Tech Mahindra

jQuery for ASP.NET MVC

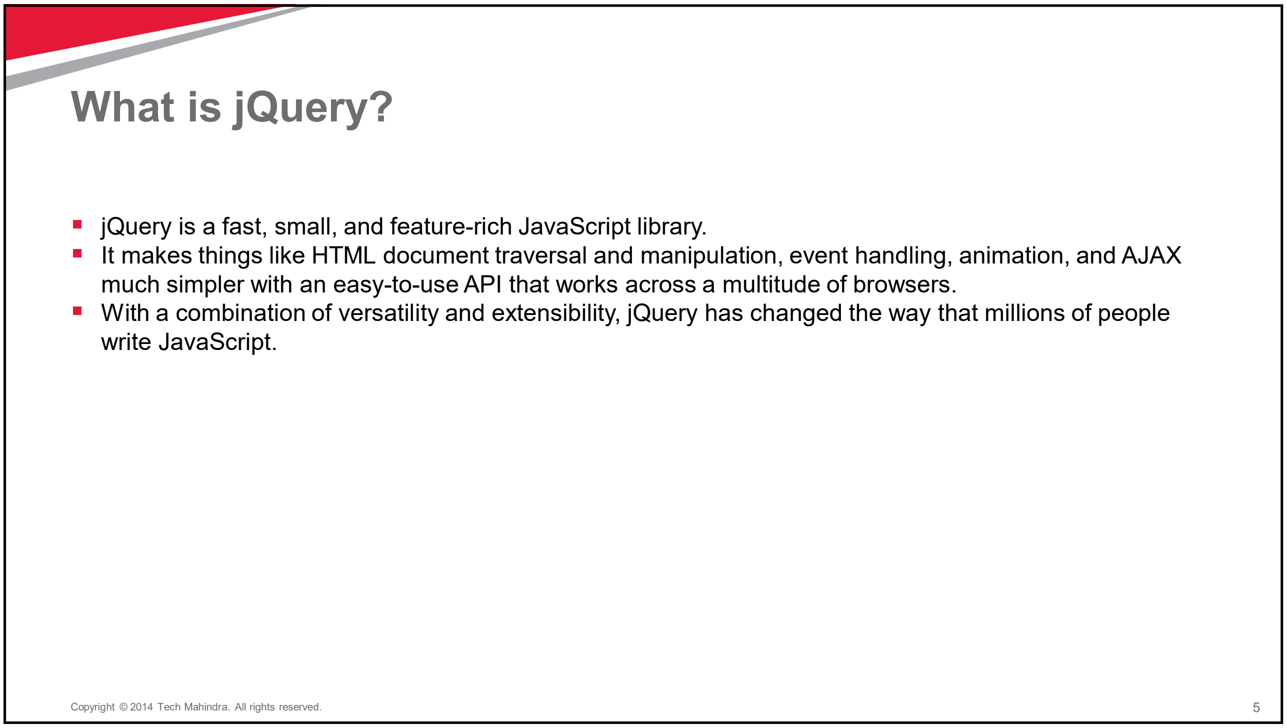
Agenda

- Brief introduction to jQuery
- jQuery Selectors
- Event Handling
- DOM Manipulation & Dynamic Content
- Traversal and other useful methods
- Effects and Animations
- AJAX – introduction
- Implementing AJAX with jQuery

Introduction - jQuery

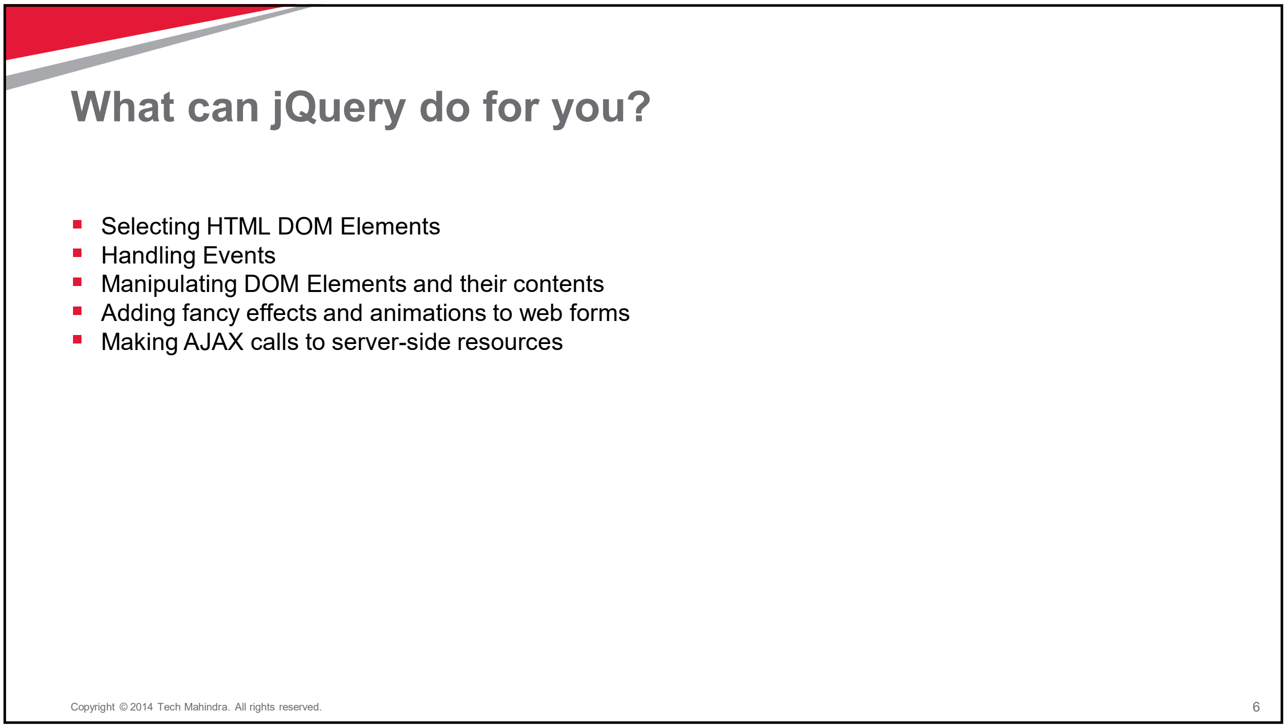
Agenda

- What jQuery is and what it can do for you
- Installing jQuery using various options
- Visual Studio features that make working with jQuery easier
- A simple program using ASP.NET MVC and jQuery



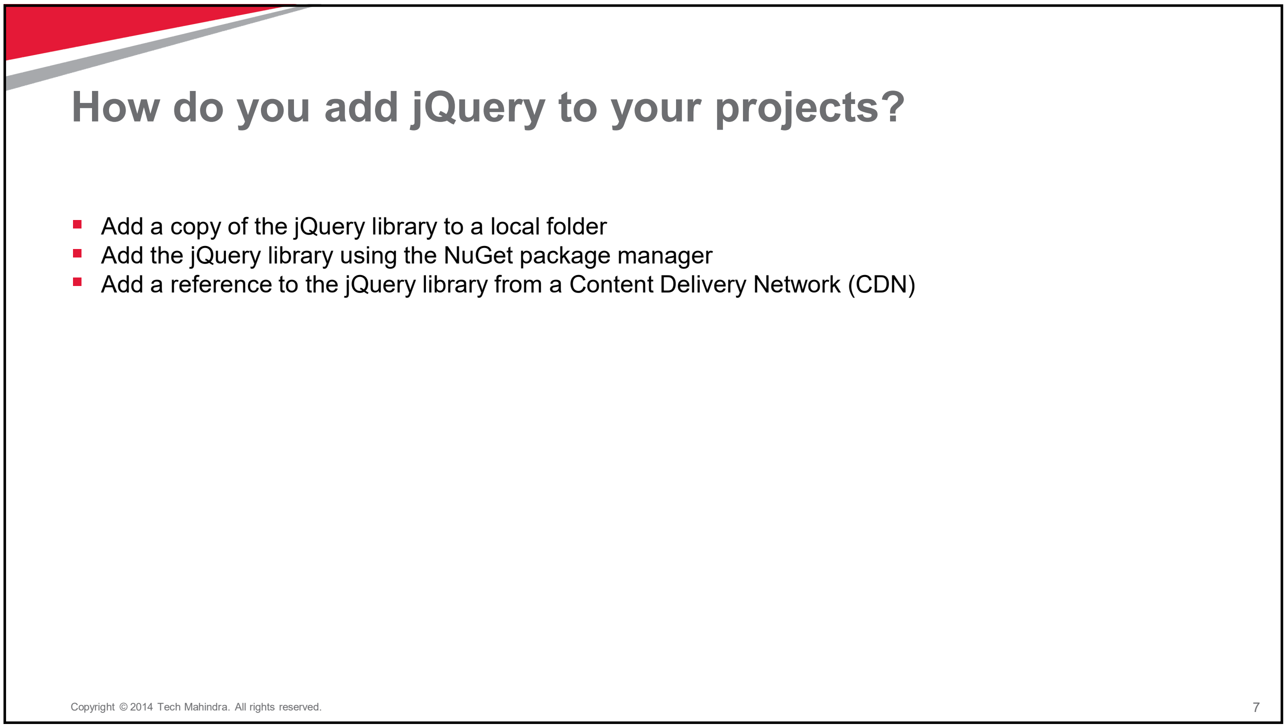
What is jQuery?

- jQuery is a fast, small, and feature-rich JavaScript library.
- It makes things like HTML document traversal and manipulation, event handling, animation, and AJAX much simpler with an easy-to-use API that works across a multitude of browsers.
- With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.



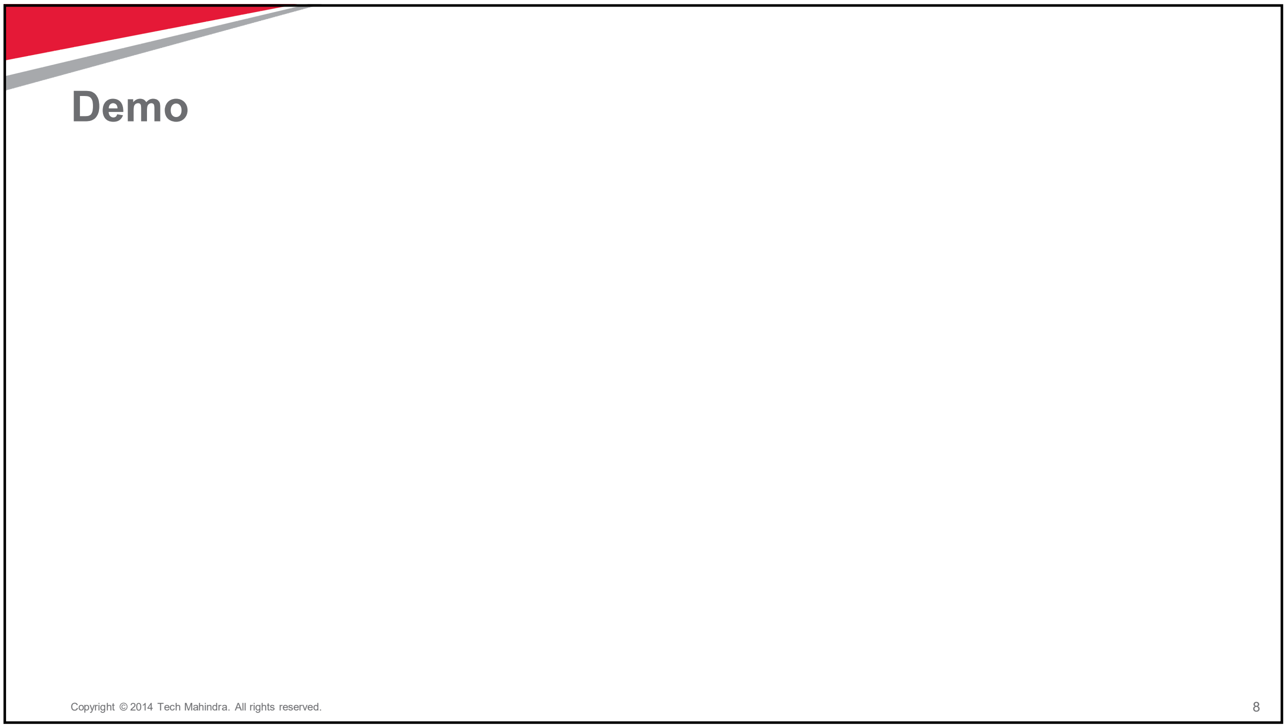
What can jQuery do for you?

- Selecting HTML DOM Elements
- Handling Events
- Manipulating DOM Elements and their contents
- Adding fancy effects and animations to web forms
- Making AJAX calls to server-side resources

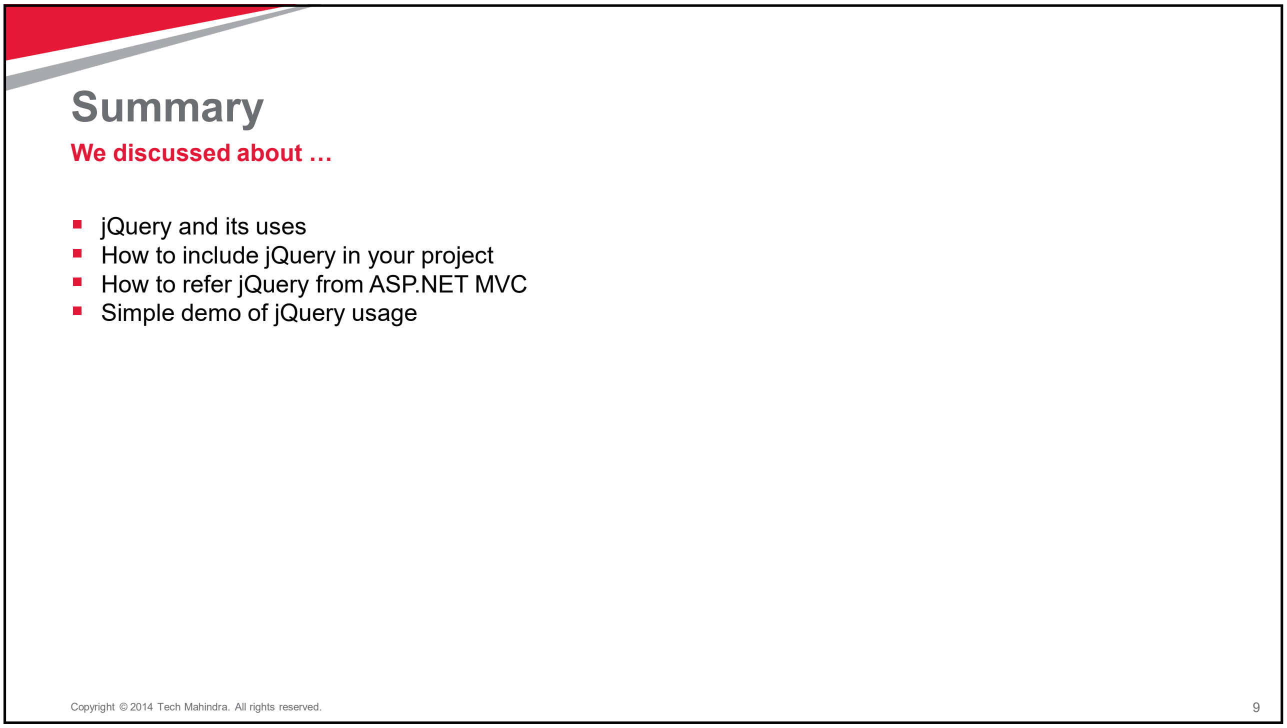


How do you add jQuery to your projects?

- Add a copy of the jQuery library to a local folder
- Add the jQuery library using the NuGet package manager
- Add a reference to the jQuery library from a Content Delivery Network (CDN)



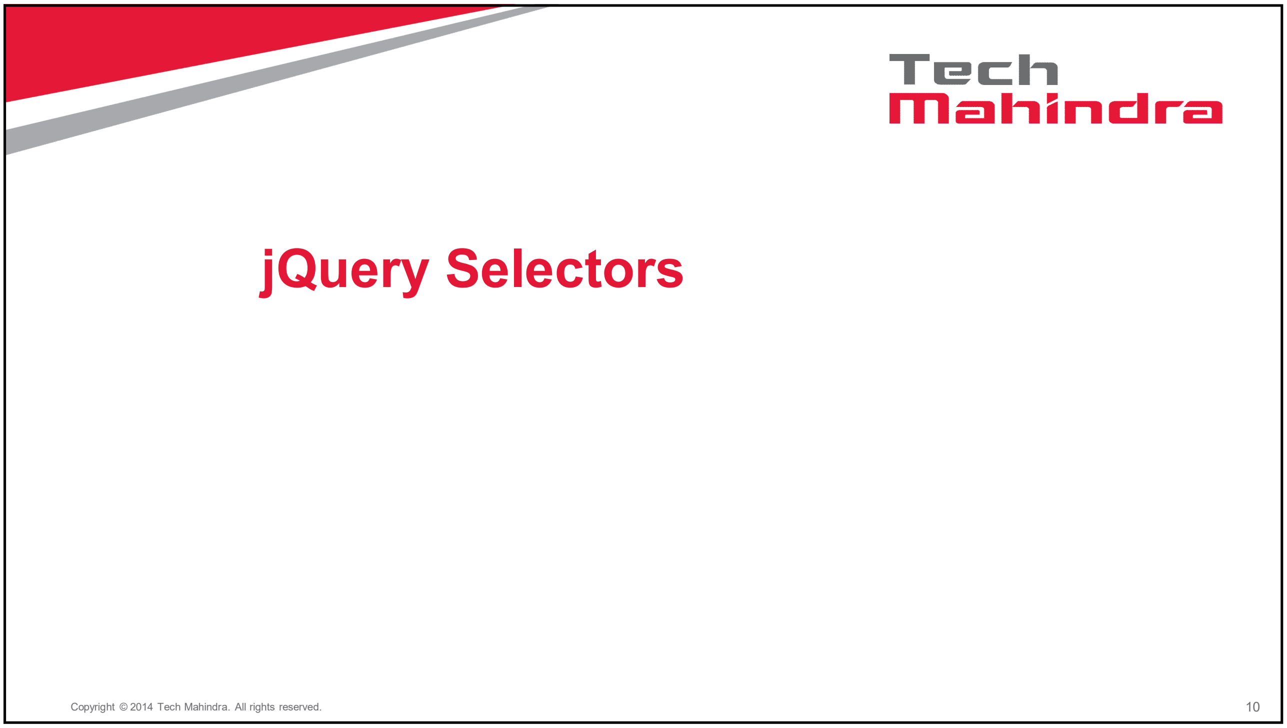
Demo



Summary

We discussed about …

- jQuery and its uses
- How to include jQuery in your project
- How to refer jQuery from ASP.NET MVC
- Simple demo of jQuery usage



Tech Mahindra

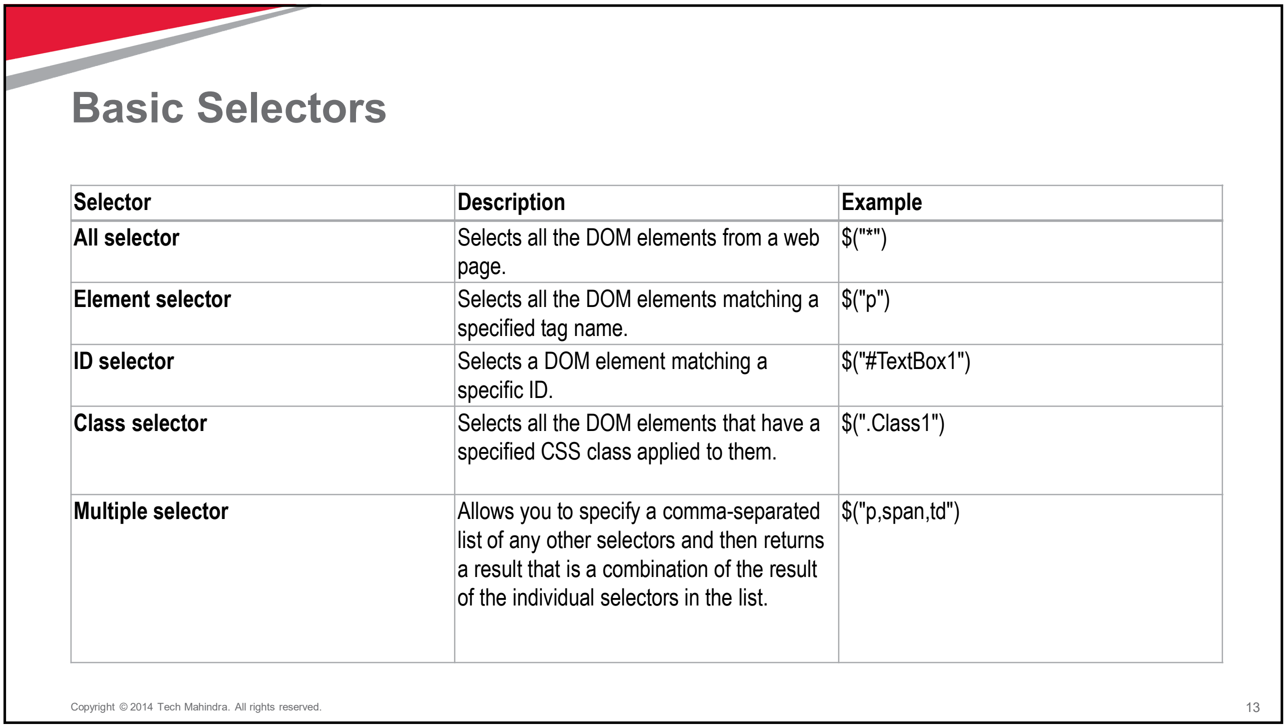
jQuery Selectors

Agenda

- What selectors are and how to use them in ASP.NET applications
- How to use basic jQuery selectors
- How to use advanced jQuery selectors

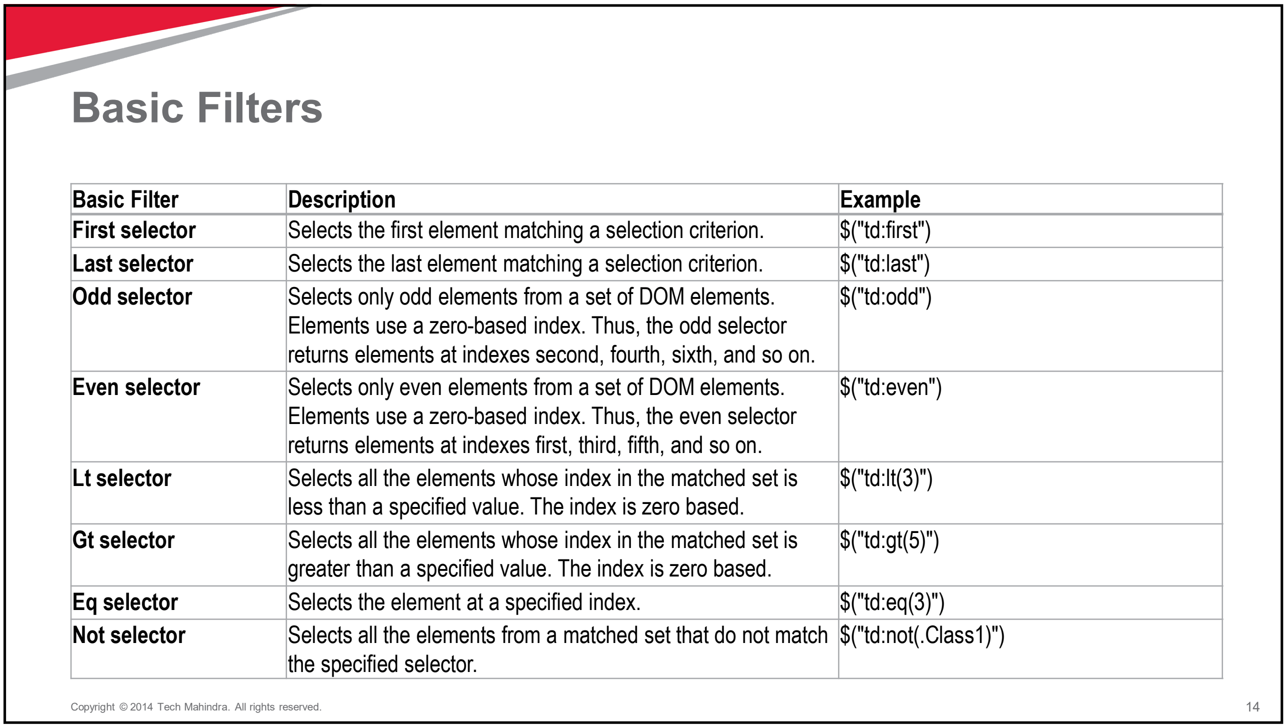
Overview of jQuery Selectors

- Basic Selector:
  - Based on certain basic conditions (ID, class, element tag name)
- Basic Filter:
  - By applying predefined filters .e.g., odd, even, >, < , = certain index, etc
- Attribute Selector:
  - Based on their attribute values; e.g., “equals to”, “starts with”, “ends with”, “contains”
- Child Filter
  - Based on position with respect to parent element e.g., first or last child, etc
- Content Filter
  - Based on specific text or a specific element it contains
- Form Filter
  - Filter elements from an HTML form, e.g., filter <input> elements that are of type Submit or Password
- Hierarchy Filter
  - Based on a hierarchy criterion/relationship e.g., parent-child, ancestor-dependent, or sibling
- Visibility Filter:
  - Based on visibility status (hidden or visible)



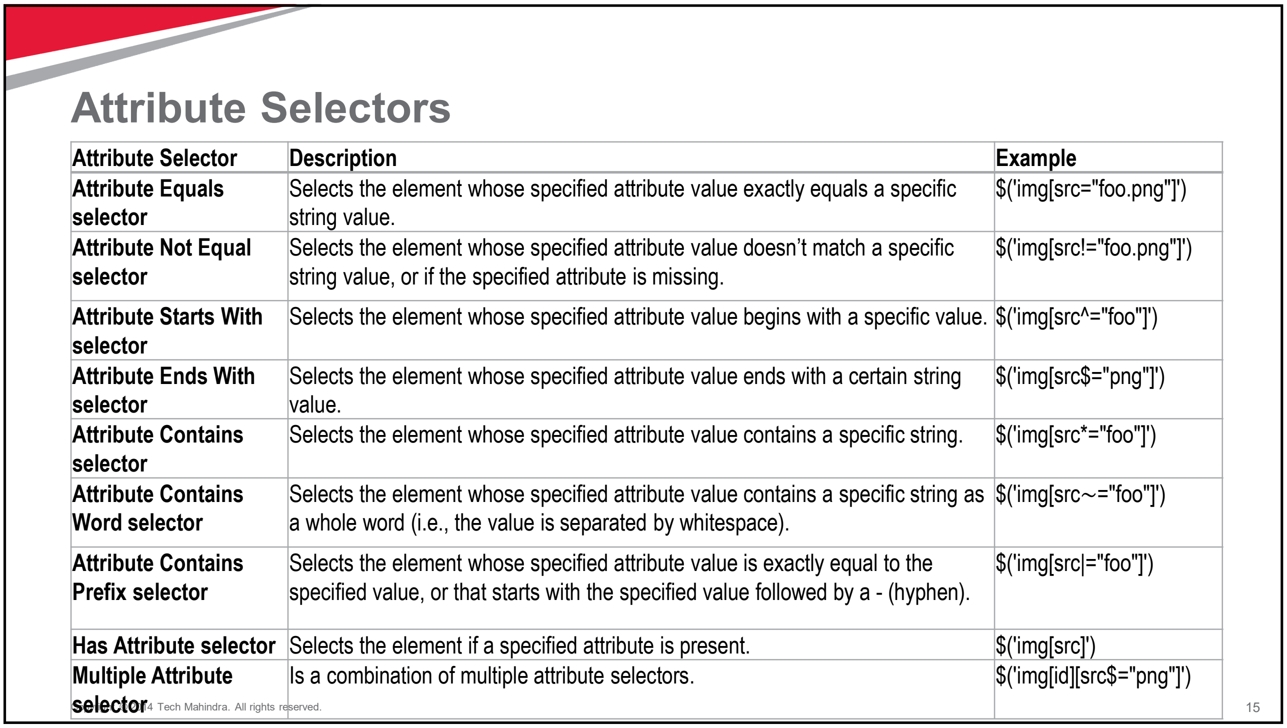
Basic Selectors

Selector	Description	Example
All selector	Selects all the DOM elements from a web page.	$("*")
Element selector	Selects all the DOM elements matching a specified tag name.	$("p")
ID selector	Selects a DOM element matching a specific ID.	$("#TextBox1")
Class selector	Selects all the DOM elements that have a specified CSS class applied to them.	$(".Class1")
Multiple selector	Allows you to specify a comma-separated list of any other selectors and then returns a result that is a combination of the result of the individual selectors in the list.	$("p,span,td")



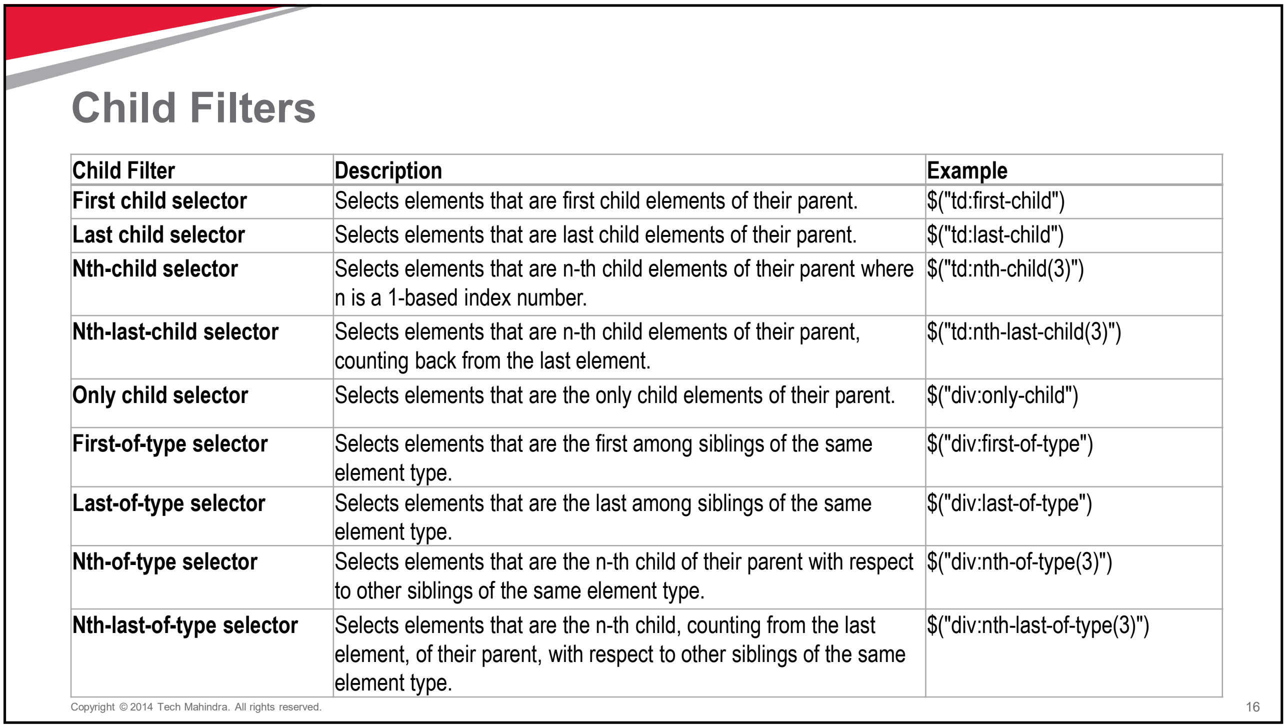
Basic Filters

Basic Filter	Description	Example
First selector	Selects the first element matching a selection criterion.	$("td:first")
Last selector	Selects the last element matching a selection criterion.	$("td:last")
Odd selector	Selects only odd elements from a set of DOM elements. Elements use a zero-based index. Thus, the odd selector returns elements at indexes second, fourth, sixth, and so on.	$("td:odd")
Even selector	Selects only even elements from a set of DOM elements. Elements use a zero-based index. Thus, the even selector returns elements at indexes first, third, fifth, and so on.	$("td:even")
Lt selector	Selects all the elements whose index in the matched set is less than a specified value. The index is zero based.	$("td:lt(3)")
Gt selector	Selects all the elements whose index in the matched set is greater than a specified value. The index is zero based.	$("td:gt(5)")
Eq selector	Selects the element at a specified index.	$("td:eq(3)")
Not selector	Selects all the elements from a matched set that do not match the specified selector.	$("td:not(.Class1)")



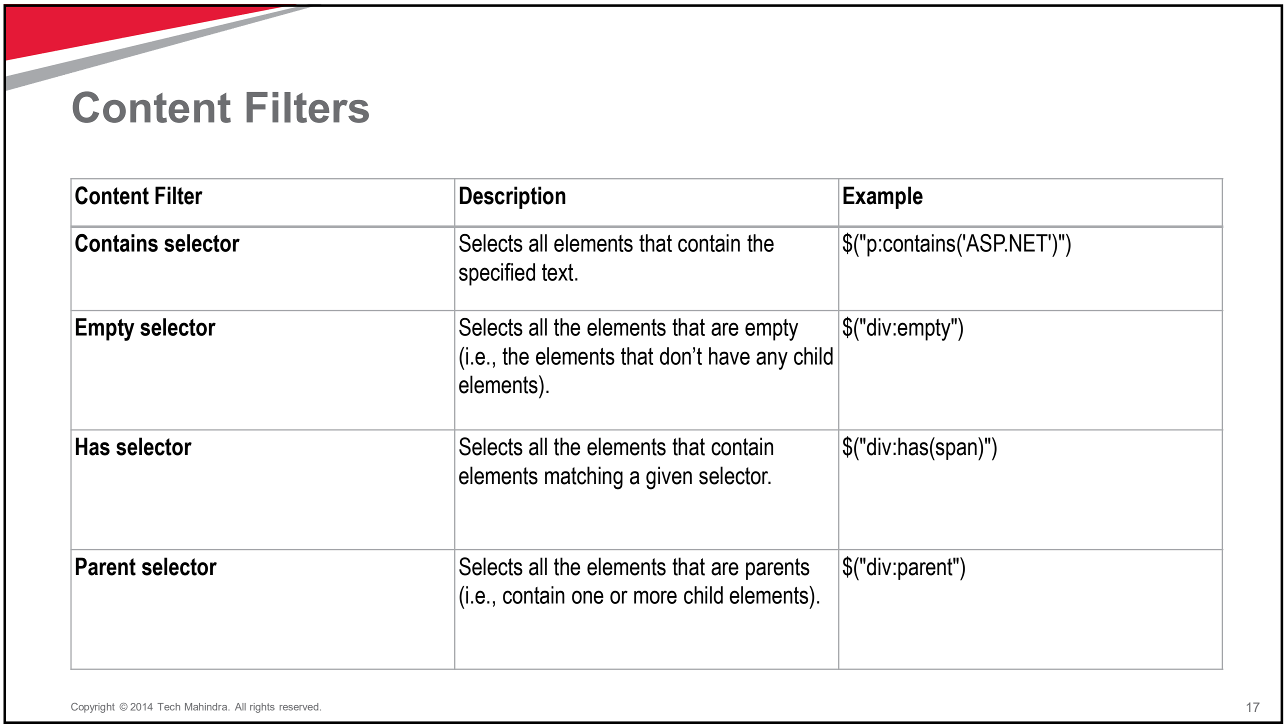
Attribute Selectors

Attribute Selector	Description	Example
Attribute Equals selector	Selects the element whose specified attribute value exactly equals a specific string value.	$('img[src="foo.png"]')
Attribute Not Equal selector	Selects the element whose specified attribute value doesn't match a specific string value, or if the specified attribute is missing.	$('img[src!="foo.png"]')
Attribute Starts With selector	Selects the element whose specified attribute value begins with a specific value.	$('img[src^="foo"]')
Attribute Ends With selector	Selects the element whose specified attribute value ends with a certain string value.	$('img[src$="png"]')
Attribute Contains selector	Selects the element whose specified attribute value contains a specific string.	$('img[src*="foo"]')
Attribute Contains Word selector	Selects the element whose specified attribute value contains a specific string as a whole word (i.e., the value is separated by whitespace).	$('img[src~="foo"]')
Attribute Contains Prefix selector	Selects the element whose specified attribute value is exactly equal to the specified value, or that starts with the specified value followed by a - (hyphen).	$('img[src|="foo"]')
Has Attribute selector	Selects the element if a specified attribute is present.	$('img[src]')
Multiple Attribute selector	Is a combination of multiple attribute selectors.	$('img[id][src$="png"]')



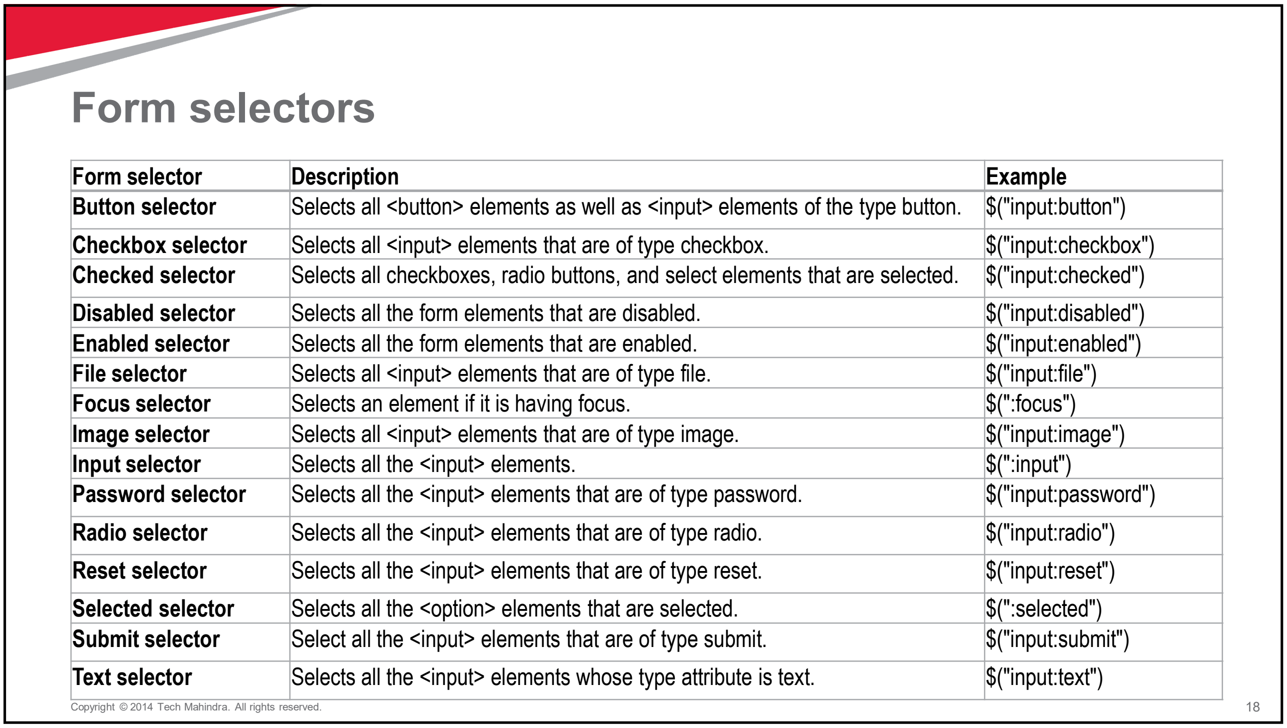
Child Filters

Child Filter	Description	Example
First child selector	Selects elements that are first child elements of their parent.	$("td:first-child")
Last child selector	Selects elements that are last child elements of their parent.	$("td:last-child")
Nth-child selector	Selects elements that are n-th child elements of their parent where n is a 1-based index number.	$("td:nth-child(3)")
Nth-last-child selector	Selects elements that are n-th child elements of their parent, counting back from the last element.	$("td:nth-last-child(3)")
Only child selector	Selects elements that are the only child elements of their parent.	$("div:only-child")
First-of-type selector	Selects elements that are the first among siblings of the same element type.	$("div:first-of-type")
Last-of-type selector	Selects elements that are the last among siblings of the same element type.	$("div:last-of-type")
Nth-of-type selector	Selects elements that are the n-th child of their parent with respect to other siblings of the same element type.	$("div:nth-of-type(3)")
Nth-last-of-type selector	Selects elements that are the n-th child, counting from the last element, of their parent, with respect to other siblings of the same element type.	$("div:nth-last-of-type(3)")



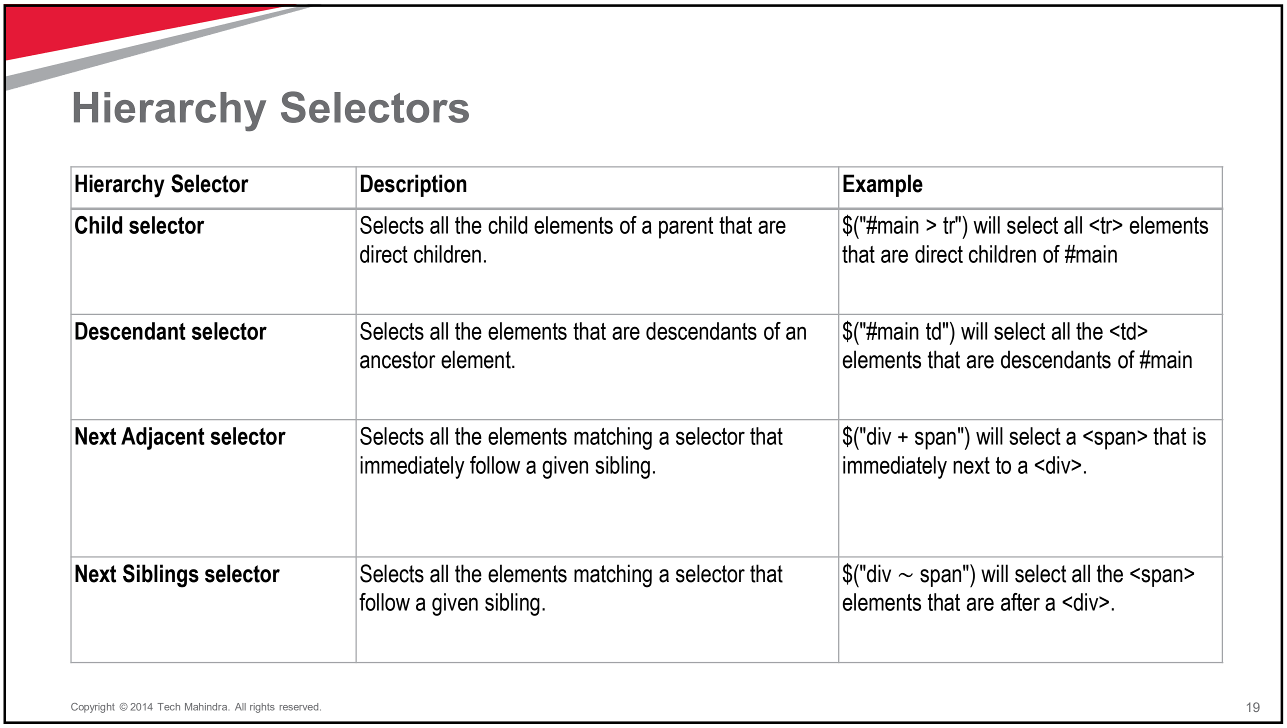
Content Filters

Content Filter	Description	Example
Contains selector	Selects all elements that contain the specified text.	$("p:contains('ASP.NET')")
Empty selector	Selects all the elements that are empty (i.e., the elements that don't have any child elements).	$("div:empty")
Has selector	Selects all the elements that contain elements matching a given selector.	$("div:has(span)")
Parent selector	Selects all the elements that are parents (i.e., contain one or more child elements).	$("div:parent")



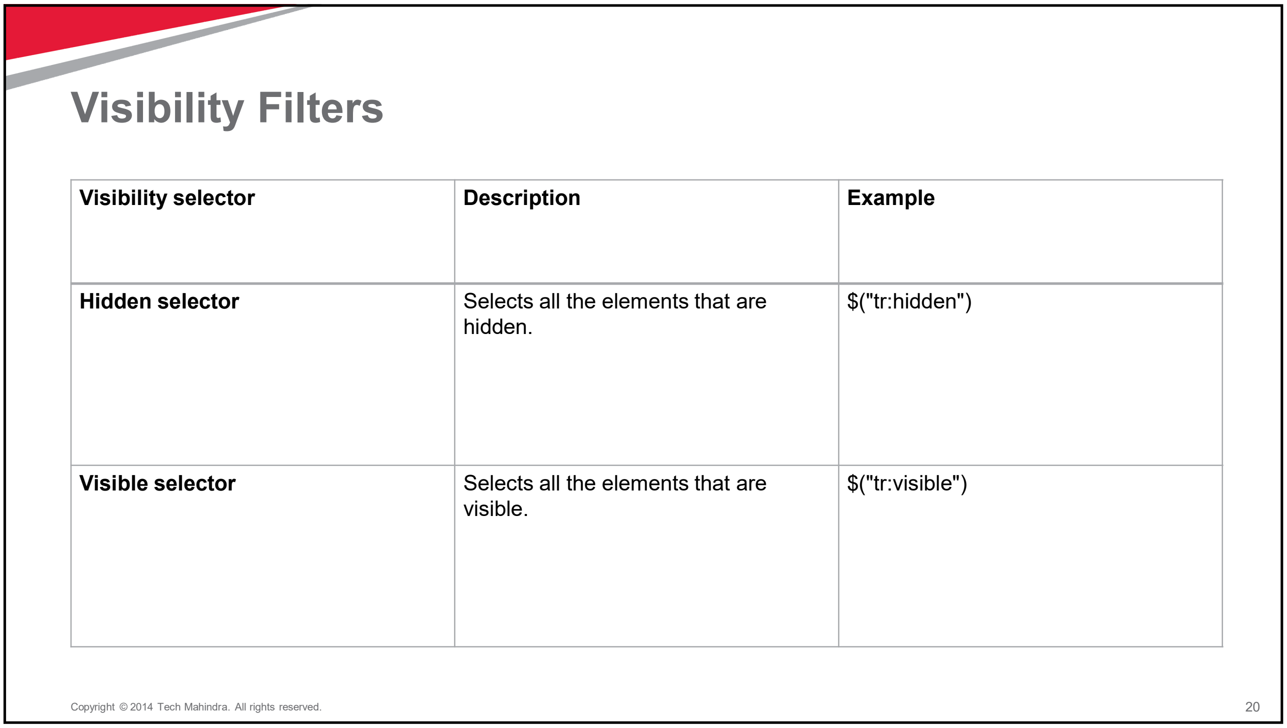
Form selectors

Form selector	Description	Example
Button selector	Selects all <button> elements as well as <input> elements of the type button.	$("input:button")
Checkbox selector	Selects all <input> elements that are of type checkbox.	$("input:checkbox")
Checked selector	Selects all checkboxes, radio buttons, and select elements that are selected.	$("input:checked")
Disabled selector	Selects all the form elements that are disabled.	$("input:disabled")
Enabled selector	Selects all the form elements that are enabled.	$("input:enabled")
File selector	Selects all <input> elements that are of type file.	$("input:file")
Focus selector	Selects an element if it is having focus.	$(":focus")
Image selector	Selects all <input> elements that are of type image.	$("input:image")
Input selector	Selects all the <input> elements.	$(":input")
Password selector	Selects all the <input> elements that are of type password.	$("input:password")
Radio selector	Selects all the <input> elements that are of type radio.	$("input:radio")
Reset selector	Selects all the <input> elements that are of type reset.	$("input:reset")
Selected selector	Selects all the <option> elements that are selected.	$(":selected")
Submit selector	Select all the <input> elements that are of type submit.	$("input:submit")
Text selector	Selects all the <input> elements whose type attribute is text.	$("input:text")



Hierarchy Selectors

Hierarchy Selector	Description	Example
Child selector	Selects all the child elements of a parent that are direct children.	$("#main > tr") will select all <tr> elements that are direct children of #main
Descendant selector	Selects all the elements that are descendants of an ancestor element.	$("#main td") will select all the <td> elements that are descendants of #main
Next Adjacent selector	Selects all the elements matching a selector that immediately follow a given sibling.	$("div + span") will select a <span> that is immediately next to a <div>.
Next Siblings selector	Selects all the elements matching a selector that follow a given sibling.	$("div ~ span") will select all the <span> elements that are after a <div>.



Visibility Filters

Visibility selector	Description	Example
Hidden selector	Selects all the elements that are hidden.	$("tr:hidden")
Visible selector	Selects all the elements that are visible.	$("tr:visible")

Demo

Demos

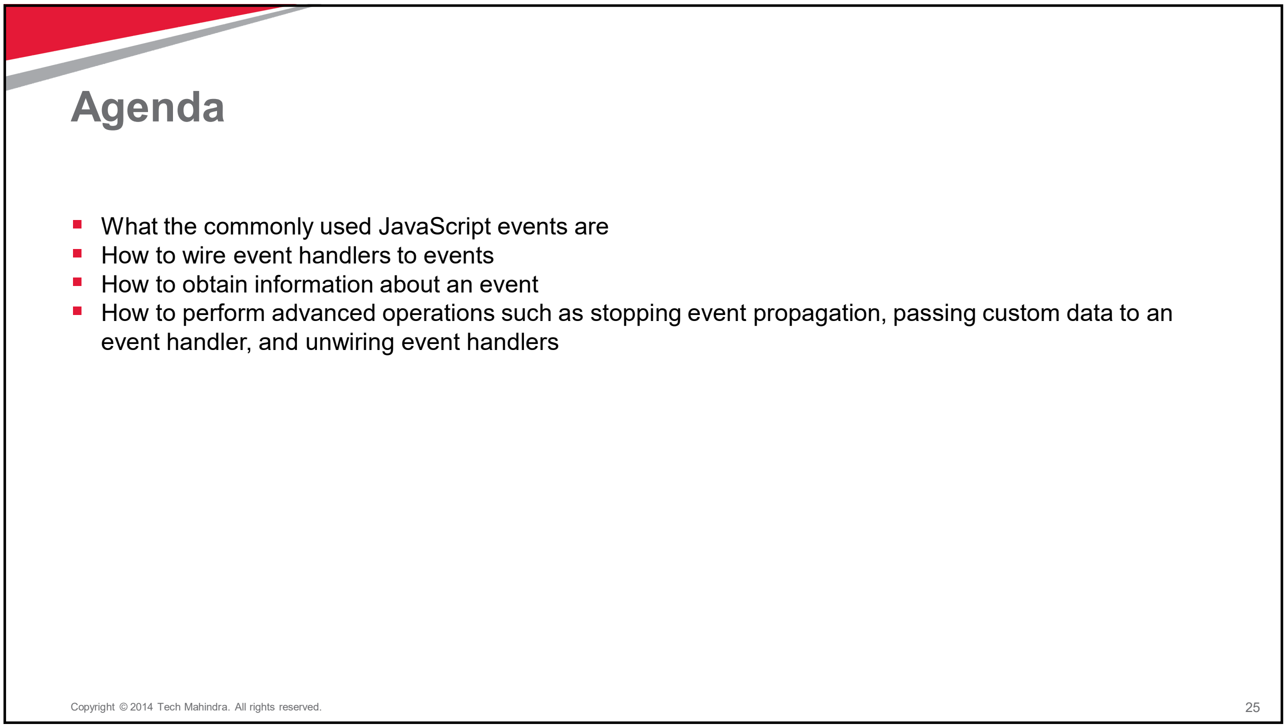
- Basic Selector: http://api.jquery.com/category/selectors/basic-css-selectors/
  - Based on certain basic conditions (ID, class, element tag name)
- Basic Filter: http://api.jquery.com/category/selectors/basic-filter-selectors/
  - By applying predefined filters .e.g., odd, even, >, < , = certain index, etc.
- Attribute Selector: http://api.jquery.com/category/selectors/attribute-selectors/
  - Based on their attribute values; e.g., “equals to”, “starts with”, “ends with”, “contains”
- Child Filter: http://api.jquery.com/category/selectors/child-filter-selectors/
  - Based on position with respect to parent element e.g., first or last child, etc.
- Content Filter: http://api.jquery.com/category/selectors/content-filter-selector/
  - Based on specific text or a specific element it contains
- Form Filter: http://api.jquery.com/category/selectors/form-selectors/
  - Filter elements from an HTML form, e.g., filter <input> elements that are of type Submit or Password
- Hierarchy Filter: http://api.jquery.com/category/selectors/hierarchy-selectors/
  - Based on a hierarchy criterion/relationship e.g., parent-child, ancestor-dependent, or sibling
- Visibility Filter: http://api.jquery.com/category/selectors/visibility-filter-selectors/
  - Based on visibility status (hidden or visible)

Summary

- We discussed:
  - JQuery borrows from CSS 1–3 and offers selectors for matching a set of elements in a document
  - Eight types of selectors

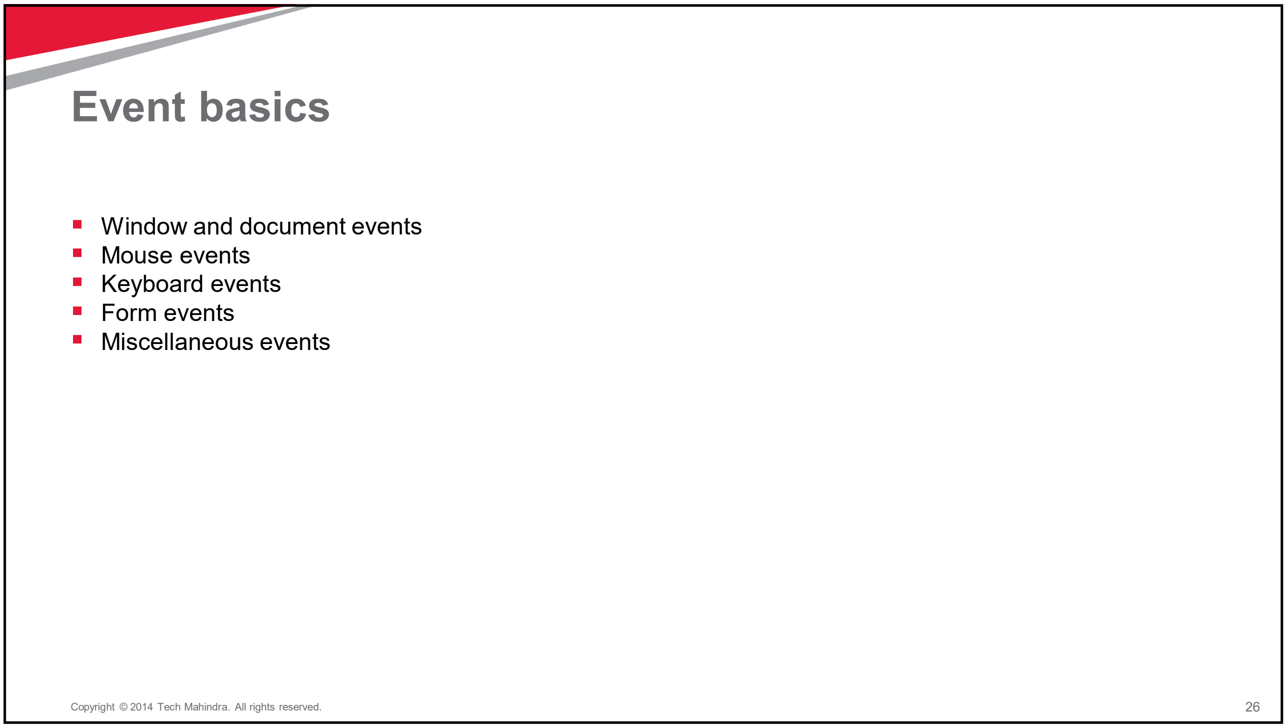
Tech Mahindra

Event Handling



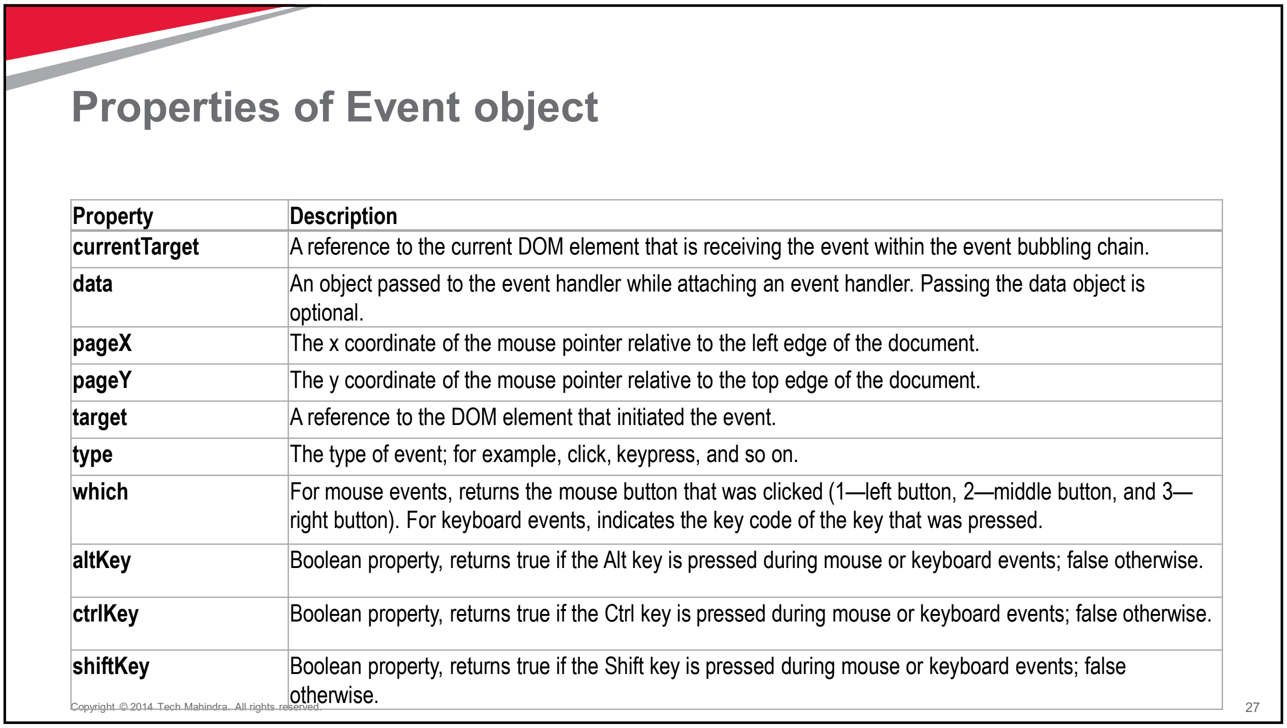
Agenda

- What the commonly used JavaScript events are
- How to wire event handlers to events
- How to obtain information about an event
- How to perform advanced operations such as stopping event propagation, passing custom data to an event handler, and unwiring event handlers



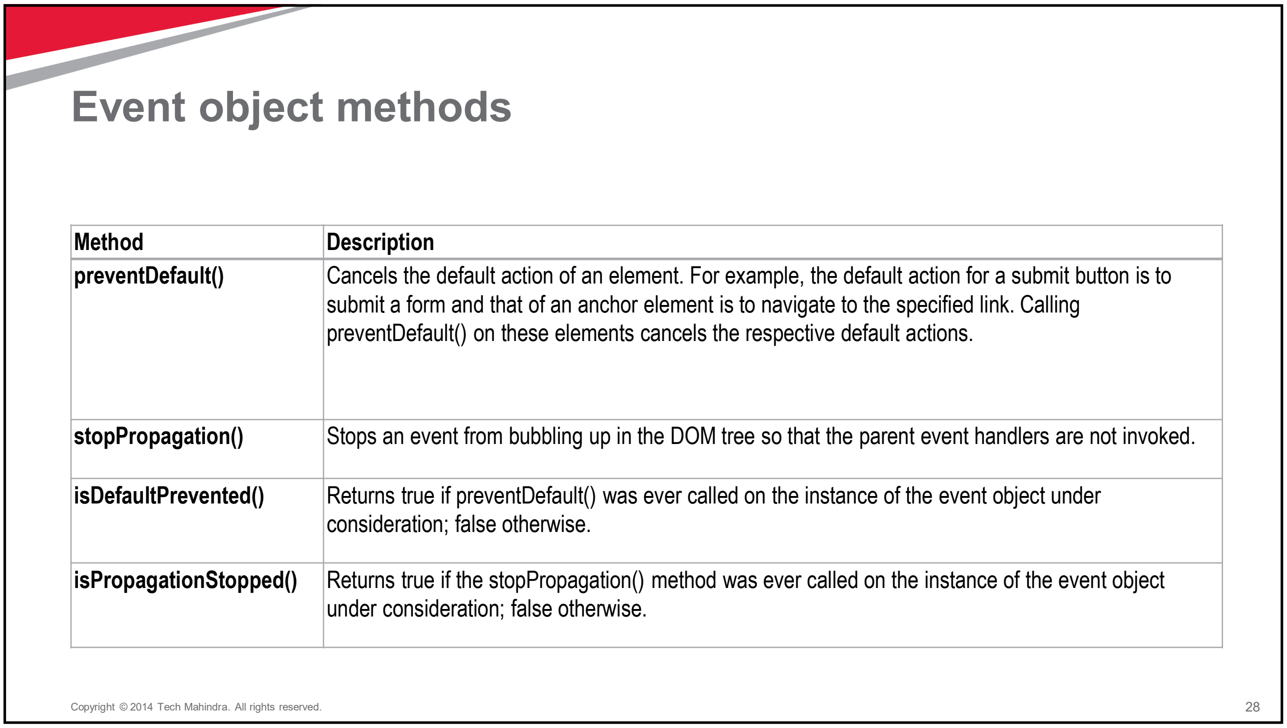
Event basics

- Window and document events
- Mouse events
- Keyboard events
- Form events
- Miscellaneous events



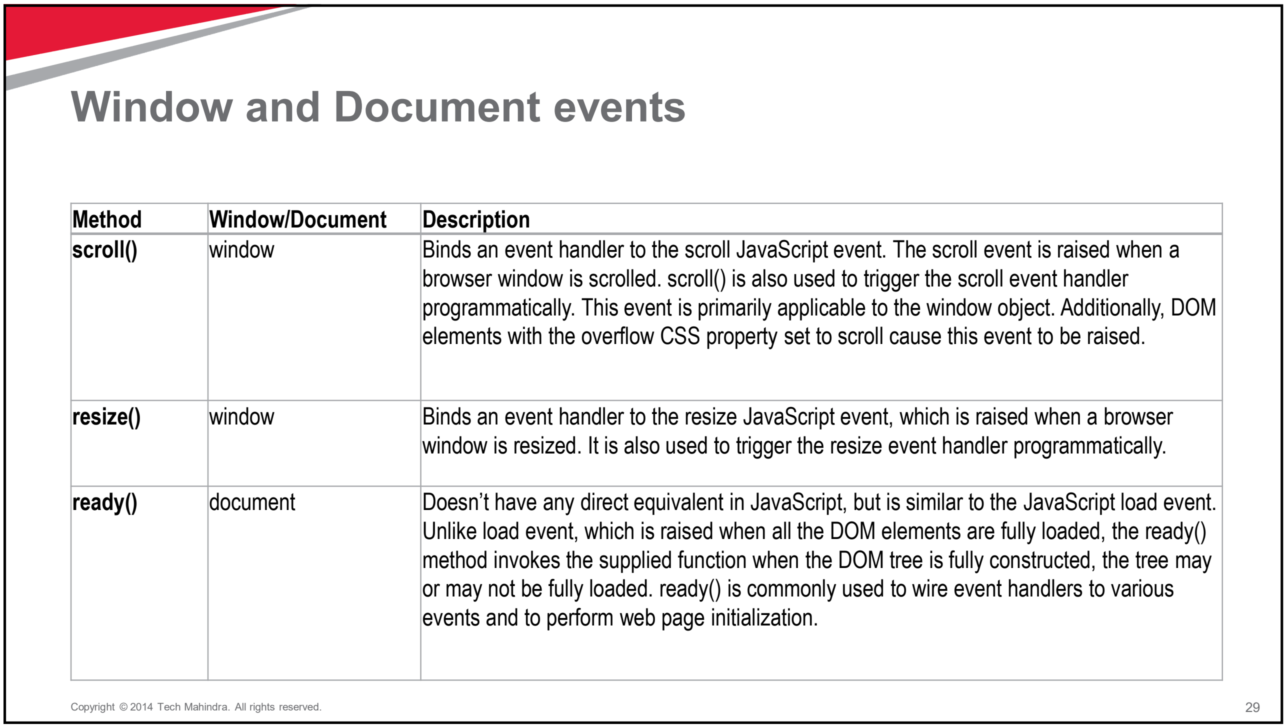
Properties of Event object

Property	Description
currentTarget	A reference to the current DOM element that is receiving the event within the event bubbling chain.
data	An object passed to the event handler while attaching an event handler. Passing the data object is optional.
pageX	The x coordinate of the mouse pointer relative to the left edge of the document.
pageY	The y coordinate of the mouse pointer relative to the top edge of the document.
target	A reference to the DOM element that initiated the event.
type	The type of event; for example, click, keypress, and so on.
which	For mouse events, returns the mouse button that was clicked (1—left button, 2—middle button, and 3—right button). For keyboard events, indicates the key code of the key that was pressed.
altKey	Boolean property, returns true if the Alt key is pressed during mouse or keyboard events; false otherwise.
ctrlKey	Boolean property, returns true if the Ctrl key is pressed during mouse or keyboard events; false otherwise.
shiftKey	Boolean property, returns true if the Shift key is pressed during mouse or keyboard events; false otherwise.



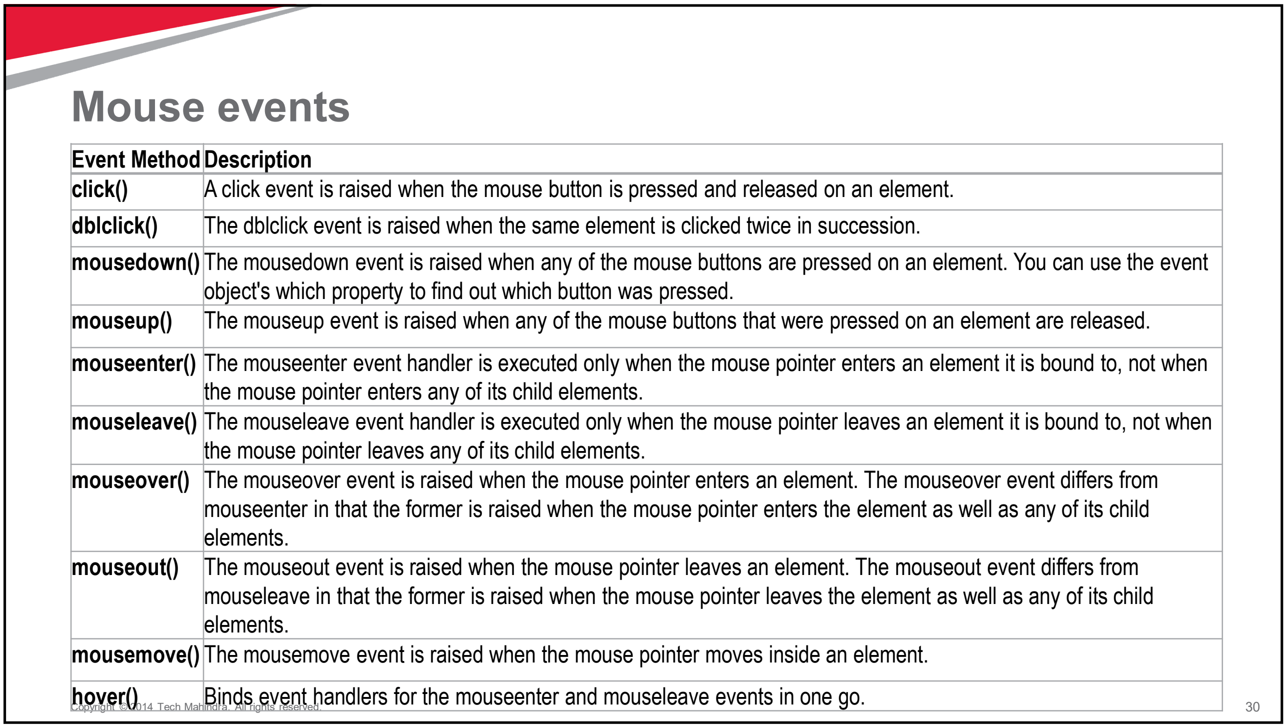
Event object methods

Method	Description
preventDefault()	Cancels the default action of an element. For example, the default action for a submit button is to submit a form and that of an anchor element is to navigate to the specified link. Calling preventDefault() on these elements cancels the respective default actions.
stopPropagation()	Stops an event from bubbling up in the DOM tree so that the parent event handlers are not invoked.
isDefaultPrevented()	Returns true if preventDefault() was ever called on the instance of the event object under consideration; false otherwise.
isPropagationStopped()	Returns true if the stopPropagation() method was ever called on the instance of the event object under consideration; false otherwise.



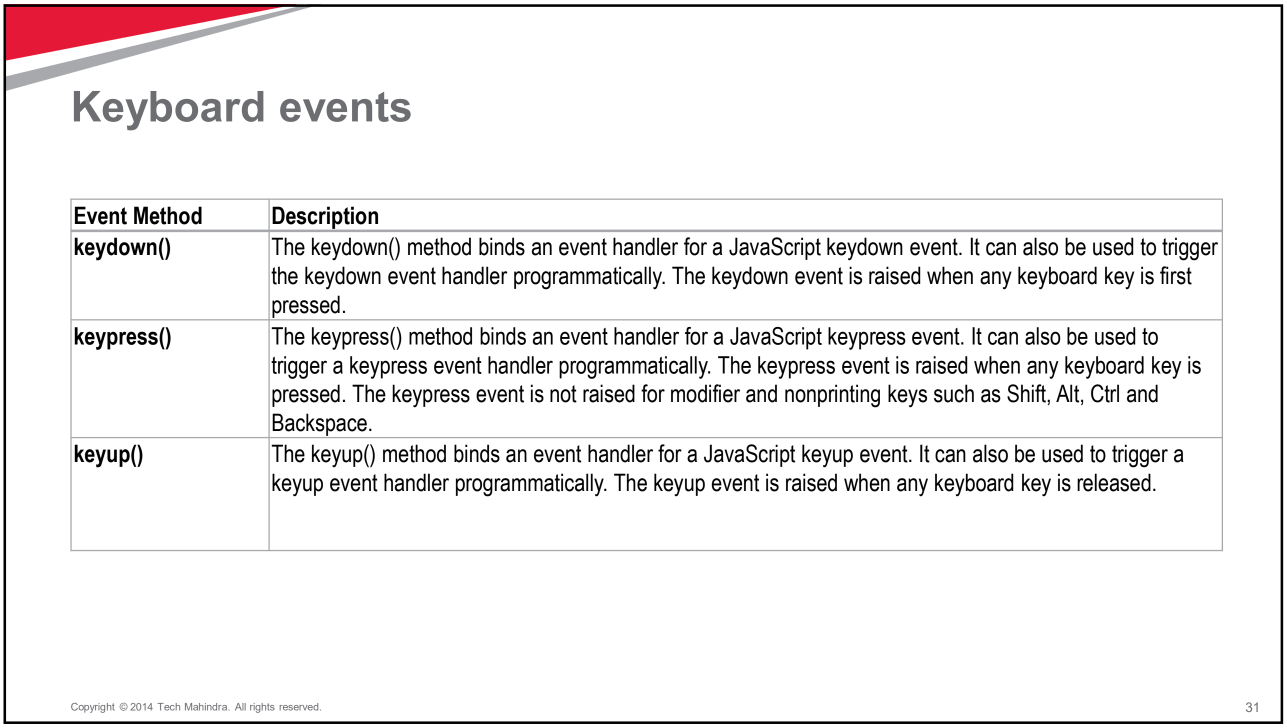
Window and Document events

Method	Window/Document	Description
scroll()	window	Binds an event handler to the scroll JavaScript event. The scroll event is raised when a browser window is scrolled. scroll() is also used to trigger the scroll event handler programmatically. This event is primarily applicable to the window object. Additionally, DOM elements with the overflow CSS property set to scroll cause this event to be raised.
resize()	window	Binds an event handler to the resize JavaScript event, which is raised when a browser window is resized. It is also used to trigger the resize event handler programmatically.
ready()	document	Doesn’t have any direct equivalent in JavaScript, but is similar to the JavaScript load event. Unlike load event, which is raised when all the DOM elements are fully loaded, the ready() method invokes the supplied function when the DOM tree is fully constructed, the tree may or may not be fully loaded. ready() is commonly used to wire event handlers to various events and to perform web page initialization.



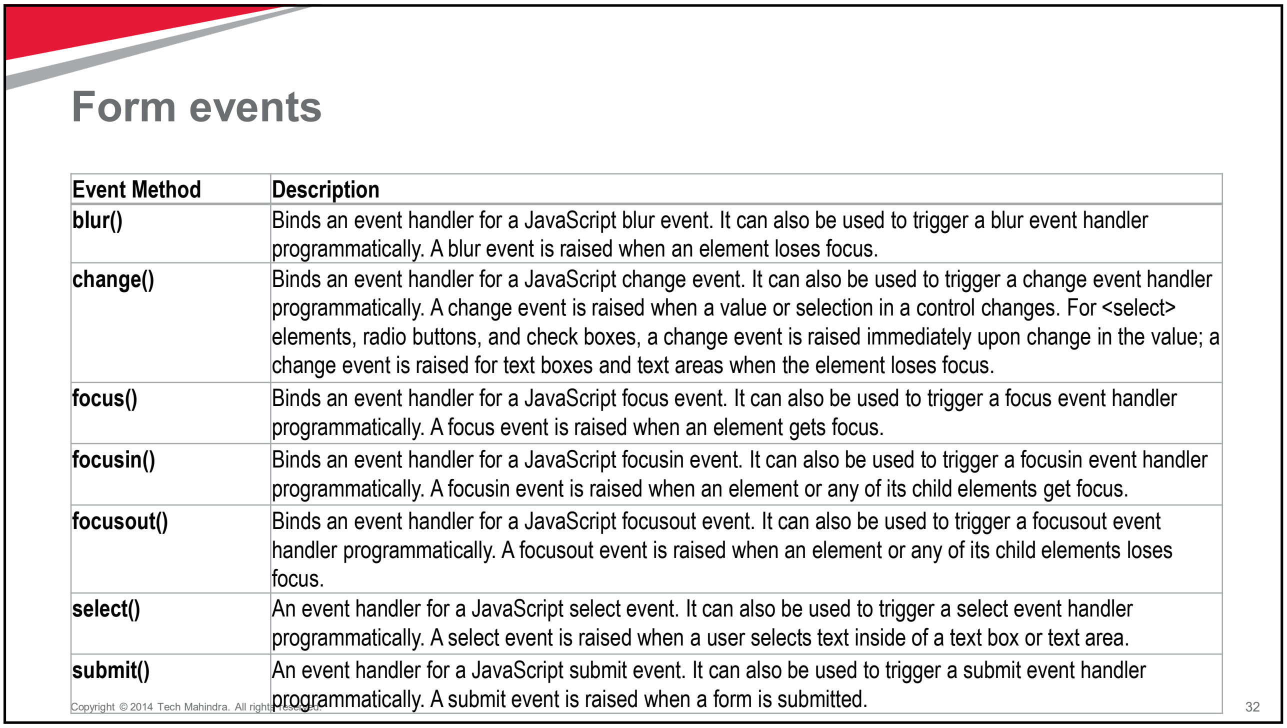
Mouse events

Event Method	Description
click()	A click event is raised when the mouse button is pressed and released on an element.
dblclick()	The dblclick event is raised when the same element is clicked twice in succession.
mousedown()	The mousedown event is raised when any of the mouse buttons are pressed on an element. You can use the event object's which property to find out which button was pressed.
mouseup()	The mouseup event is raised when any of the mouse buttons that were pressed on an element are released.
mouseenter()	The mouseenter event handler is executed only when the mouse pointer enters an element it is bound to, not when the mouse pointer enters any of its child elements.
mouseleave()	The mouseleave event handler is executed only when the mouse pointer leaves an element it is bound to, not when the mouse pointer leaves any of its child elements.
mouseover()	The mouseover event is raised when the mouse pointer enters an element. The mouseover event differs from mouseenter in that the former is raised when the mouse pointer enters the element as well as any of its child elements.
mouseout()	The mouseout event is raised when the mouse pointer leaves an element. The mouseout event differs from mouseleave in that the former is raised when the mouse pointer leaves the element as well as any of its child elements.
mousemove()	The mousemove event is raised when the mouse pointer moves inside an element.
hover()	Binds event handlers for the mouseenter and mouseleave events in one go.



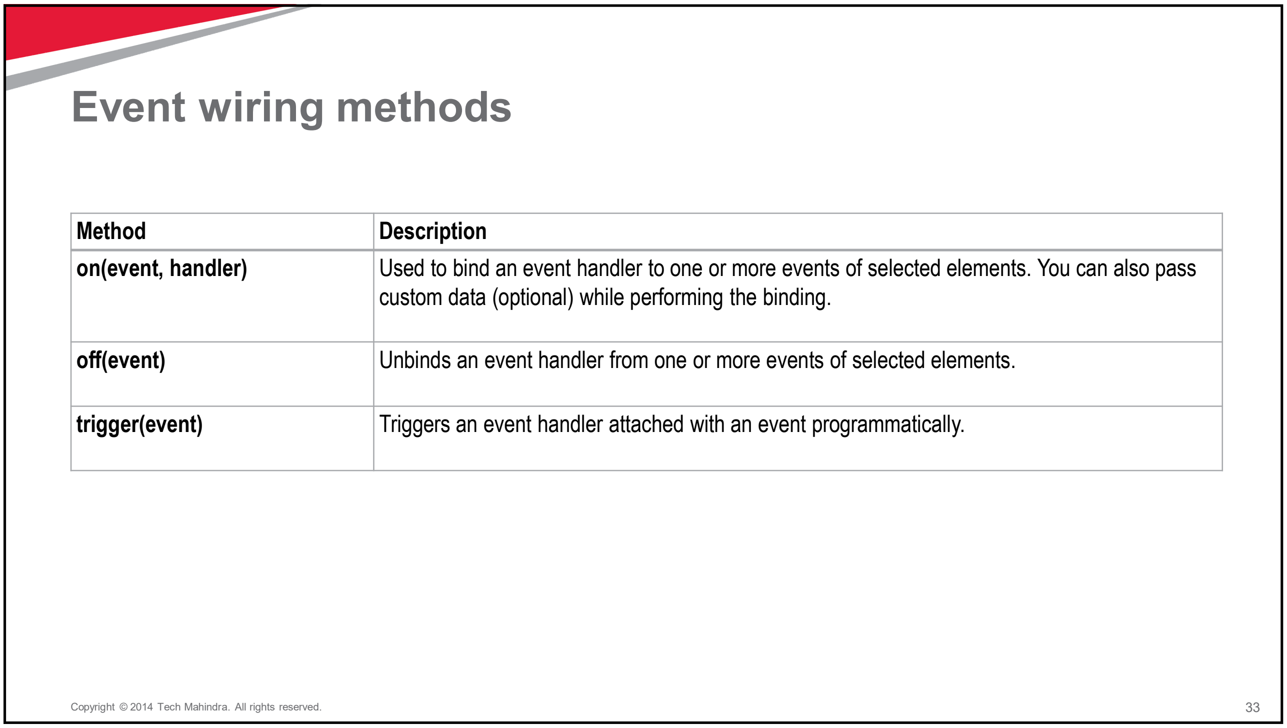
Keyboard events

Event Method	Description
keydown()	The keydown() method binds an event handler for a JavaScript keydown event. It can also be used to trigger the keydown event handler programmatically. The keydown event is raised when any keyboard key is first pressed.
keypress()	The keypress() method binds an event handler for a JavaScript keypress event. It can also be used to trigger a keypress event handler programmatically. The keypress event is raised when any keyboard key is pressed. The keypress event is not raised for modifier and nonprinting keys such as Shift, Alt, Ctrl and Backspace.
keyup()	The keyup() method binds an event handler for a JavaScript keyup event. It can also be used to trigger a keyup event handler programmatically. The keyup event is raised when any keyboard key is released.



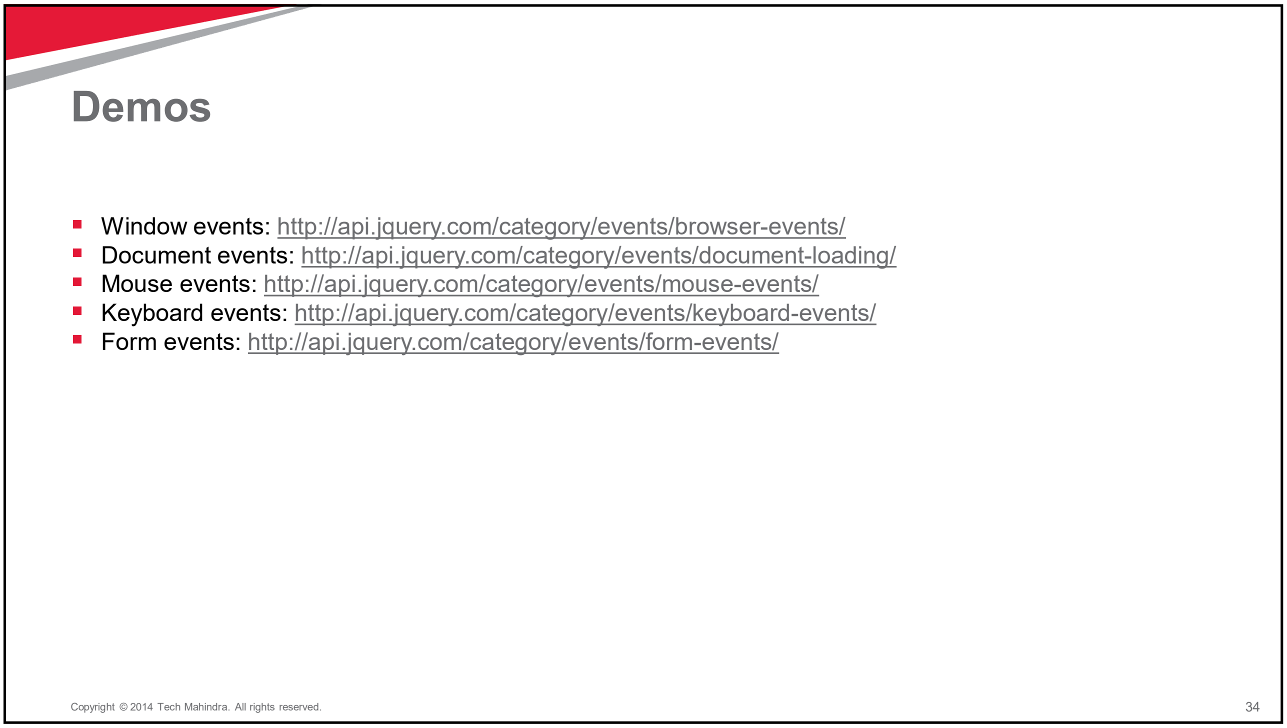
Form events

Event Method	Description
blur()	Binds an event handler for a JavaScript blur event. It can also be used to trigger a blur event handler programmatically. A blur event is raised when an element loses focus.
change()	Binds an event handler for a JavaScript change event. It can also be used to trigger a change event handler programmatically. A change event is raised when a value or selection in a control changes. For <select> elements, radio buttons, and check boxes, a change event is raised immediately upon change in the value; a change event is raised for text boxes and text areas when the element loses focus.
focus()	Binds an event handler for a JavaScript focus event. It can also be used to trigger a focus event handler programmatically. A focus event is raised when an element gets focus.
focusin()	Binds an event handler for a JavaScript focusin event. It can also be used to trigger a focusin event handler programmatically. A focusin event is raised when an element or any of its child elements get focus.
focusout()	Binds an event handler for a JavaScript focusout event. It can also be used to trigger a focusout event handler programmatically. A focusout event is raised when an element or any of its child elements loses focus.
select()	An event handler for a JavaScript select event. It can also be used to trigger a select event handler programmatically. A select event is raised when a user selects text inside of a text box or text area.
submit()	An event handler for a JavaScript submit event. It can also be used to trigger a submit event handler programmatically. A submit event is raised when a form is submitted.



Event wiring methods

Method	Description
on(event, handler)	Used to bind an event handler to one or more events of selected elements. You can also pass custom data (optional) while performing the binding.
off(event)	Unbinds an event handler from one or more events of selected elements.
trigger(event)	Triggers an event handler attached with an event programmatically.

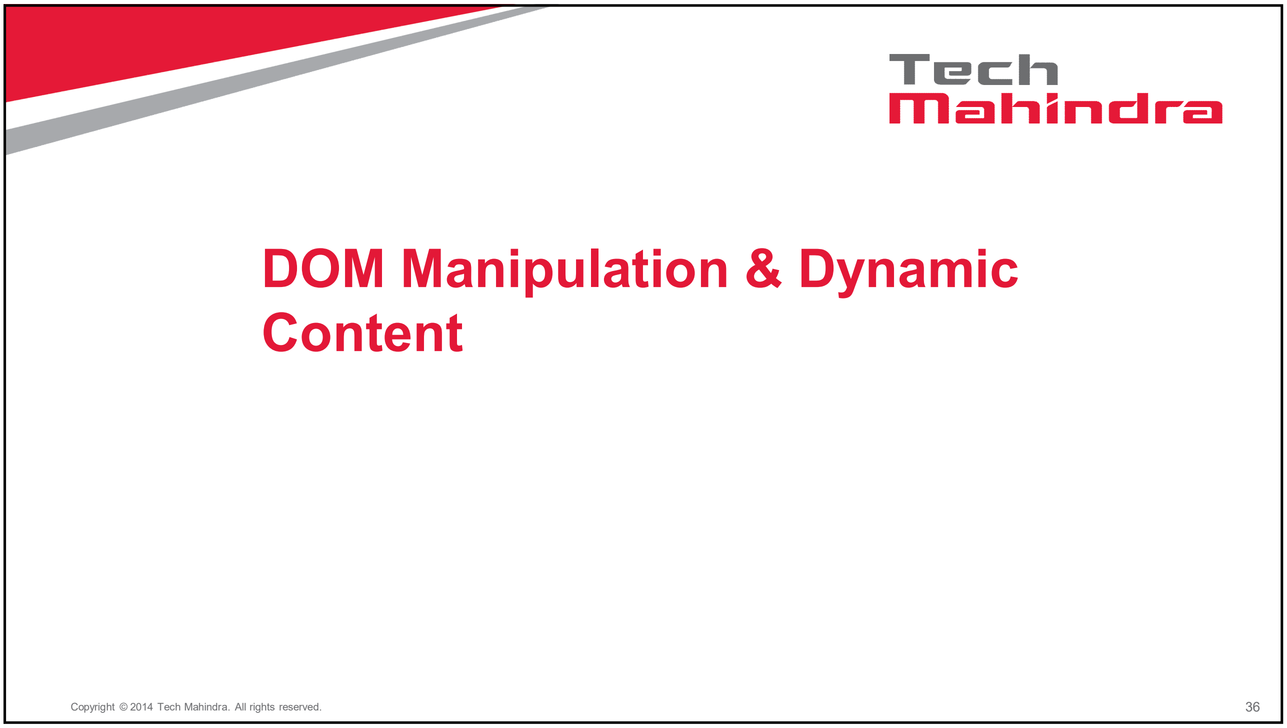


Demos

- Window events: http://api.jquery.com/category/events/browser-events/
- Document events: http://api.jquery.com/category/events/document-loading/
- Mouse events: http://api.jquery.com/category/events/mouse-events/
- Keyboard events: http://api.jquery.com/category/events/keyboard-events/
- Form events: http://api.jquery.com/category/events/form-events/

Summary

- Window and document events
- Mouse events
- Keyboard events
- Form events
- Event object properties
- Event object methods



Tech Mahindra

DOM Manipulation & Dynamic Content

Agenda

- How to work with CSS classes and CSS style properties
- How to manipulate the content of a DOM element
- How to add, remove, and replace DOM elements in a web page

Methods to inspect and manipulate CSS classes

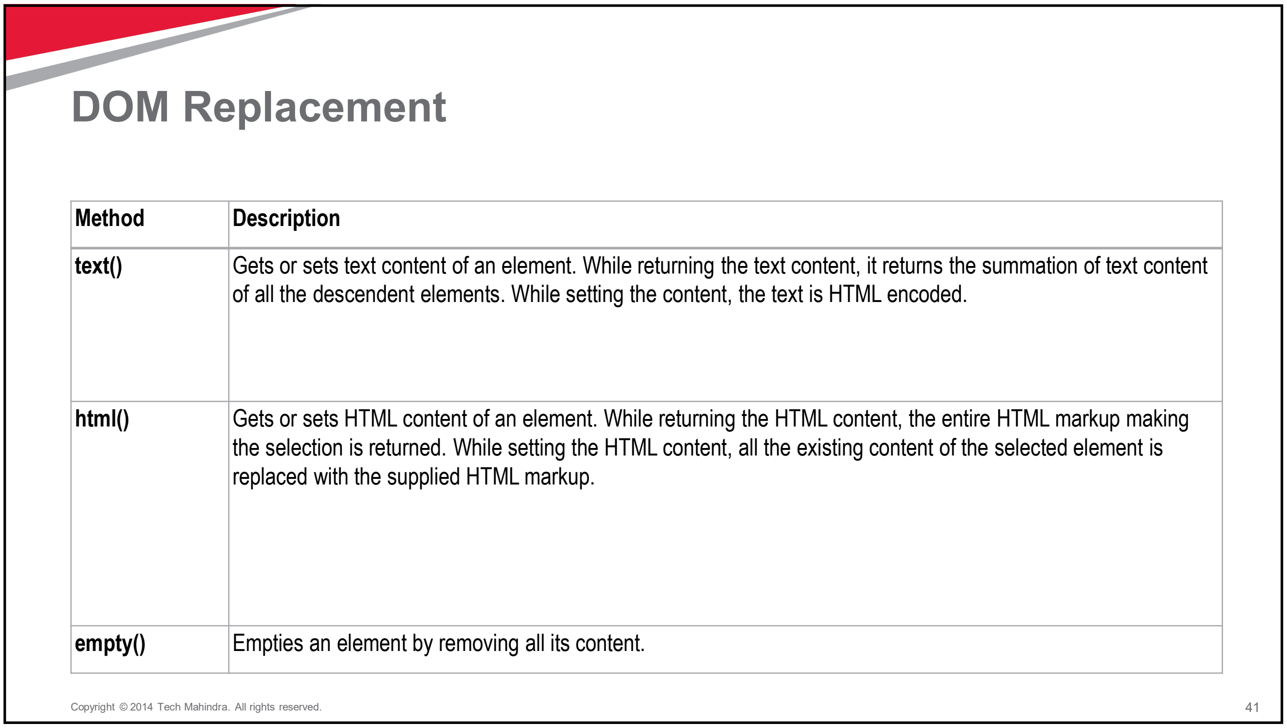
addClass()	Adds one or more CSS classes to a set of selected elements. Class names to be added are specified as a parameter. Multiple class names are separated by a white space.
removeClass()	Removes one or more classes from a set of selected elements. The class name to be removed can be specified as a parameter. Multiple class names are separated by a white space. If no class name is supplied as a parameter, all the CSS classes are removed.
hasClass()	Returns true if a specified CSS class is assigned to any of the selected elements; false otherwise.
toggleClass()	Adds or removes one or more classes from a set of selected elements, depending on whether the CSS class is present. Alternatively, you can also specify whether the class is to be added or removed.

CSS Properties

Method	Description
css()	Gets or sets a value of a CSS style property for the selected elements.
height()	Gets or sets the CSS height property of an element, excludes border and padding.
innerHeight()	Returns the height of an element excluding the border, but including padding.
outerHeight()	Returns the height of an element including the border, padding, and optionally the margin.
width()	Gets or sets the CSS height property of an element, excludes border and padding.
innerWidth()	Returns the width of an element, excluding the border but including padding.
outerWidth()	Returns the width of an element, including the border, padding, and optionally the margin.
scrollLeft()	Gets or sets the horizontal scroll position, which is the number of pixels that are hidden from view to the left of the scrollable area. If an element is not scrollable/scrollbar thumb is at extreme left, scrollLeft() will return 0.
scrollTop()	Gets or sets the vertical scroll position, which is the number of pixels that are hidden from view from the top of the scrollable area. If an element is not scrollable/scrollbar thumb is at extreme top, scrollTop() will return 0.
position()	Returns an object that gives information about the x and y coordinates of an element. The returned object has two properties: left and top. The coordinates are returned relative to the offset parent, which is the one that is a closed container to the target element.
offset()	Gets or sets x and y coordinates of an element relative to the document. The coordinates are specified as an object with two properties: left and top.

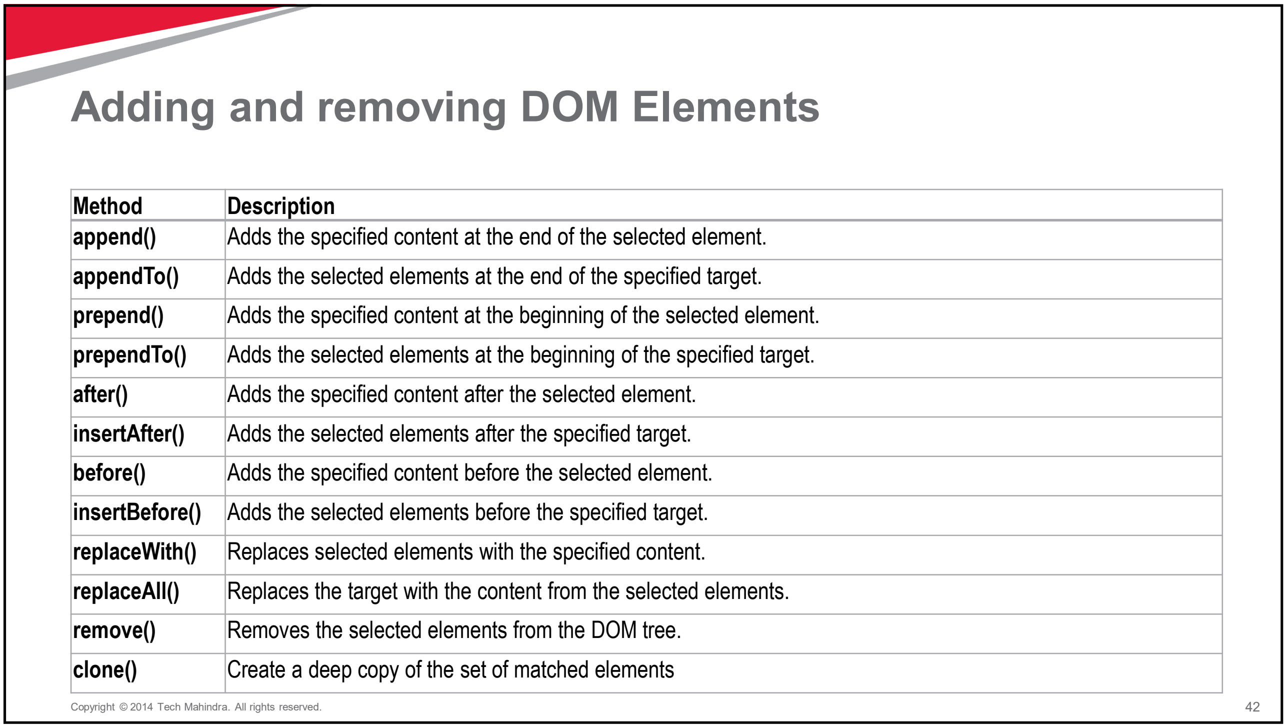
General Attributes

Method	Description
attr()	Allows you to get or set the value of an attribute for the selected element. If the attribute is not specified on an element, undefined is returned.
removeAttr()	Removes an attribute from a selected element.
prop()	Allows you to get or set the value of a DOM property for the selected element. If the property is not assigned for the DOM element, undefined is returned.
removeProp()	Removes a property from a selected element.
val()	Used to get or set values of input, select, and textarea elements.



DOM Replacement

Method	Description
text()	Gets or sets text content of an element. While returning the text content, it returns the summation of text content of all the descendent elements. While setting the content, the text is HTML encoded.
html()	Gets or sets HTML content of an element. While returning the HTML content, the entire HTML markup making the selection is returned. While setting the HTML content, all the existing content of the selected element is replaced with the supplied HTML markup.
empty()	Empties an element by removing all its content.



Adding and removing DOM Elements

Method	Description
append()	Adds the specified content at the end of the selected element.
appendTo()	Adds the selected elements at the end of the specified target.
prepend()	Adds the specified content at the beginning of the selected element.
prependTo()	Adds the selected elements at the beginning of the specified target.
after()	Adds the specified content after the selected element.
insertAfter()	Adds the selected elements after the specified target.
before()	Adds the specified content before the selected element.
insertBefore()	Adds the selected elements before the specified target.
replaceWith()	Replaces selected elements with the specified content.
replaceAll()	Replaces the target with the content from the selected elements.
remove()	Removes the selected elements from the DOM tree.
clone()	Create a deep copy of the set of matched elements

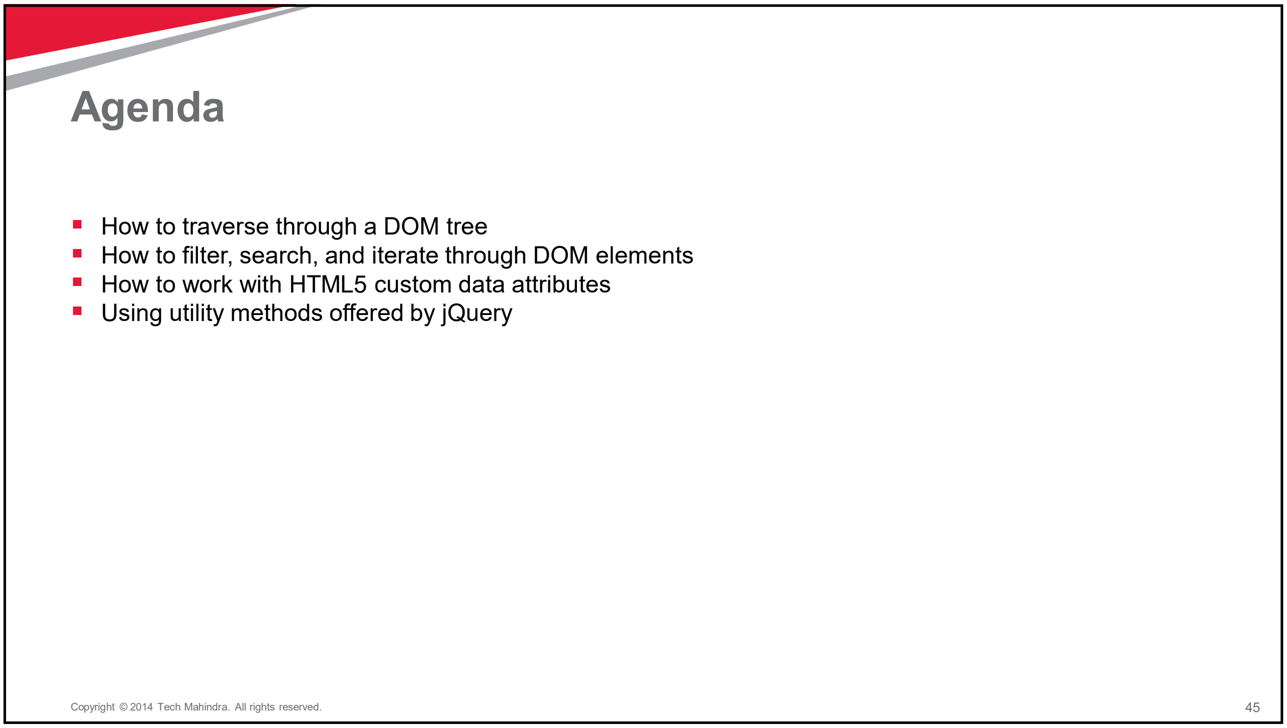
Summary

We discussed …

- How to work with CSS classes and CSS style properties
- How to manipulate the content of a DOM element
- How to add, remove, and replace DOM elements in a web page

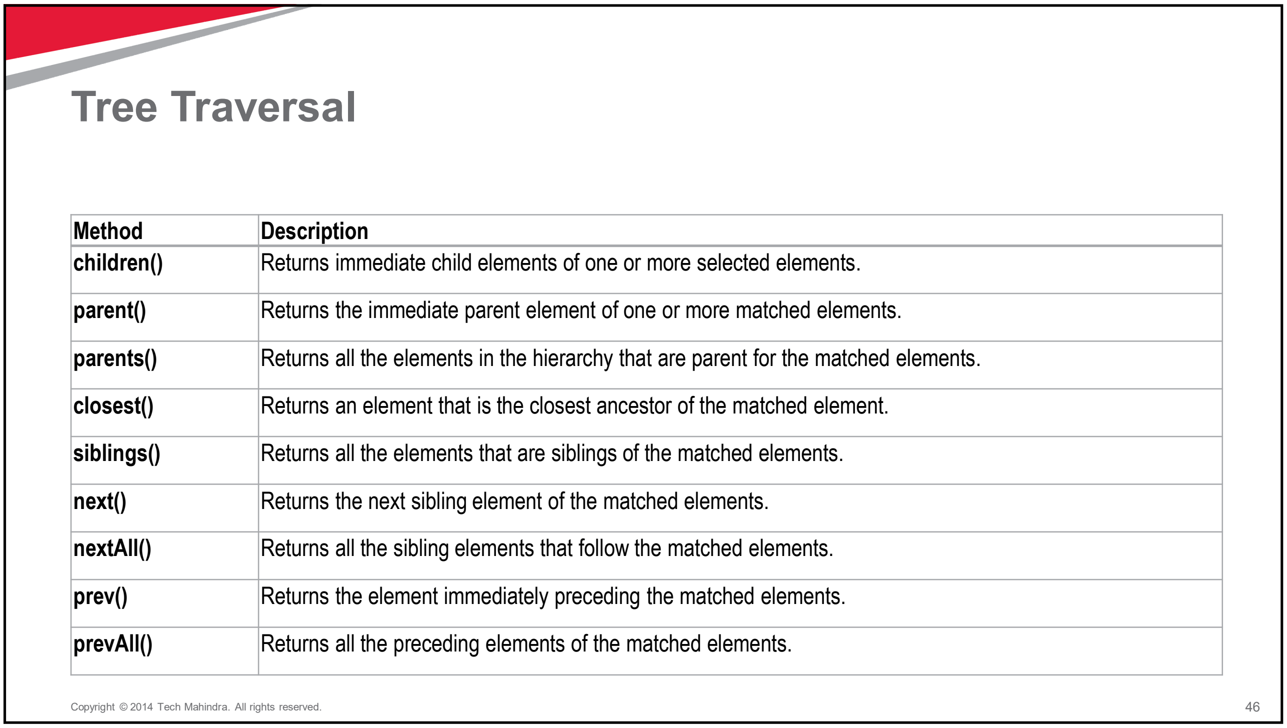
Tech Mahindra

Traversal and other useful methods



Agenda

- How to traverse through a DOM tree
- How to filter, search, and iterate through DOM elements
- How to work with HTML5 custom data attributes
- Using utility methods offered by jQuery



Tree Traversal

Method	Description
children()	Returns immediate child elements of one or more selected elements.
parent()	Returns the immediate parent element of one or more matched elements.
parents()	Returns all the elements in the hierarchy that are parent for the matched elements.
closest()	Returns an element that is the closest ancestor of the matched element.
siblings()	Returns all the elements that are siblings of the matched elements.
next()	Returns the next sibling element of the matched elements.
nextAll()	Returns all the sibling elements that follow the matched elements.
prev()	Returns the element immediately preceding the matched elements.
prevAll()	Returns all the preceding elements of the matched elements.

Filtering methods

Method	Description
first()	Returns the first element from the selected DOM elements.
last()	Returns the last element from the selected DOM elements.
eq()	Returns an element at specified index (zero-based) from the selected DOM elements.
has()	Returns all the elements from a set of DOM elements that have a given element as their descendant.
is()	Returns true if at least one element matches the criteria specified in the parameter; returns false otherwise.
not()	Returns all the elements that do not match the criteria specified in the parameter.
slice()	Returns all the elements between a zero-based start index and an end index (or until the end if the end index is not specified).
filter()	Returns all the elements from the selected DOM elements that match the specified condition.
map()	Used to pass each element from the selected DOM elements though a function and get the return values of the function calls.

Helper methods

Method / Property	Description
get()	Returns the DOM element(s) matched by the jQuery object under consideration. It can take a zero-based index to retrieve just an object at that index.
index()	Returns a zero-based index of a DOM element with respect to its siblings. It can also take a selector or element as its parameter and return the index of the matching element.
toArray()	Returns a JavaScript array from a matched set of elements.
length	Returns the number of elements in a matched set.
data()	Returns the value of the custom data-* attribute

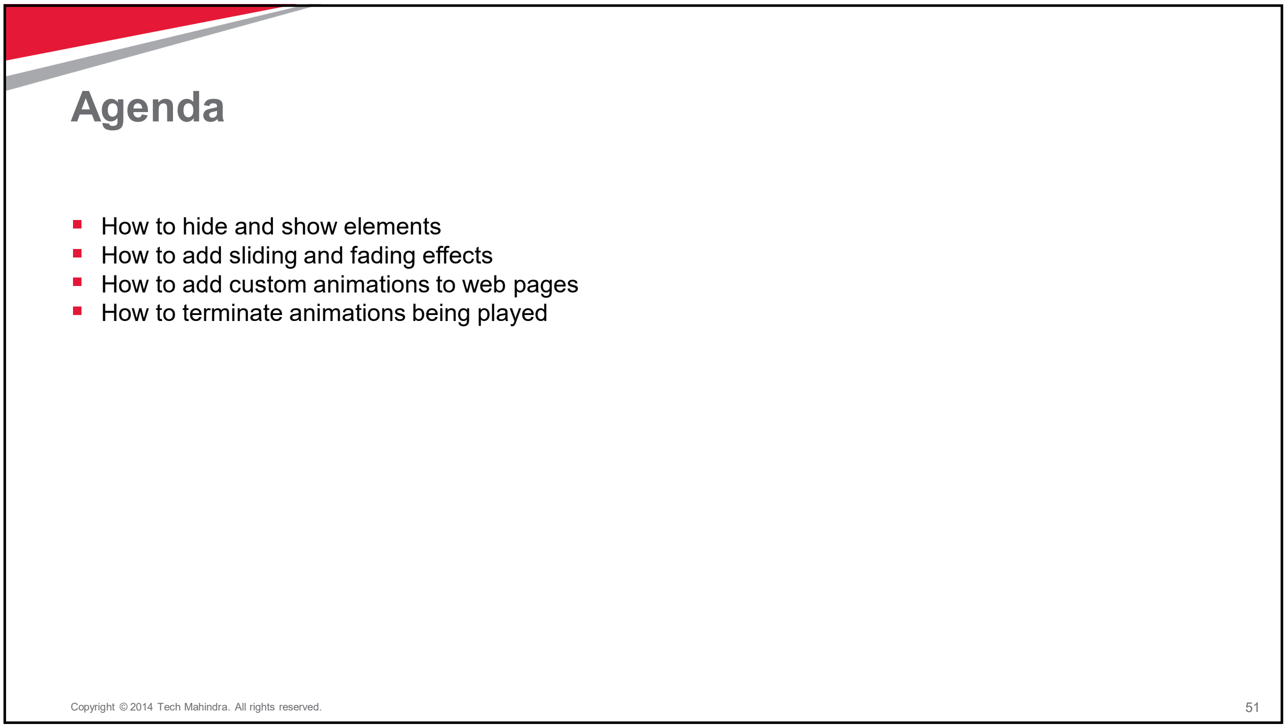
Summary

We discussed …

- How to traverse through a DOM tree
- How to filter, search, and iterate through DOM elements
- How to work with HTML5 custom data attributes
- Using utility methods offered by jQuery

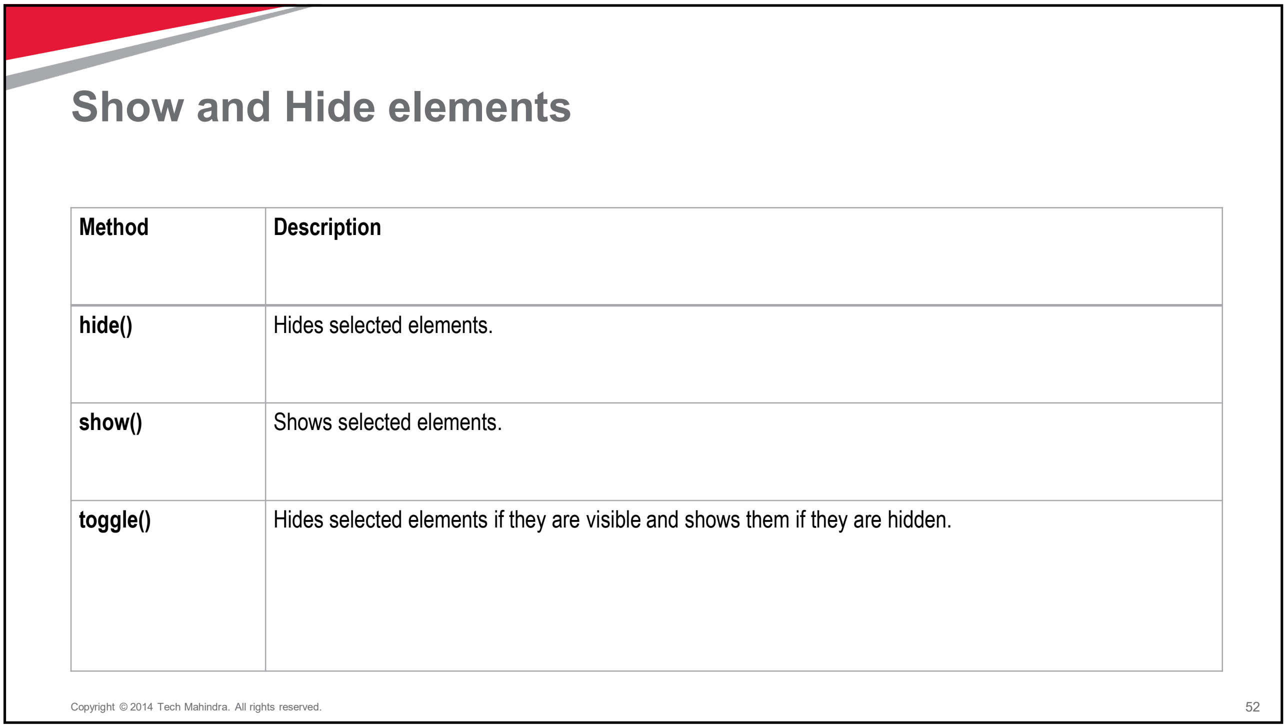
Tech Mahindra

Effects and Animations



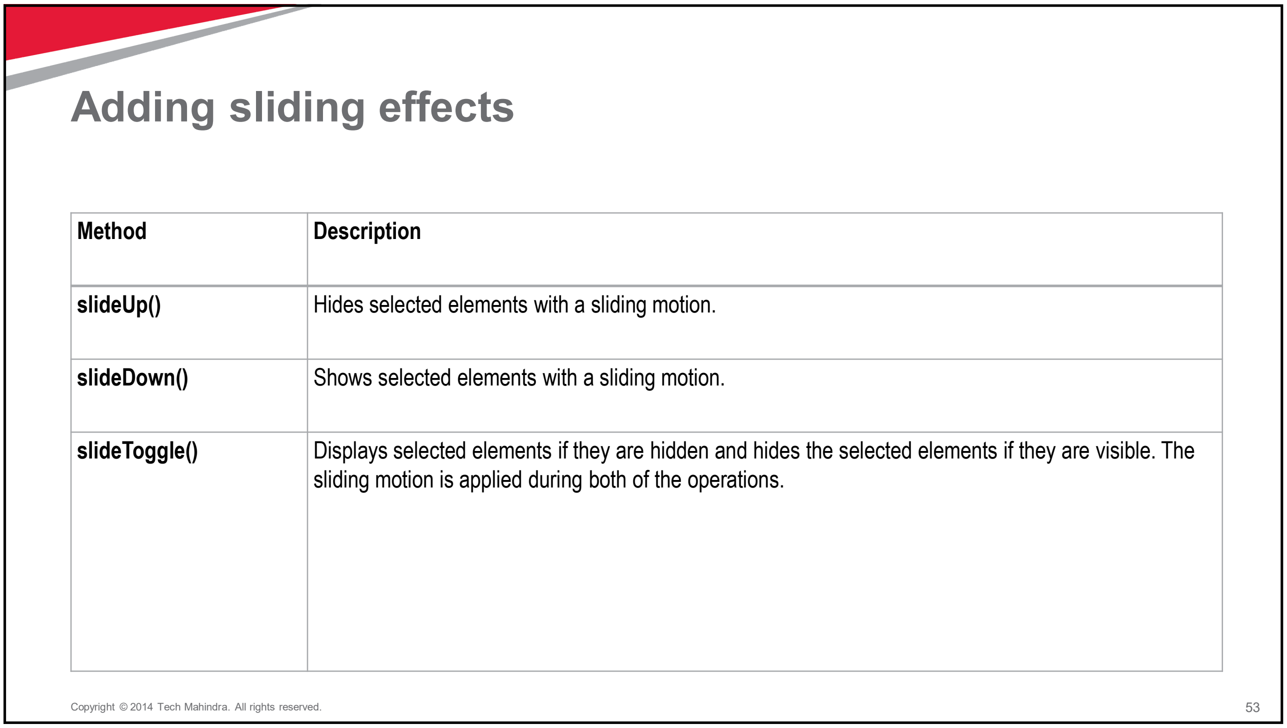
Agenda

- How to hide and show elements
- How to add sliding and fading effects
- How to add custom animations to web pages
- How to terminate animations being played



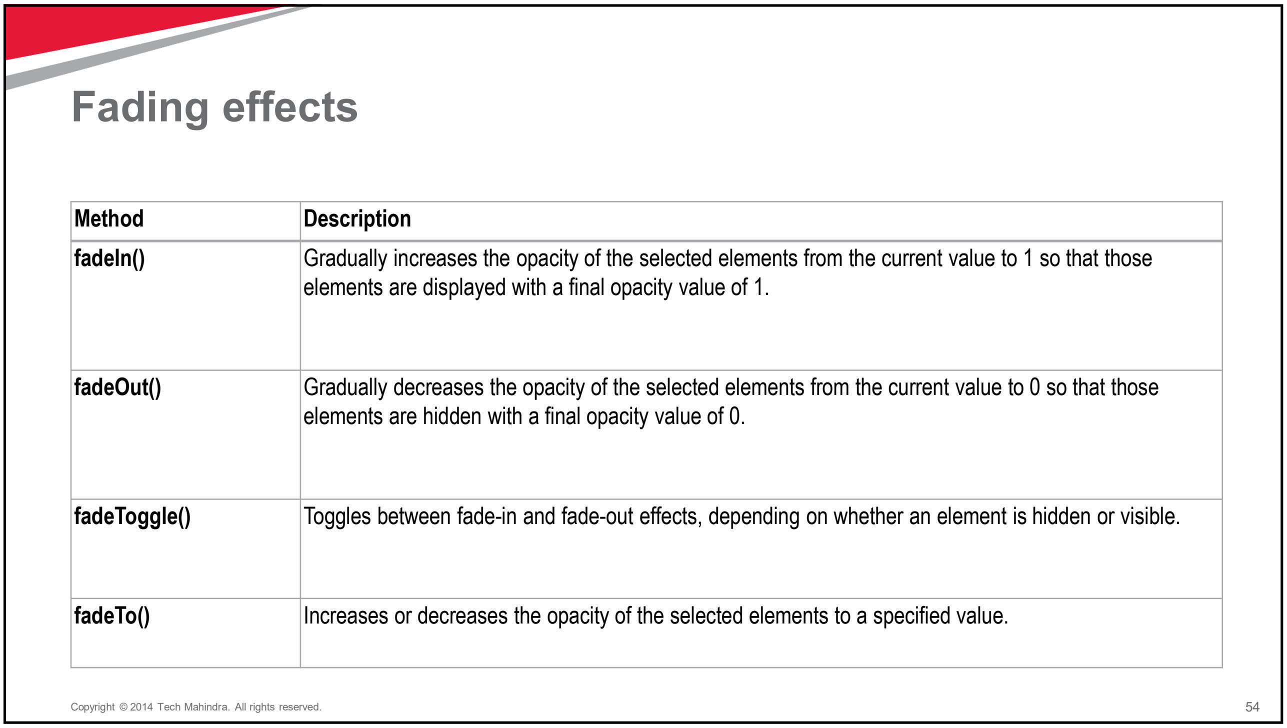
Show and Hide elements

Method	Description
hide()	Hides selected elements.
show()	Shows selected elements.
toggle()	Hides selected elements if they are visible and shows them if they are hidden.



Adding sliding effects

Method	Description
slideUp()	Hides selected elements with a sliding motion.
slideDown()	Shows selected elements with a sliding motion.
slideToggle()	Displays selected elements if they are hidden and hides the selected elements if they are visible. The sliding motion is applied during both of the operations.



Fading effects

Method	Description
fadeIn()	Gradually increases the opacity of the selected elements from the current value to 1 so that those elements are displayed with a final opacity value of 1.
fadeOut()	Gradually decreases the opacity of the selected elements from the current value to 0 so that those elements are hidden with a final opacity value of 0.
fadeToggle()	Toggles between fade-in and fade-out effects, depending on whether an element is hidden or visible.
fadeTo()	Increases or decreases the opacity of the selected elements to a specified value.

Summary

We discussed …

- How to hide and show elements
- How to add sliding and fading effects
- How to add custom animations to web pages
- How to terminate animations being played

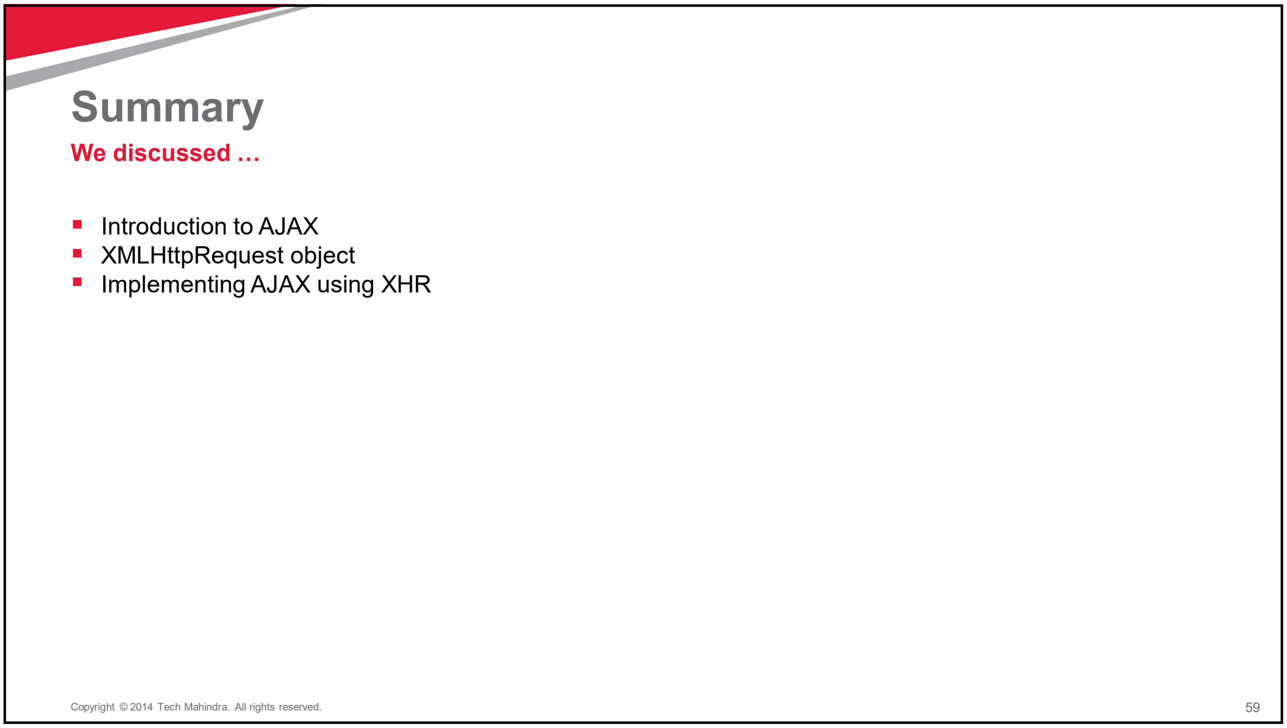
Tech Mahindra

AJAX Introduction

Agenda

- Introduction to AJAX
- XMLHttpRequest object
- Implementing AJAX using XHR

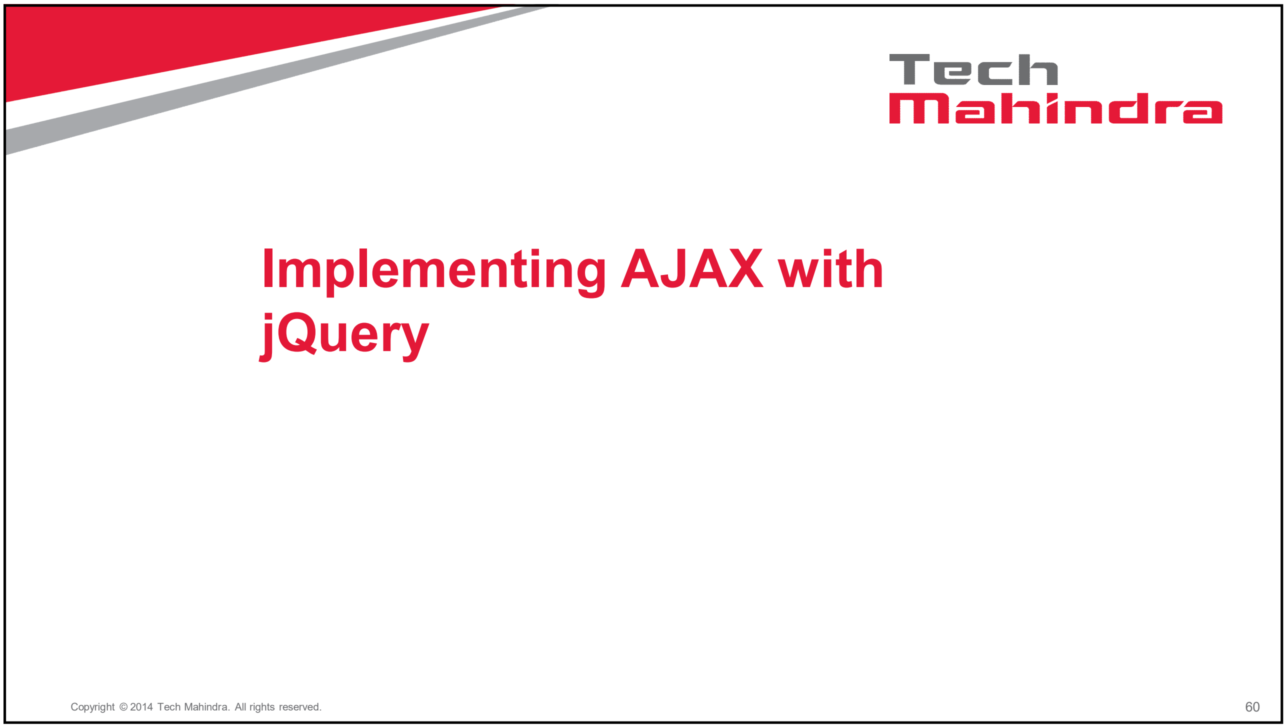
Demo



Summary

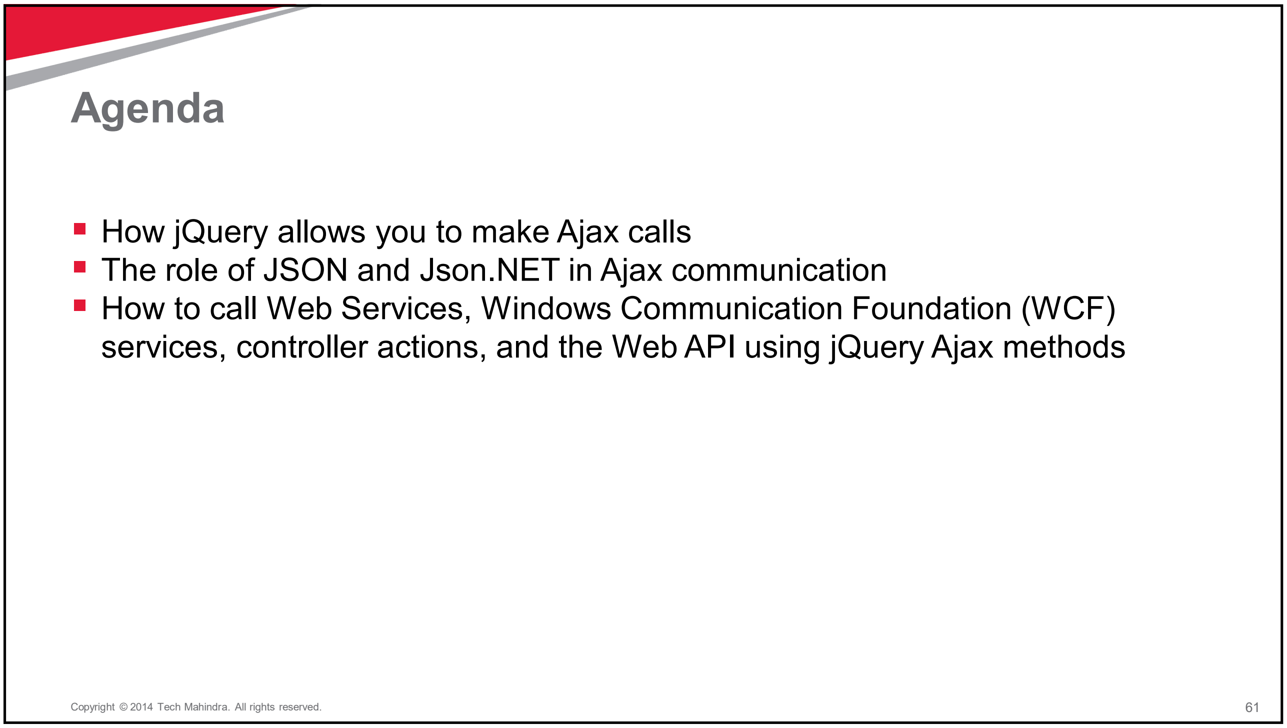
We discussed …

- Introduction to AJAX
- XMLHttpRequest object
- Implementing AJAX using XHR



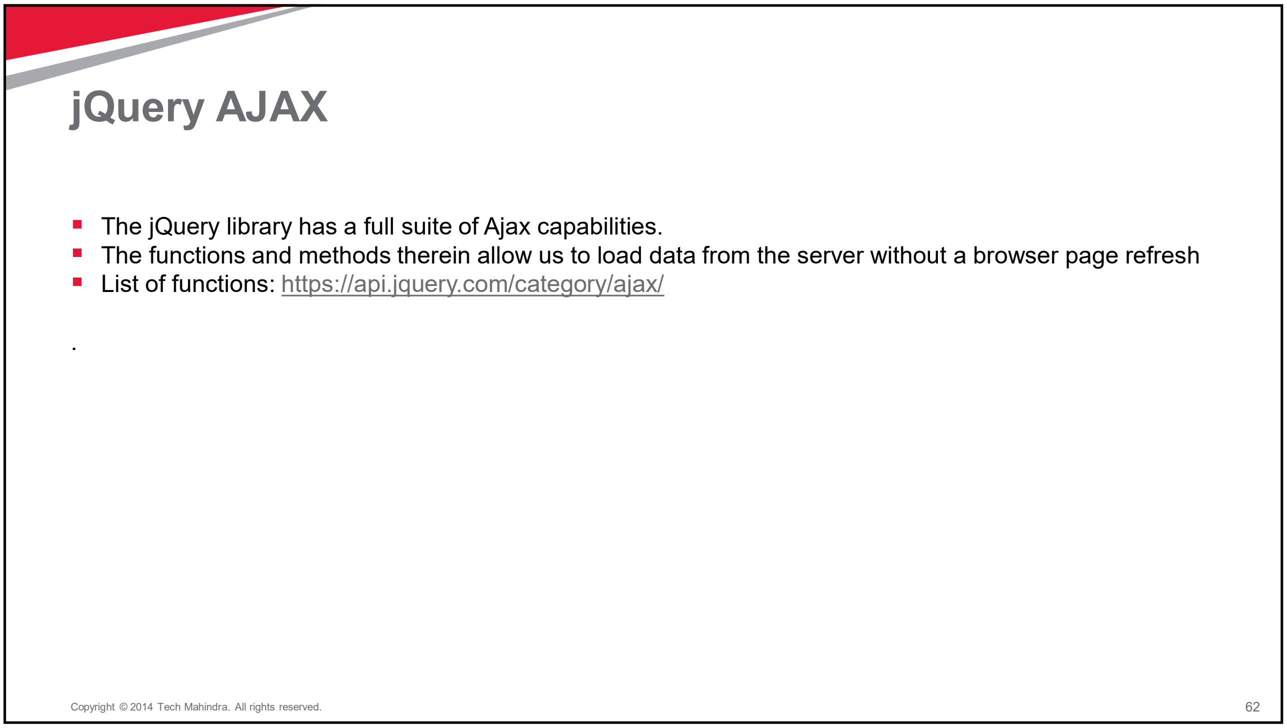
Tech Mahindra

Implementing AJAX with jQuery



Agenda

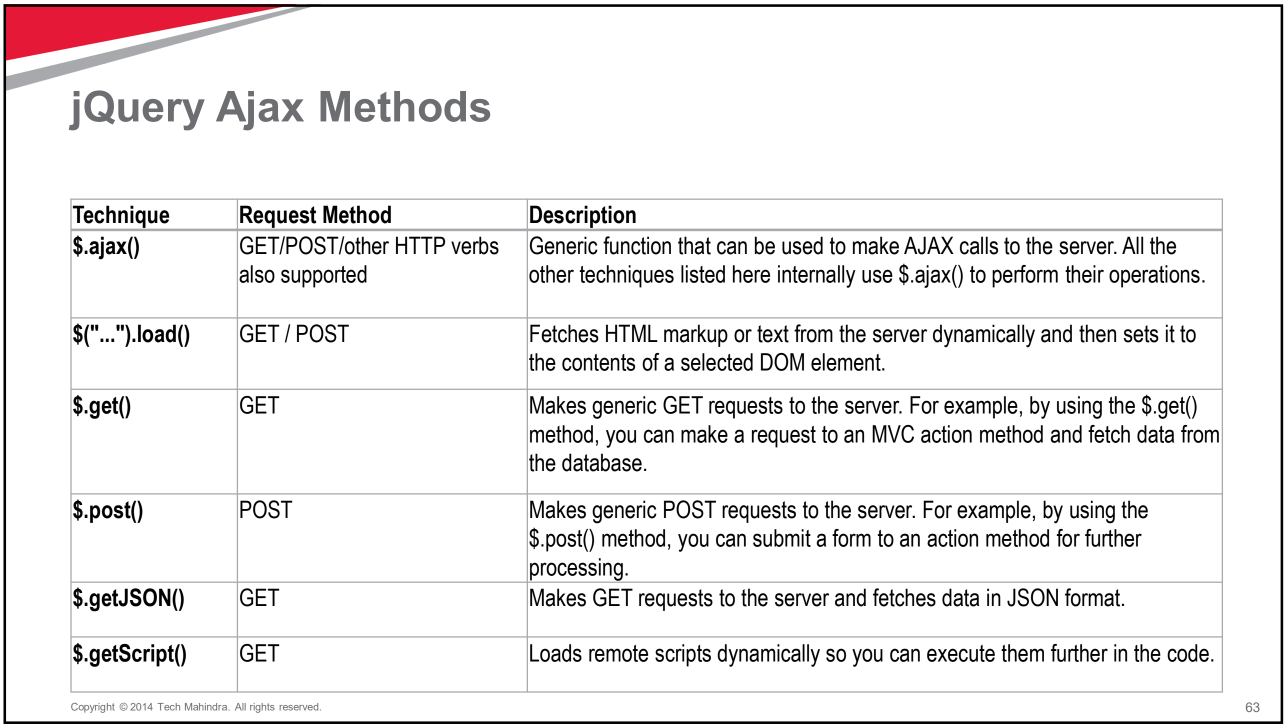
- How jQuery allows you to make Ajax calls
- The role of JSON and Json.NET in Ajax communication
- How to call Web Services, Windows Communication Foundation (WCF) services, controller actions, and the Web API using jQuery Ajax methods



jQuery AJAX

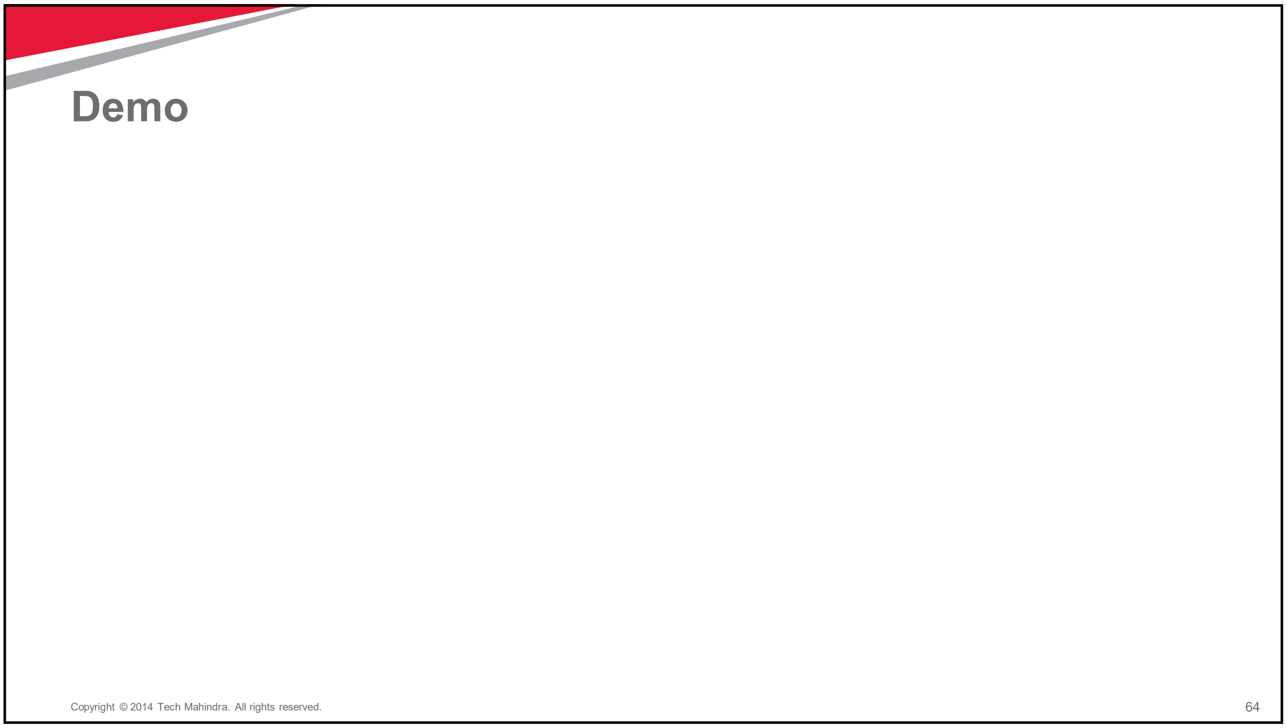
- The jQuery library has a full suite of Ajax capabilities.
- The functions and methods therein allow us to load data from the server without a browser page refresh
- List of functions: https://api.jquery.com/category/ajax/

.

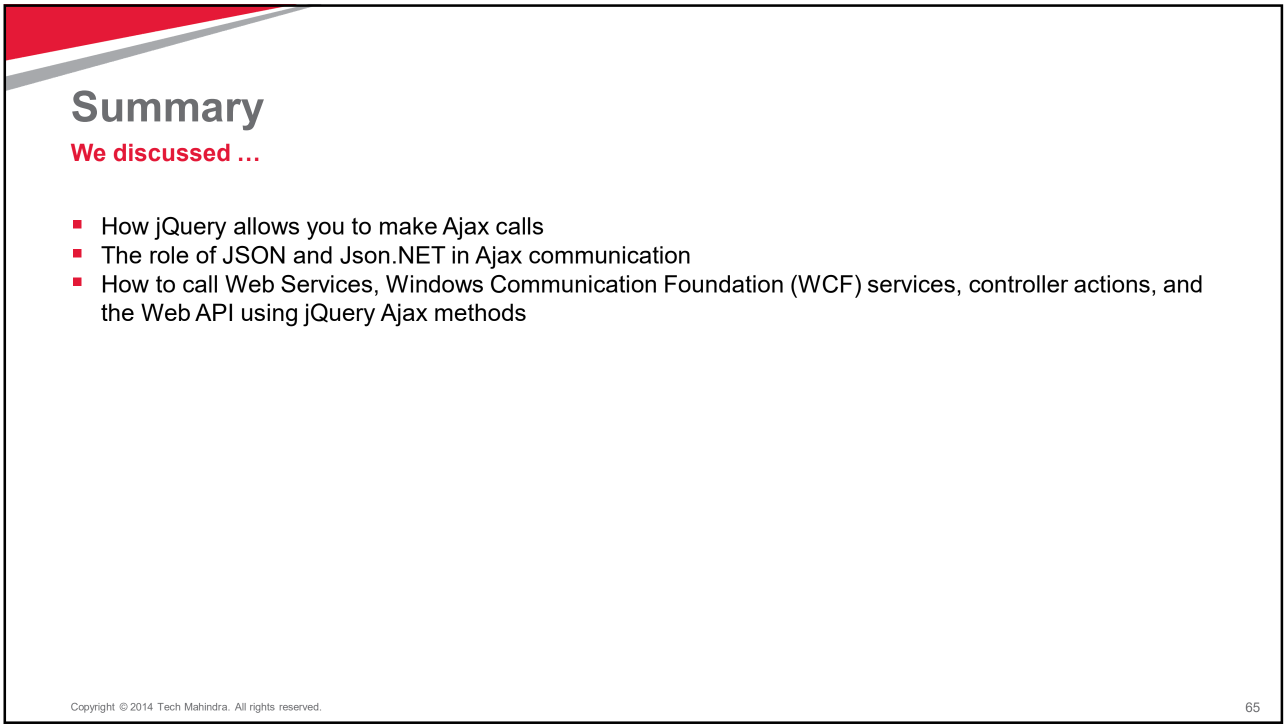


jQuery Ajax Methods

Technique	Request Method	Description
$.ajax()	GET/POST/other HTTP verbs also supported	Generic function that can be used to make AJAX calls to the server. All the other techniques listed here internally use $.ajax() to perform their operations.
$("...").load()	GET / POST	Fetches HTML markup or text from the server dynamically and then sets it to the contents of a selected DOM element.
$.get()	GET	Makes generic GET requests to the server. For example, by using the $.get() method, you can make a request to an MVC action method and fetch data from the database.
$.post()	POST	Makes generic POST requests to the server. For example, by using the $.post() method, you can submit a form to an action method for further processing.
$.getJSON()	GET	Makes GET requests to the server and fetches data in JSON format.
$.getScript()	GET	Loads remote scripts dynamically so you can execute them further in the code.



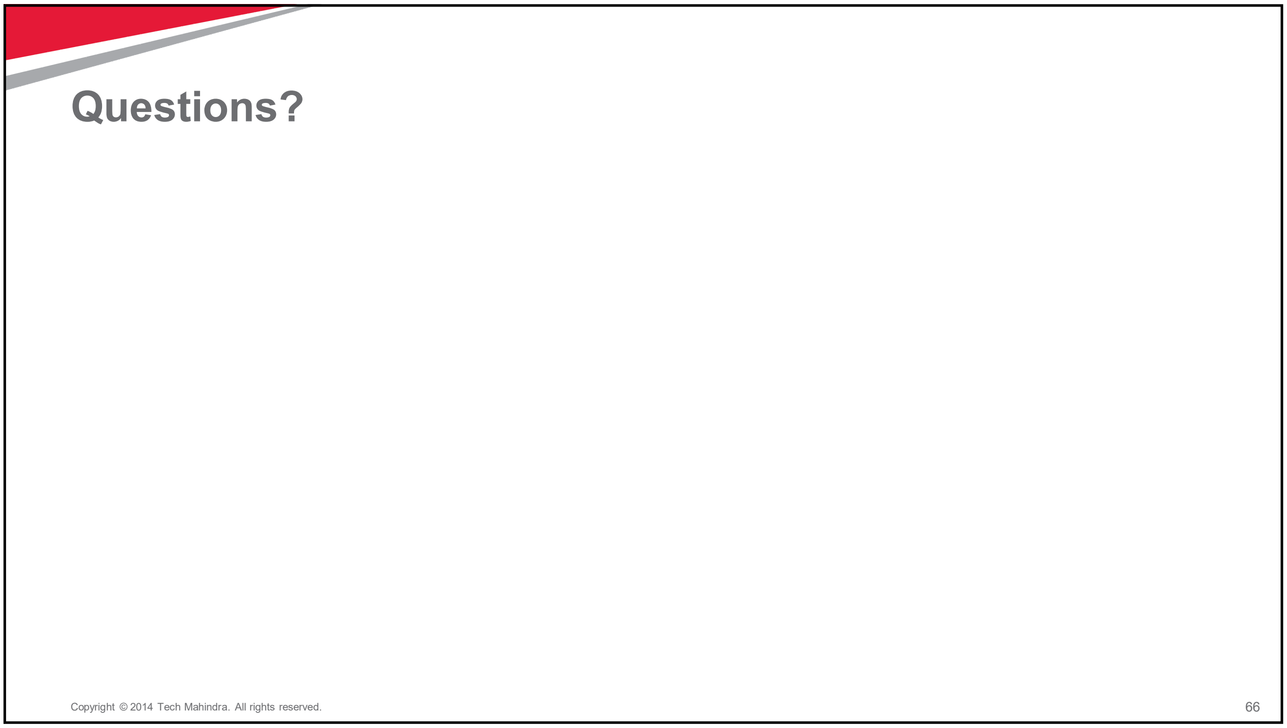
Demo



Summary

We discussed …

- How jQuery allows you to make Ajax calls
- The role of JSON and Json.NET in Ajax communication
- How to call Web Services, Windows Communication Foundation (WCF) services, controller actions, and the Web API using jQuery Ajax methods



Questions?

Thank you

Disclaimer

Tech Mahindra Limited, herein referred to as TechM provide a wide array of presentations and reports, with the contributions of various professionals. These presentations and reports are for informational purposes and private circulation only and do not constitute an offer to buy or sell any securities mentioned therein. They do not purport to be a complete description of the markets conditions or developments referred to in the material. While utmost care has been taken in preparing the above, we claim no responsibility for their accuracy. We shall not be liable for any direct or indirect losses arising from the use thereof and the viewers are requested to use the information contained herein at their own risk. These presentations and reports should not be reproduced, re-circulated, published in any media, website or otherwise, in any form or manner, in part or as a whole, without the express consent in writing of TechM or its subsidiaries. Any unauthorized use, disclosure or public dissemination of information contained herein is prohibited. Unless specifically noted, TechM is not responsible for the content of these presentations and/or the opinions of the presenters. Individual situations and local practices and standards may vary, so viewers and others utilizing information contained within a presentation are free to adopt differing standards and approaches as they see fit. You may not repackage or sell the presentation. Products and names mentioned in materials or presentations are the property of their respective owners and the mention of them does not constitute an endorsement by TechM. Information contained in a presentation hosted or promoted by TechM is provided “as is” without warranty of any kind, either expressed or implied, including any warranty of merchantability or fitness for a particular purpose. TechM assumes no liability or responsibility for the contents of a presentation or the opinions expressed by the presenters. All expressions of opinion are subject to change without notice.