - Mouse Events
 - Click
 - Dblclick
 - Mouseenter
 - Mouseleave
 - Keyboard events
 - Keypress
 - Keydown
 - keyup
 - Form Events
 - Submit
 - Change
 - Focus
 - Blur





Selecting Form Elements:

- Assign an ID to it

```
<input name="user" type="text" id="user">  
var userfield = $('#user');
```

- Manipulating Selected form element

```
.val will get the value of the element  
var userfield = $('#user').val();
```

`$(':text')`

Selecting all Form Elements of certain type:

- Selects all text fields
- Use with :input (all form elements), :password, :radio, :checkbox, :submit, :image, :reset, :button, :file, :hidden
- Can use descendant selectors as well

```
$('#signup : text')
```





jQuery Forms(contd.)

Set the value of a form element

```
var fieldvalue = $('#total').val(Yourvalue);
```

Determine if box is checked

```
If ($('#total').attr('checked'))
```

```
{
```

```
    Do stuff if box is checked
```

```
}
```

```
else
```

```
{
```

```
    Do stuff if box is not checked
```

```
}
```





Form Event – Submit()

Example:

```
$(document).ready(function() {  
    $('#signup').submit(function() {  
        if ($('#username').val() == "") {  
            alert ('Please supply name to name field');  
            return false;  
        }  
    })  
});
```





Event Delegation

- **Event delegation** is an efficient way to watch for an event over a large number of elements.
- **delegate()** method attaches one or more event handlers for specified elements.
- Specifies a function to run when the events occurs.
- Event handlers attached using the delegate() method will work for both current and future elements (like a new element created by a script)

Syntax : *\$(selector).delegate(childSelector,event,data,function)*

- You can also make use of “On” function for delegation.
- Works by binding to a point further up the DOM tree and watching for bubbling events.
- **jQuery** has two ways to do event delegation:
 - .live() (as a direct replacement for .bind())
 - .delegate().
- **Works on all current and future elements.**





jQuery – AJAX Introduction

- AJAX is an acronym for **A**synchronous **J**avaScript and **X**ML.
- AJAX is **the art of exchanging data with a server**, and updating parts of a web page without reloading the whole page.
- jQuery **provides several methods for AJAX functionality** to develop next generation web applications.
- 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 also load the external data directly into the selected HTML elements of your web page.





jQuery AJAX Method – load()

- The **jQuery load()** method is a simple but powerful AJAX method.
- The load() method loads data from a server and puts the returned data into the selected element.

Syntax:

```
$(selector).load(URL,data,callback);
```





jQuery AJAX Method – get(), post()

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

Two commonly used methods for a request-response between a client and server are: GET and POST.

 - GET - Requests data from a specified resource
 - POST - Submits data to be processed to a specified resource
- **GET** is used to retrieve or get some data from the server. (The GET method may return cached data)
- **POST** can also be used to get some data from the server but POST method never caches data. It is often used to send data along with the request.





jQuery AJAX Method – get(), post()(contd.)

- The **\$.get()** method requests data from the server with an HTTP GET request

Syntax:

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

- The **\$.post()** method requests data from the server using an HTTP POST request.

Syntax:

```
$.post(URL,data,callback);
```





jQuery – noConflict()

- The **noConflict() method** releases the hold on the \$ shortcut identifier, so that other scripts can use it.
- You can of course still use jQuery by writing the full name instead of the shortcut.

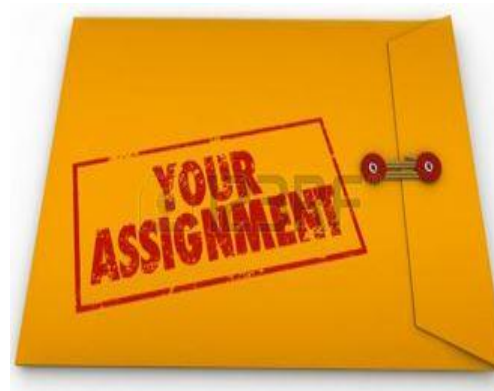
```
$.noConflict();  
jQuery(document).ready(function(){  
    jQuery("button").click(function(){  
        jQuery("p").text("jQuery is still working!");  
    });  
});
```

- Note: jQuery uses the \$ sign as a shortcut for jQuery.





Lets Discuss Assignments



Assignment





Get in Touch with ACADGILD

Contact Info:

- Website : <http://www.acadgild.com>
- LinkedIn : <https://www.linkedin.com/company/acadgild>
- Facebook : <https://www.facebook.com/acadgild>
- Support: support@acadgild.com

