

LEARN. DO. EARN

ACADGILD

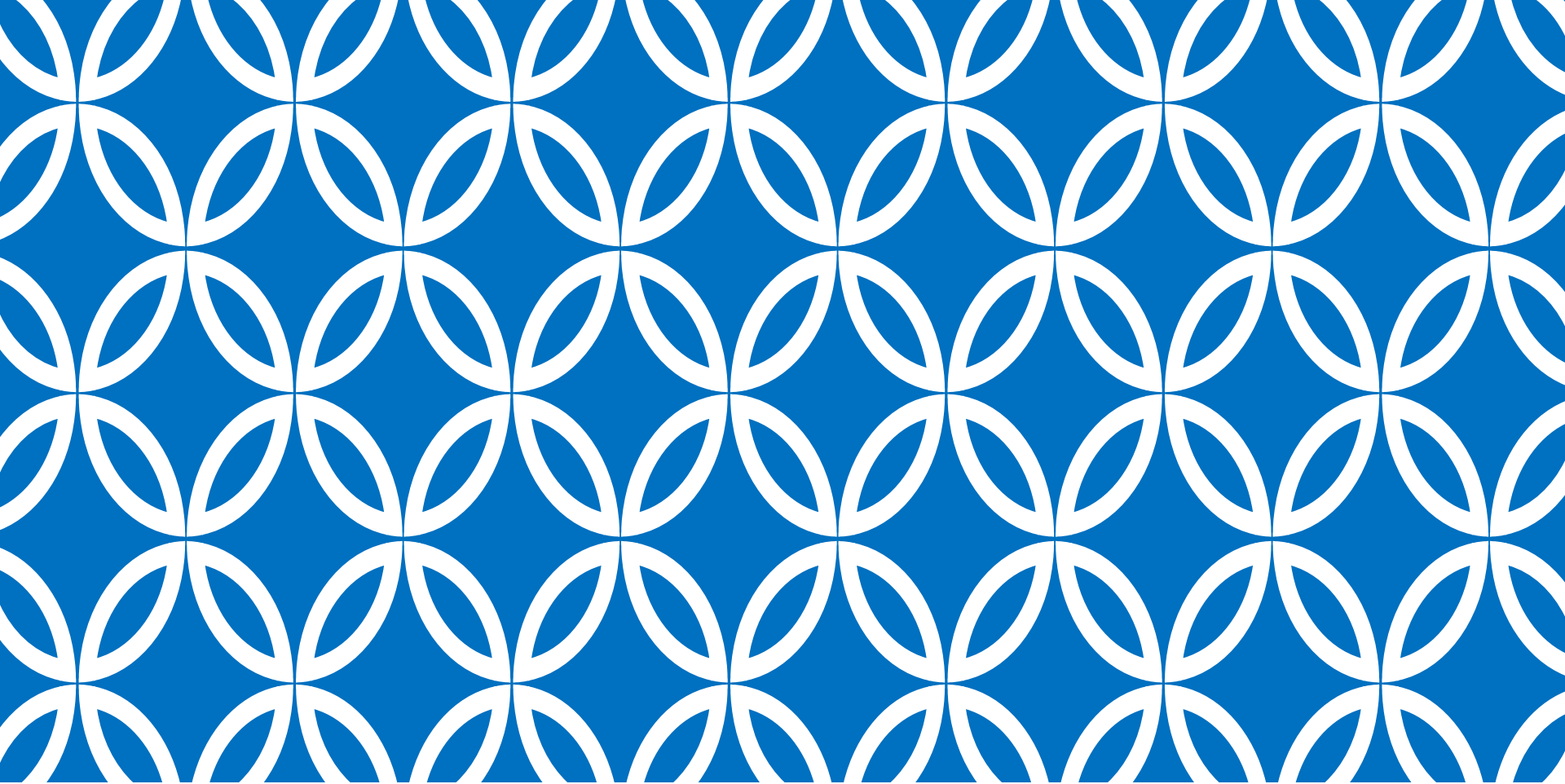


FRONT END DEVELOPMENT (WITH ANGULARJS)



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

© copyright ACADGILD



Session 10 – jQuery





Agenda – jQuery

Sl No.	Agenda Title
1	Events
2	Types of Events
3	Querying Input Elements
4	Handle Form Submission
5	Event Delegation
6	Ajax API
7	Applying HTTP Verbs Using Ajax API
8	Handling Conflicts with Other Libraries
9	Mustache.js
10	Understanding Promise Pattern





Events

- **Events** represents any action on a web page.
- jQuery provides methods for assigning event handlers to elements consistently across browsers.
- **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(); });
```





Types of Events

- Here are some **common events**:

- Mouse Events:

- Click
 - Dblclick
 - Mouseenter
 - Mouseleave

- Keyboard events:

- Keypress
 - Keydown
 - Keyup

- Form Events:

- Submit
 - Change
 - Focus
 - Blur





Querying Input Element

Reffer by ID

HTML :

```
<input name="user" type="text" id="user">
```

JS:

```
var userfield = $('#user');  
//get the value of the textbox  
var value=userfield.val();  
    //set the value of text box  
    userfield.val('updated');
```

Refer by pseudo-selector

HTML :

```
<input type="password" id="user-password"/>
```

JS :

```
var password=$('#:password').val();
```





Querying Input Element (contd.)

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')
```

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
```

```
}
```





Handle Form Submission

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 with 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)`

- Works by binding to a point further up the DOM tree and watching for bubbling events.





Event Delegation (contd.)

- **jQuery** has two ways to do event delegation:

- `.live()` (as a direct replacement for `.bind()`)
- `.delegate()`.

- **Works on all current and future elements.**

As of jQuery version 1.7, the `on()` method is the new replacement for the `bind()`, `live()` and `delegate()` methods.

Syntax :

`$(selector).on(childSelector,event,data,function)`





Ajax API

- 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.





Applying HTTP Verbs Using Ajax API

- 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);`

- **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 .





Applying HTTP Verbs using Ajax API (contd.)

- **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.
- 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);
```





Handling Conflicts with Other Libraries

- 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.

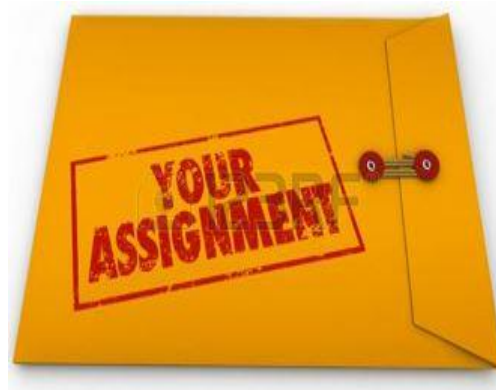
```
$.noConflict(); jQuery(document).ready(function(){
    jQuery("button").click(function(){
        jQuery("p").text("jQuery is still working!");
    });
});
```

- **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

