



WESTCLIFF
UNIVERSITY

Educate. Inspire. Empower.

WEEK 1 DAY 2: jQuery



Agenda

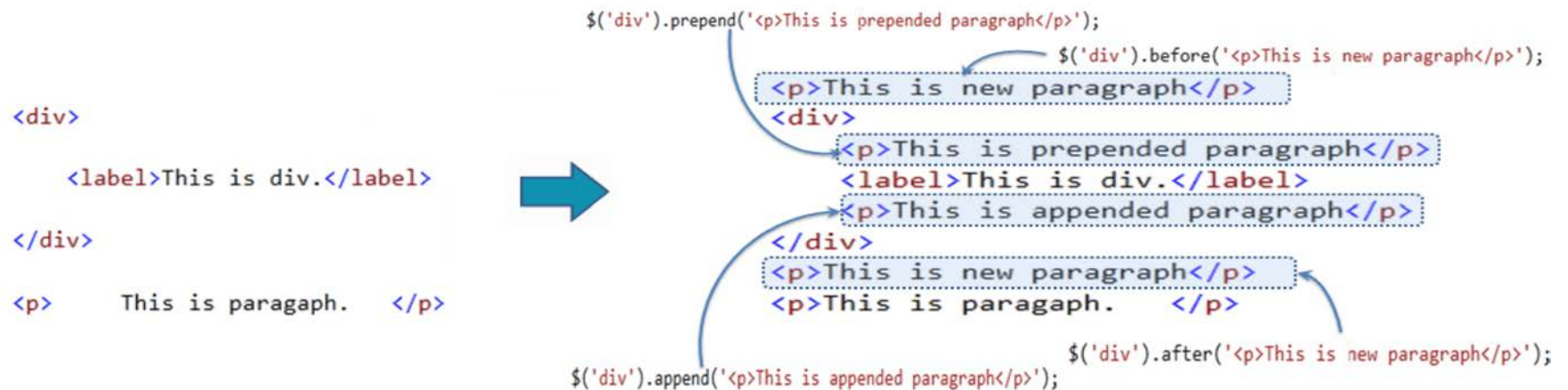
1. jQuery DOM
2. jQuery Event
3. jQuery Ajax
4. jQuery Effects
5. In-Class Demo Exercise
6. Homework



jQuery DOM Manipulation

jQuery DOM

- provides various methods to add, edit or delete DOM element(s) in the HTML page.
- provides methods to manipulate DOM in efficient way. You do not need to write big code to modify the value of any element's attribute or to extract HTML code from a paragraph or division.
- provides methods such as .attr(), .html(), and .val() which act as getters, retrieving information from DOM elements for later use.





jQuery DOM Manipulation

#1

Method	Description
<code>after(content)</code>	Insert content after each of the matched elements.
<code>append(content)</code>	Append content to the inside of every matched element.
<code>appendTo(selector)</code>	Append all of the matched elements to another, specified, set of elements.
<code>before(content)</code>	Insert content before each of the matched elements.
<code>clone(bool)</code>	Clone matched DOM Elements, and all their event handlers, and select the clones.
<code>clone()</code>	Clone matched DOM Elements and select the clones.



jQuery DOM Manipulation

#2

Method	Description
<code>empty()</code>	Remove all child nodes from the set of matched elements.
<code>html(val)</code>	Set the html contents of every matched element.
<code>html()</code>	Get the html contents (innerHTML) of the first matched element.
<code>insertAfter(selector)</code>	Insert all of the matched elements after another, specified, set of elements.
<code>insertBefore(selector)</code>	Insert all of the matched elements before another, specified, set of elements.
<code>prepend(content)</code>	Prepend content to the inside of every matched element.



jQuery DOM Manipulation

#3

Method	Description
<code>prependTo(selector)</code>	Prepend all of the matched elements to another, specified, set of elements.
<code>remove(expr)</code>	Removes all matched elements from the DOM.
<code>replaceAll(selector)</code>	Replaces the elements matched by the specified selector with the matched elements.
<code>replaceWith(content)</code>	Replaces all matched elements with the specified HTML or DOM elements.
<code>text(val)</code>	Set the text contents of all matched elements.
<code>text()</code>	Get the combined text contents of all matched elements.



jQuery DOM Manipulation

#4

Method	Description
<code>wrap(elem)</code>	Wrap each matched element with the specified element.
<code>wrap(html)</code>	Wrap each matched element with the specified HTML content.
<code>wrapAll(elem)</code>	Wrap all the elements in the matched set into a single wrapper element.
<code>wrapAll(html)</code>	Wrap all the elements in the matched set into a single wrapper element.
<code>wrapInner(elem)</code>	Wrap the inner child contents of each matched element (including text nodes) with a DOM element.
<code>wrapInner(html)</code>	Wrap the inner child contents of each matched element (including text nodes) with an HTML structure.



jQuery DOM Manipulation Example

```
<html>
<head>
  <title>Westcliff Fullstack Bootcamp jQuery Dom Manipulation Example</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
      $("div").click(function () {
        $(this).before('<div class="div"></div>' );
      });
    });
  </script>

  <style>
    .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
  </style>
</head>
<body>
  <p>Click on any square below:</p>
  <span id = "result"> </span>

  <div class = "div" style = "background-color: blue;"></div>
  <div class = "div" style = "background-color: green;"></div>
  <div class = "div" style = "background-color: red;"></div>
</body>
</html>
```


jQuery Events

- Actions that can be detected and/or fired by/from your Web Application.
- Examples of Events:
 - A mouse click
 - A web page loading
 - Taking mouse over an element
 - Submitting an HTML form
 - A keystroke on your keyboard, etc.





jQuery Event Types

#1

Event Type	Description
blur	Occurs when the element loses focus.
change	Occurs when the element changes.
click	Occurs when a mouse click.
dblclick	Occurs when a mouse double-click.
error	Occurs when there is an error in loading or unloading etc.
focus	Occurs when the element gets focus.



jQuery Event Types

#2

Sr.No.	Event Type & Description
keydown	Occurs when key is pressed.
keypress	Occurs when key is pressed and released.
keyup	Occurs when key is released.
load	Occurs when document is loaded.
mousedown	Occurs when mouse button is pressed.
mouseenter	Occurs when mouse enters in an element region.



jQuery Event Types

#3

Sr.No.	Event Type & Description
mouseleave	Occurs when mouse leaves an element region.
mousemove	Occurs when mouse pointer moves.
mouseout	Occurs when mouse pointer moves out of an element.
mouseover	Occurs when mouse pointer moves over an element.
mouseup	Occurs when mouse button is released.
resize	Occurs when window is resized.



jQuery Event Types

#4

Sr.No.	Event Type & Description
scroll	Occurs when window is scrolled.
select	Occurs when a text is selected.
submit	Occurs when form is submitted.
unload	Occurs when documents is unloaded.



jQuery Event Attributes

#1

Property	Description
altKey	Set to true if the Alt key was pressed when the event was triggered, false if not. The Alt key is labeled Option on most Mac keyboards.
ctrlKey	Set to true if the Ctrl key was pressed when the event was triggered, false if not.
data	The value, if any, passed as the second parameter to the bind() command when the handler was established.
keyCode	For keyup and keydown events, this returns the key that was pressed.
metaKey	Set to true if the Meta key was pressed when the event was triggered, false if not. The Meta key is the Ctrl key on PCs and the Command key on Macs.



jQuery Event Attributes

#2

Property	Description
pageX	For mouse events, specifies the horizontal coordinate of the event relative from the page origin.
pageY	For mouse events, specifies the vertical coordinate of the event relative from the page origin.
relatedTarget	For some mouse events, identifies the element that the cursor left or entered when the event was triggered.
screenX	For mouse events, specifies the horizontal coordinate of the event relative from the screen origin.
screenY	For mouse events, specifies the vertical coordinate of the event relative from the screen origin.
shiftKey	Set to true if the Shift key was pressed when the event was triggered, false if not.



jQuery Event Attributes

#3

Property	Description
target	Identifies the element for which the event was triggered.
timeStamp	The timestamp (in milliseconds) when the event was created.
type	For all events, specifies the type of event that was triggered (for example, click).
which	For keyboard events, specifies the numeric code for the key that caused the event, and for mouse events, specifies which button was pressed (1 for left, 2 for middle, 3 for right).



jQuery Event Methods

Method	Description
<code>preventDefault()</code>	Prevents the browser from executing the default action.
<code>isDefaultPrevented()</code>	Returns whether <code>event.preventDefault()</code> was ever called on this event object.
<code>stopPropagation()</code>	Stops the bubbling of an event to parent elements, preventing any parent handlers from being notified of the event.
<code>isPropagationStopped()</code>	Returns whether <code>event.stopPropagation()</code> was ever called on this event object.
<code>stopImmediatePropagation()</code>	Stops the rest of the handlers from being executed.
<code>isImmediatePropagationStopped()</code>	Returns whether <code>event.stopImmediatePropagation()</code> was ever called on this event object.



jQuery Event Manipulation Methods #1

Method	Description
<code>bind(type, [data], fn)</code>	Binds a handler to one or more events (like click) for each matched element. Can also bind custom events.
<code>off(events, [selector], [handler(eventObject)])</code>	This does the opposite of live, it removes a bound live event.
<code>hover(over, out)</code>	Simulates hovering for example moving the mouse on, and off, an object.
<code>on(events, [selector], [data], handler)</code>	Binds a handler to an event (like click) for all current – and future – matched element. Can also bind custom events.
<code>one(type, [data], fn)</code>	Binds a handler to one or more events to be executed once for each matched element.



jQuery Event Manipulation Methods #2

Method	Description
<code>ready(fn)</code>	Binds a function to be executed whenever the DOM is ready to be traversed and manipulated.
<code>trigger(event, [data])</code>	Trigger an event on every matched element.
<code>triggerHandler(event, [data])</code>	Triggers all bound event handlers on an element.
<code>unbind([type], [fn])</code>	This does the opposite of bind, it removes bound events from each of the matched elements.



jQuery Event Helper Methods

#1

Method	Description
<code>blur()</code>	Triggers the blur event of each matched element.
<code>blur(fn)</code>	Bind a function to the blur event of each matched element.
<code>change()</code>	Triggers the change event of each matched element.
<code>change(fn)</code>	Binds a function to the change event of each matched element.
<code>click()</code>	Triggers the click event of each matched element.
<code>click(fn)</code>	Binds a function to the click event of each matched element.
<code>dblclick()</code>	Triggers the dblclick event of each matched element.



jQuery Event Helper Methods

#2

Method	Description
<code>dblclick(fn)</code>	Binds a function to the <code>dblclick</code> event of each matched element.
<code>error()</code>	Triggers the <code>error</code> event of each matched element.
<code>error(fn)</code>	Binds a function to the <code>error</code> event of each matched element.
<code>focus()</code>	Triggers the <code>focus</code> event of each matched element.
<code>focus(fn)</code>	Binds a function to the <code>focus</code> event of each matched element.
<code>keydown()</code>	Triggers the <code>keydown</code> event of each matched element.
<code>keydown(fn)</code>	Bind a function to the <code>keydown</code> event of each matched element.



jQuery Event Helper Methods

#3

Method	Description
keypress()	Triggers the keypress event of each matched element.
keypress(fn)	Binds a function to the keypress event of each matched element.
keyup()	Triggers the keyup event of each matched element.
keyup(fn)	Bind a function to the keyup event of each matched element.
load(fn)	Binds a function to the load event of each matched element.
mousedown(fn)	Binds a function to the mousedown event of each matched element.
mouseenter(fn)	Bind a function to the mouseenter event of each matched element.



jQuery Event Helper Methods

#4

Method	Description
mouseleave(fn)	Bind a function to the mouseleave event of each matched element.
mousemove(fn)	Bind a function to the mousemove event of each matched element.
mouseout(fn)	Bind a function to the mouseout event of each matched element.
mouseover(fn)	Bind a function to the mouseover event of each matched element.
mouseup(fn)	Bind a function to the mouseup event of each matched element.
resize(fn)	Bind a function to the resize event of each matched element.
scroll(fn)	Bind a function to the scroll event of each matched element.



jQuery Event Helper Methods

#5

Method	Description
<code>select()</code>	Trigger the select event of each matched element.
<code>select(fn)</code>	Bind a function to the select event of each matched element.
<code>submit()</code>	Trigger the submit event of each matched element.
<code>submit(fn)</code>	Bind a function to the submit event of each matched element.
<code>unload(fn)</code>	Binds a function to the unload event of each matched element.



jQuery Events

```
<html>
<head>
  <title>Westcliff Fullstack Bootcamp jQuery Event Example</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $('div').bind('click', function (event) {
        alert('Event type is ' + event.type);
        alert('pageX : ' + event.pageX);
        alert('pageY : ' + event.pageY);
        alert('Target : ' + event.target.innerHTML);
      });
    });
  </script>
  <style>
    .div {
      margin: 10px;
      padding: 12px;
      border: 2px solid #666;
      width: 60px;
    }
  </style>
</head>
<body>
  <p>Click on any square below to see the result:</p>

  <div class="div" style="background-color: blue;">ONE</div>
  <div class="div" style="background-color: green;">TWO</div>
  <div class="div" style="background-color: red;">THREE</div>
</body>
</html>
```



jQuery Ajax

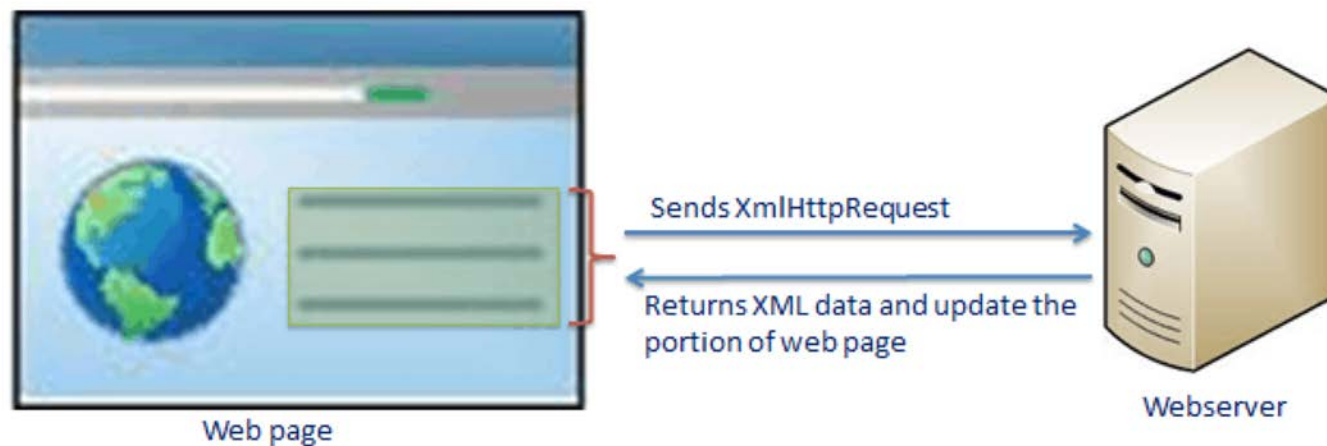
Ajax stands for "**A**synchronous **J**avaScript **a**nd **X**ML".

JavaScript includes features of sending asynchronous http request using *XMLHttpRequest* object. Ajax is about using this ability of JavaScript to send asynchronous http request and get the xml data as a response (also in other formats) and update the part of a web page (using JavaScript) without reloading or refreshing entire web page.

With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!

jQuery Ajax

The following figure illustrates the Ajax functionality.



jQuery Ajax

- Here's the Syntax for a method:

`selector.method()`

- For example:
 - `selector.html()`
 - `selector.load()`
 - `selector.append()`
- Script examples:
 - `$('#div').html('Name:' + team.name + '')`
 - `$('#info').load('results.html')`
 - `$('#p .p1').append(data)`



jQuery Ajax

- As an example, let's look at the `load()` method in a little more detail and understand how it works. The syntax for this method is:

`[selector].load(URL, [data], [callback]);`

- **URL** – The URL of the server-side resource to which the request is sent.
- **data** – This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the POST method. If omitted, the GET method is used.
- **callback** – A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.



jQuery Ajax

- So we have the following jQuery script:

```
$(document).ready(function() {  
    $("#btn").click(function(event){  
        $('#showinfo').load('/jq/result.html');  
    });  
});
```

- And the following html:

```
<p>Click button to load result.html file</p>  
  
<div id = "showinfo" style = "background-color:cc0;">SHOW DATA</div>  
  
<input type = "button" id = "btn" value = "Load Data" />
```

- Here load() initiates an Ajax request to the specified URL /jq/result.html file. After loading this file, all the content would be populated inside <div> tagged with ID *showinfo*. Assuming, our /jq/result.html file has just one HTML line:

```
<h1>THIS IS RESULT...</h1>
```



jQuery Ajax

- Let's look at another example - the `getJSON()` method in a little more detail and understand how it works as well. The syntax for this method is:

`[selector].getJSON(URL, [data], [callback]);`

- **URL** – The URL of the server-side resource to which the request is sent.
- **data** – This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the POST method. If omitted, the GET method is used.
- **callback** – A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.



jQuery Ajax

- So we have the following jQuery script:

```
$(document).ready(function(){  
    $("btn").click(function(event){  
        $.getJSON('/jq/result.json', function(jqdata) {  
            $('#showinfo').html('<p> Name: ' + jqdata.name + '</p>');  
            $('#showinfo').append('<p>Age : ' + jqdata.age+ '</p>');  
            $('#showinfo').append('<p> Sex: ' + jqdata.sex+ '</p>');  
        });  
    });  
});
```

- And the following html:

```
<p>Click button to load result.json file</p>  
<div id = "showinfo" style = "background-color:#eee;">SHOW DATA</div>  
<input type = "button" id = "btn" value = "Load Data" />
```
- Here JQuery utility method `getJSON()` initiates an Ajax request to the specified URL `result.json` file. After loading this file, all the content would be passed to the callback function which finally would be populated inside `<div>` tagged with ID `stage`. Assuming, our `result.json` file has following json formatted content:

```
{  
    "name": "Zara Ali",  
    "age" : "67",  
    "sex": "female"  
}
```




jQuery Ajax Methods

#1

Methods	Description
jQuery.ajax(options)	Load a remote page using an HTTP request.
jQuery.ajaxSetup(options)	Setup global settings for AJAX requests.
jQuery.get(url, [data], [callback], [type])	Load a remote page using an HTTP GET request.
jQuerygetJSON(url, [data], [callback])	Load JSON data using an HTTP GET request.
jQuery.getScript(url, [callback])	Loads and executes a JavaScript file using an HTTP GET request.
jQuery.post(url, [data], [callback], [type])	Load a remote page using an HTTP POST request.



jQuery Ajax Methods

#2

Methods	Description
<code>load(url, [data], [callback])</code>	Load HTML from a remote file and inject it into the DOM.
<code>serialize()</code>	Serializes a set of input elements into a string of data.
<code>serializeArray()</code>	Serializes all forms and form elements like the <code>.serialize()</code> method but returns a JSON data structure for you to work with.



jQuery Ajax Events

Methods	Description
<code>ajaxComplete(callback)</code>	Attach a function to be executed whenever an AJAX request completes.
<code>ajaxStart(callback)</code>	Attach a function to be executed whenever an AJAX request begins and there is none already active.
<code>ajaxError(callback)</code>	Attach a function to be executed whenever an AJAX request fails.
<code>ajaxSend(callback)</code>	Attach a function to be executed before an AJAX request is sent.
<code>ajaxStop(callback)</code>	Attach a function to be executed whenever all AJAX requests have ended.
<code>ajaxSuccess(callback)</code>	Attach a function to be executed whenever an AJAX request completes successfully.



jQuery Ajax Example

```
<html>
<head>
  <title>Westcliff Fullstack Bootcamp jQuery Ajax Example</title>
  <script type = "text/javascript"
    src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
  </script>

  <script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
      $("#btn").click(function(event){
        $.getJSON('/jq/result.json', function(jqdata) {
          $('#showinfo').html('<p> Name: ' + jqdata.name + '</p>');
          $('#showinfo').append('<p>Age : ' + jqdata.age+ '</p>');
          $('#showinfo').append('<p> Sex: ' + jqdata.sex+ '</p>');
        });
      });
    });
  </script>
</head>

<body>
  <p>Click button to load result.json file</p>

  <div id = "showinfo" style = "background-color: #eee;">SHOW DATA</div>

  <input type = "button" id = "btn" value = "Load Data" />
</body>
</html>
```



jQuery Effects

jQuery provides a simple interface for doing various kind of visual effects. jQuery methods allow us to quickly apply commonly used effects with a minimum configuration. Effects can use the built-in settings, or provide a customized duration. You can also create custom animations of arbitrary CSS properties.

We will use one of the most common effects as an example: show/hide elements. The commands for showing and hiding elements are pretty much what we would expect – `show()` to show the elements in a wrapped set and `hide()` to hide them.

jQuery Effects

Here is the simple syntax for `show()` method:

```
[selector].show( speed, [callback] );
```

And here is the syntax for `hide()` method:

```
[selector].hide( speed, [callback] );
```

- **speed** – A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
- **callback** – This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.



jQuery Effects

We have the following jQuery scripts:

```
$(document).ready(function() {  
    $("#show").click(function () {  
        $(".mydiv").show( 1000 );  
    });  
    $("#hide").click(function () {  
        $(".mydiv").hide( 1000 );  
    });  
});
```

And the following HTML:

```
<div class = "mydiv">I Can Show and Hide!</div>  
  
<input id = "hide" type = "button" value = "Hide" />  
<input id = "show" type = "button" value = "Show" />
```

And CSS:

```
<style>  
    .mydiv{  
        margin:10px;  
        padding:12px;  
        font-size: 2em;  
        background-color:rgb(236, 196, 65);  
        width:90%;  
        height:150px;  
    }  
</style>
```



jQuery Effects

#1

Methods	Description
<code>animate(params, [duration, easing, callback])</code>	A function for making custom animations.
<code>fadeIn(speed, [callback])</code>	Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.
<code>fadeOut(speed, [callback])</code>	Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion.
<code>fadeTo(speed, opacity, callback)</code>	Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.
<code>hide()</code>	Hides each of the set of matched elements if they are shown.



jQuery Effects

#2

Methods	Description
<code>hide(speed, [callback])</code>	Hide all matched elements using a graceful animation and firing an optional callback after completion.
<code>show()</code>	Displays each of the set of matched elements if they are hidden.
<code>show(speed, [callback])</code>	Show all matched elements using a graceful animation and firing an optional callback after completion.
<code>slideDown(speed, [callback])</code>	Reveal all matched elements by adjusting their height and firing an optional callback after completion.
<code>slideToggle(speed, [callback])</code>	Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.
<code>slideUp(speed, [callback])</code>	Hide all matched elements by adjusting their height and firing an optional callback after completion.



jQuery Effects

#3

Methods	Description
<code>stop([clearQueue, gotoEnd])</code>	Stops all the currently running animations on all the specified elements.
<code>toggle()</code>	Toggle displaying each of the set of matched elements.
<code>toggle(speed, [callback])</code>	Toggle displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion.
<code>toggle(switch)</code>	Toggle displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements).
<code>jQuery.fx.off</code>	Globally disable all animations.



UI Library Based Effects

#1

Sr.No.	Methods & Description
Blind	Blinds the element away or shows it by blinding it in.
Bounce	Bounces the element vertically or horizontally n-times.
Clip	Clips the element on or off, vertically or horizontally.
Drop	Drops the element away or shows it by dropping it in.
Explode	Explodes the element into multiple pieces.
Fold	Folds the element like a piece of paper.



UI Library Based Effects

#2

Sr.No.	Methods & Description
Highlight	Highlights the background with a defined color.
Puff	Scale and fade out animations create the puff effect.
Pulsate	Pulsates the opacity of the element multiple times.
Scale	Shrink or grow an element by a percentage factor.
Shake	Shakes the element vertically or horizontally n-times.
Size	Resize an element to a specified width and height.



UI Library Based Effects

#3

Sr.No.	Methods & Description
Slide	Slides the element out of the viewport.
Transfer	Transfers the outline of an element to another.



jQuery Effects Example

```
<html>
<head>
  <title>Westcliff Fullstack Bootcamp jQuery Effects Example</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $(".clickme").click(function (event) {
        $(".target").toggle('slow', function () {
          $(".log").text('Transition Complete');
        });
      });
    });
  </script>
  <style>
    .clickme {
      margin: 10px;
      padding: 12px;
      border: 2px solid #666;
      width: 100px;
      height: 50px;
    }
  </style>
</head>
<body>
  <div class="content">
    <div class="clickme">Click Me</div>
    <div class="target">
      
    </div>
    <div class="log"></div>
  </div>
</body>
</html>
```

Questions?

Resources

<https://www.w3schools.com/jquery/>
<https://learn.jquery.com/using-jquery-core/>