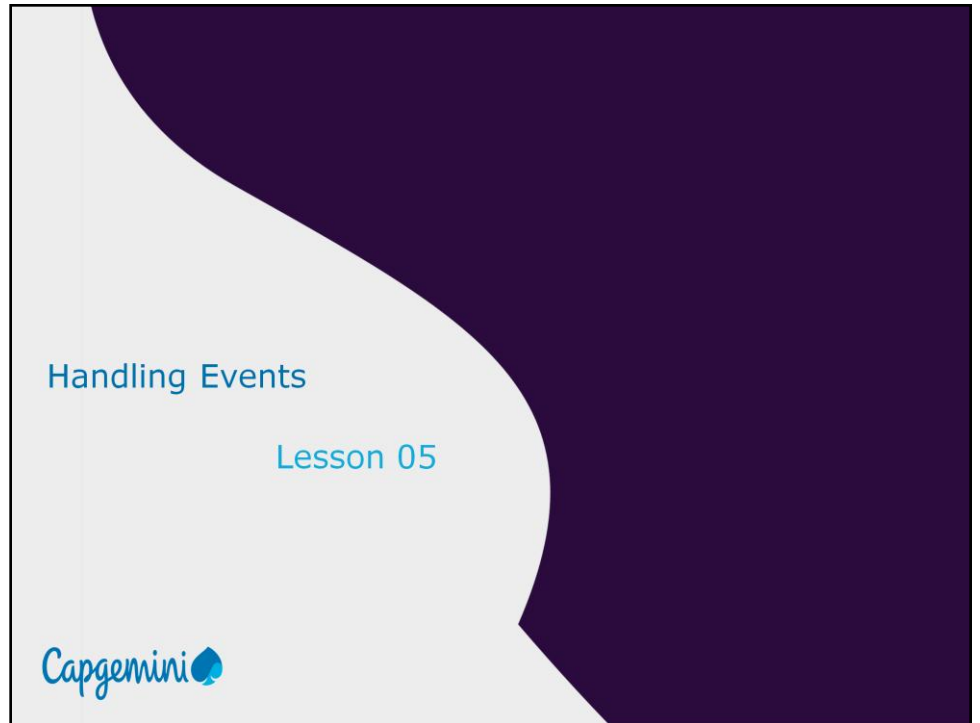


Instructor Notes:



Instructor Notes:

Explain the lesson coverage

Lesson Objectives



- Introduction
- jQuery Event Model Benefits
- Attach and detach Events from DOM Elements
- Triggering Events
- Passing data through events



Instructor Notes:

Additional notes for instructor

5.1. Introduction Introduction



- All the different visitor's actions that a web page can respond to are called events.
- An event represents the precise moment when something happens.
- Examples:
 - moving a mouse over an element
 - selecting a radio button
 - clicking on an element

Capgemini 

Instructor Notes:

Additional notes for instructor

5.2. jQuery Event Model Benefits

jQuery Event Model Benefits



- Events notify a program that a user performed some type of action
- jQuery provides a cross-browser event model that works common across all browsers.
- jQuery event model is simple to use and provides a compact syntax

Capgemini 

Instructor Notes:

Additional notes for instructor

5.2. jQuery Event Model Benefits

jQuery Events

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Capgemini

Instructor Notes:

Additional notes for instructor

5.3. Attach and detach Events from DOM Elements

bind() and unbind() method



- bind() method is used to bind events dynamically
- .bind(eventType,handler(eventObject)) attaches a handler to an event for the selected element(s).
 - \$('#submitButton').bind('click', function() { //handle event });
- .click() is the same as .bind('click')
- .unbind(event) is used to remove handler previously bound to an element.
 - \$('#submitButton').unbind();
 - \$('#submitButton').unbind('click'); - unbind specific event
- bind() allows multiple events to be bound to one or more elements
 - \$('#targetDiv').bind('mouseenter mouseleave', function() { //handle event });

Capgemini 

Instructor Notes:

Additional notes for instructor

5.3. Attach and detach Events from DOM Elements

on() method

- on() method attaches an event handler function for one or more events to the selected elements
- on() method can bind events to elements that don't exist yet which is called as event delegation.
 - `$('#target').on('click', '.spl:button', sayHi);`
- Above statement will attach sayHi() event handler function to the click event on the button with class spl. The button may be yet to append in the div with id target.

Instructor Notes:

Additional notes for instructor

5.3. Attach and detach Events from DOM Elements

off() method

- off() method removes event handlers that were attached with on()
 - \$('#target').off('click', '.spl:button');
- one() method especially useful if you need to do some complicated setup the first time an element is clicked, but not subsequent times. It accepts the same arguments as .on()

Capgemini 



Instructor Notes:

Additional notes for instructor

5.4. Triggering Events

Triggering Events

- jQuery provides a way to trigger the event handlers bound to an element without any user interaction via the .trigger() method
 - `$("#btn1").trigger("click");`
- It will trigger the click event on element selected by id btn1

Capgemini

Trigger() function can be used to trigger the predefined as well as user defined events.

```
$('#btnSecond').bind('karthikClick',function(event){  
    alert('Second Button Custom Event Occured');  
});
```

...

```
$('#btnFirst').bind('click', function(){  
    $('#btnSecond').trigger('karthikClick');  
});
```

Instructor Notes:

Additional notes for instructor

5.5. Passing data through events

Passing data through events

- While attaching the event, data in JSON form can be passed to event handler function which can be accessed through event.data property
 - `$('#btnData').bind('click',{id:1},function(event){
alert(event.data.id);
});`

Capgemini 

Instructor Notes:

Additional notes for
instructor

Summary



- jQuery simplifies handling cross-browser event attachments
- `bind()` and `unbind()` provide a flexible way to work with different events
- `on()` allows future child elements to be wired to event handlers
- `one()` allows elements to be wired to event handlers for one time event handling
- `trigger()` allows to trigger the event

Capgemini 

Add the notes here.

Instructor Notes:

Additional notes for instructor

Lab



- 5.1. Adding and Removing Style at runtime using Click Event
- 5.2. Preventing Cut-Copy-Paste in textbox
- 5.3. Identifying mouse button
- 5.4. Limiting number of characters in textarea
- 5.5. Check/Uncheck all checkboxes at once
- 5.6. Event Triggering

Capgemini 

Add the notes here.

Instructor Notes:

Answers for the Review

Question 1: A

Question 2: A

Review Question



- What does the following statement do:
 - `$('#myDiv').bind('click',function(){// do something});`
- It attaches the event handler function to click event on element selected by `#myDiv`
- It triggers the click event on the element selected by `#myDiv`.
- What does this do: `$('#myDiv').trigger('click');`
- It simulates a click on the element and runs all the event handlers associated with it.
- It sets up a trigger. When a click occurs, the trigger is going to be activated.



Add the notes here.