TechLift

KICKSTART YOUR CAREER

WEB ENGINEERING

.Net

Week 2 - Day 6

Learning Objectives

By the end of this session, the students will have developed an understanding of:

▶ What is jQuery

▶ jQuery Syntax

▶ jQuery Selectors

▶ Manipulating HTML DOM with jQuery

JavaScript Quiz

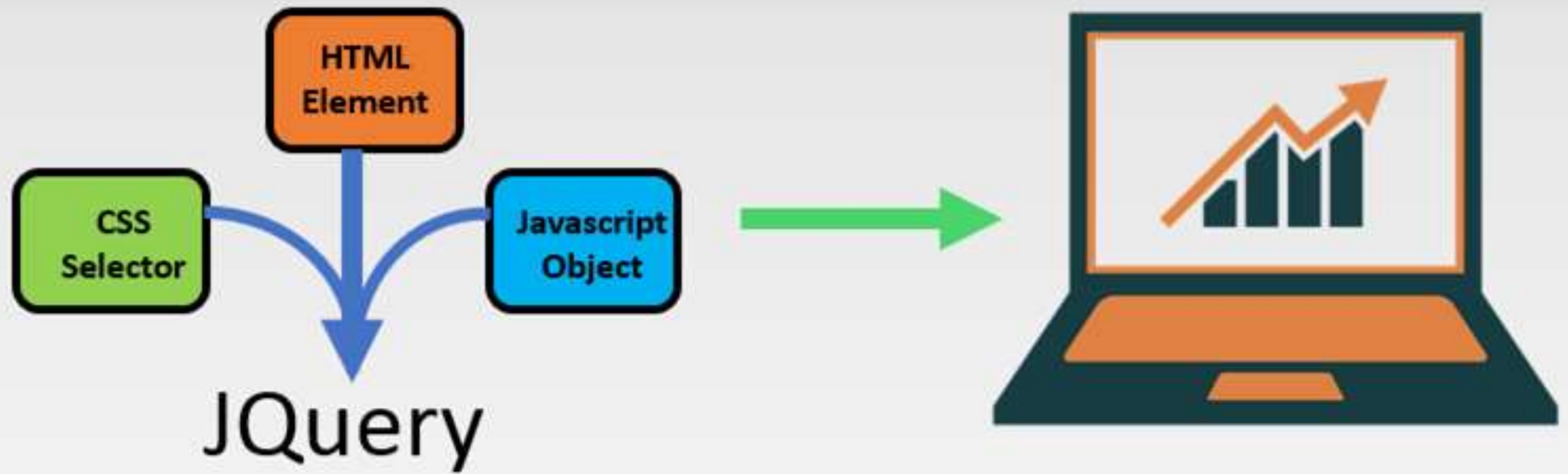Total Marks: 20
Time Allocated: 1 hr

: Basics

Install JQuery

JQuery will be installed in all the computers before starting its introduction.
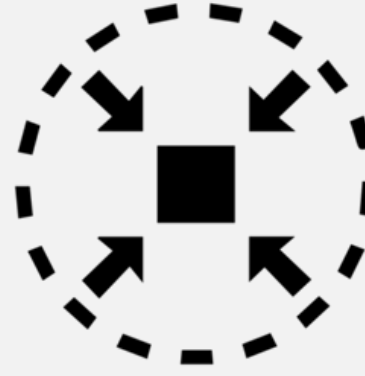
What is jQuery?

What is jQuery?

edureka!

Simplifies JavaScript

Event Handling

Lightweight

Animations

What is JQuery

- ➢ **Simplifies JavaScript:**

    It simplifies DOM manipulation and event handling for rapid web development

- ➢ **Event handling:**

    jQuery offers an effective manner to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code

- ➢ **Lightweight:**

    jQuery is a compact, lightweight library of about 19KB in size

- ➢ **Animations:**

    It comes with plenty of built-in animation effects which you can use in your web app to make it more interactive

What is it?

A library of JavaScript functions, just like...

- ➢ Prototype
- ➢ YUI
- ➢ Dojo
- ➢ mooTools

What is it?

➢ The jQuery library contains the following features:
- ○ HTML element selections
- ○ HTML element manipulation
- ○ CSS manipulation
- ○ HTML event functions
- ○ JavaScript effects and animations
- ○ HTML DOM traversal and modification
- ○ AJAX
- ○ Utilities

➢ An open source project, maintained by a group of developers with active support base and well written documentation.

jQuery philosophy

➢ Focus on the interaction between JavaScript

and HTML

➢ (Almost) every operation boils down to:
  ○ Find some stuff
  ○ Do something to it

jQuery

➢ a fast, small JavaScript library included in a single .js file
➢ built-in functions
➢ a framework built using JavaScript capabilities.
➢ all the functions and other capabilities available in JavaScript can be used
➢ Any JavaScript editor can be used to write jQuery such as
  ► Notepad
  ► Visual Studio
  ► Eclipse
  ► Ultra edit

Advantages of jQuery

- ➢ **Easy to learn:** jQuery is easy to learn because it supports same JavaScript style coding.
- ➢ **Write less do more**: jQuery provides a rich set of features that increase developers' productivity by writing less and readable code.
- ➢ **Excellent API Documentation:** jQuery provides excellent online API documentation.
- ➢ **Cross-browser support:** jQuery provides excellent cross-browser support without writing extra code.
- ➢ **Unobtrusive: j**Query is unobtrusive which allows separation of concerns by separating html and jQuery code.

Features of jQuery

jQuery includes the following features:

➢ DOM element selections using the cross-browser open source selector engine Sizzle, a spinoff out of the jQuery project
➢ DOM traversal and modification (including support for CSS 1-3)
➢ DOM manipulation based on CSS selectors that uses node elements name and node elements
➢ attributes (id and class) as criteria to build selectors
➢ Events
➢ Effects and animations
➢ Ajax
➢ Extensibility through plug-ins
➢ Cross-browser support

Features of jQuery

➢ jQuery simplifies various tasks of a progammer by writing less code. Here is the list of important core features supported by jQuery −

➢ DOM manipulation − The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called Sizzle.

➢ Event handling − The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.

➢ AJAX Support − The jQuery helps you a lot to develop a responsive and feature rich site using AJAX technology.

Features of jQuery

➢ Animations − The jQuery comes with plenty of built-in animation effects which you can use in your websites.

➢ Lightweight − The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).

➢ Cross Browser Support − The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+

➢ Latest Technology − The jQuery supports CSS3 selectors and basic XPath syntax.

Setting up jQuery

There are two ways to use jQuery.

➢ **Local Installation −** You can download jQuery library on your local machine and include it in your HTML code.

➢ **CDN Based Installation −** You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

JQuery Syntax

Document Ready Event

➢ Before we look into jQuery Syntax, let's try to understand what is Document Ready Event. Actually, before we execute any jQuery statement, we would like to wait for the document to be fully loaded.

➢ This is because jQuery works on DOM and if complete DOM is not available before executing jQuery statements, then we will not get desired result.

```
$(document).ready(function(){

    // jQuery code goes here...

});
```

Document Ready Event

➢ Following is basic syntax of a Document Ready Event:

```
$(document).ready(function(){

    // jQuery code goes here...


});
```

➢ Alternatively you can also use the following syntax for document ready event:

```
$(function(){

    // jQuery code goes here...


});
```

Standard jQuery Syntax

➢ A jQuery statement typically starts with the dollar sign ($) and ends with a semicolon (;).

➢ In jQuery, the dollar sign ($) is just an alias for jQuery. Let's consider the following example code which demonstrates the most basic statement of the jQuery

```
1  <script>
2      $(document).ready(function(){
3          // Some code to be executed...
4          alert("Hello World!");
5      });
6  </script>
```

Try this code on your computers and then check its output here.

JQuery Syntax

➤ With jQuery you select (query) HTML elements and perform "actions" on them.
➤ The jQuery syntax is tailor made for selecting HTML elements and perform some action on the element(s).
➤ Basic syntax is: **$(selector).action()**

► A dollar sign to define jQuery

► A (selector) to "query (or find)" HTML elements

► A jQuery action() to be performed on the element(s)

Examples:

➢ $(this).hide() - hides current element

➢ $("p").hide() - hides all paragraphs

➢ $("p.test").hide() - hides all paragraphs with class="test"

➢ $("#test").hide() - hides the element with id="test"

Example

Here are few examples to illustrate the basic jQuery Syntax. This example will select all the **<p>** elements from an HTML document and will hide those elements.

Write this code on your computer and tally your output from the one given on the next slide.

```html
<!doctype html>
<html>
<head>
<title>The jQuery Example</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("p").hide()
    });
</script>
</head>
<body>
    <h1>jQuery Basic Syntax</h1>

    <p>This is p tag</p>
    <p>This is another p tag</p>
    <span>This is span tag</span>
    <div>This is div tag</div>
</body>
</html>
```

Output

jQuery Basic Syntax

This is span tag
This is div tag

Example

Let's re-write the above example using jQuery() method instead of $():

Write this code on your computer and tally your output from the one given on the next slide.

```html
<!doctype html>
<html>
<head>
<title>The jQuery Example</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        jQuery("p").hide()
    });
</script>
</head>
<body>
    <h1>jQuery Basic Syntax</h1>

    <p>This is p tag</p>
    <p>This is another p tag</p>
    <span>This is span tag</span>
    <div>This is div tag</div>
</body>
</html>
```

Output

jQuery Basic Syntax

This is span tag
This is div tag

jQuery - Selectors

➢ jQuery Selectors are used to select HTML element(s) from an HTML document. Consider an HTML document is given and you need to select all the <div> from this document. This is where jQuery Selectors will help.

➢ jQuery Selectors can find HTML elements (ie. Select HTML elements) based on the following:

➢ HTML element Name
  ○ Element ID
  ○ Element Class
  ○ Element attribute name
  ○ Element attribute value
  ○ Many more criteria

jQuery - Selectors

➢ The jQuery library harnesses the power of Cascading Style Sheets (CSS) selectors to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).

➢ jQuery Selectors works in very similar way on an HTML document like an SQL Select Statement works on a Database to select the records.

jQuery Selector Syntax

➢ Following is the jQuery Selector Syntax for selecting HTML elements:

```
$(document).ready(function(){
        $(selector)
});
```

jQuery Selector Syntax

➢ Following is the jQuery Selector Syntax for selecting HTML elements:

```
$(document).ready(function(){
    $(selector)
});
```

➢ A jQuery selector starts with a dollar sign $ and then we put a selector inside the braces (). Here $() is called factory function, which makes use of following three building blocks while selecting elements in a given document:

jQuery Selector Syntax

➢ A jQuery selector starts with a dollar sign $ and then we put a selector inside the braces (). Here $() is called factory function, which makes use of following three building blocks while selecting elements in a given document:

Selector Name	Description
The element Selector	Represents an HTML element name available in the DOM. For example $('p') selects all paragraphs <p> in the document.
The #id Selector	Represents a HTML element available with the given ID in the DOM. For example $('#some-id') selects the single element in the document that has some-id as element Id.
The .class Selector	Represents a HTML elements available with the given class in the DOM. For example $('.some-class') selects all elements in the document that have a class of some-class.

1- The element Selector

➢ The jQuery element selector selects HTML element(s) based on the element name. Following is a simple syntax of an element selector:

```
$(document).ready(function(){
    $("Html Element Name")
});
```

Example

➢ This is an example to select all the <p> elements from an HTML document and then change the background color of those paragraphs. You will not see any <p> element in the output generated by this example.

➢ You can also change the code to use different element names as selector and then move to the next slide to verify the result.

```html
<!doctype html>
<html>
<head>
<title>The jQuery Example</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("p").css("background-color", "yellow");
    });
</script>
</head>
<body>
    <h1>jQuery element Selector</h1>

    <p>This is p tag</p>
    <span>This is span tag</span>
    <div>This is div tag</div>
</body>
</html>
```

Example Output

jQuery element Selector

This is p tag

This is span tag
This is div tag

2- The #id Selector

➢ The jQuery #id selector selects an HTML element based on the element id attribute. Following is a simple syntax of a #id selector:

```
$(document).ready(function(){
    $("#id of the element")
});
```

Example

➢ This is an example to select the <p> element whose id is foo and change the background color of those paragraphs..

➢ You can also change the code to use different element names as selector and then move to the next slide to verify the result.

```html
<!doctype html>
<html>
<head>
<title>The jQuery Example</title>
<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $("#foo").css("background-color", "yellow");
    });
</script>
</head>
<body>
    <h1>jQuery #id Selector</h1>

    <p id="foo">This is foo p tag</p>
    <span id="bar">This is bar span tag</span>
    <div id="bill">This is bill div tag</div>
</body>
</html>
```

Example Output

3- The .class Selector

➢ The jQuery .class selector selects HTML element(s) based on the element class attribute. Following is a simple syntax of a .class selector:

```
$(document).ready(function(){
    $(".class of the element")
});
```

Example

➢ This is an example to select all the elements whose class is foo and change the background color of those elements.

➢ You can also change the code to use different element names as selector and then move to the next slide to verify the result.

```html
<!doctype html>
<html>
<head>
<title>The jQuery Example</title>
<script src = "https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function() {
        $(".foo").css("background-color", "yellow");
    });
</script>
</head>
<body>
    <h1>jQuery .class Selector</h1>

    <p class="foo">This is foo p tag</p>
    <p class="foo">This is one more foo p tag</p>
    <span class="bar">This is bar span tag</span>
    <div class="bill">This is bill div tag</div>
</body>
</html>
```

Example Output

jQuery .class Selector

This is foo p tag

This is one more foo p tag

This is bar span tag
This is bill div tag

jQuery - DOM Manipulation

jQuery - DOM Manipulation

➤ jQuery provides a number of methods to manipulate DOM in efficient way. You do not need to write big and complex code to set or get the content of any HTML element.

jQuery DOM Manipulation

➤ jQuery provides methods such as attr(), html(), text() and val() which act as getters and setters to manipulate the content from HTML documents.

Document Object Model (DOM) - is a W3C (World Wide Web Consortium) standard that allows us to create, change, or remove elements from the HTML or XML documents.

jQuery - DOM Manipulation

Here are some basic operations which you can perform on DOM elements with the help of jQuery standard library methods −

- ➢ Extract the content of an element
- ➢ Change the content of an element
- ➢ Adding a child element under an existing element
- ➢ Adding a parent element above an existing element
- ➢ Adding an element before or after an existing element
- ➢ Replace an existing element with another element
- ➢ Delete an existing element
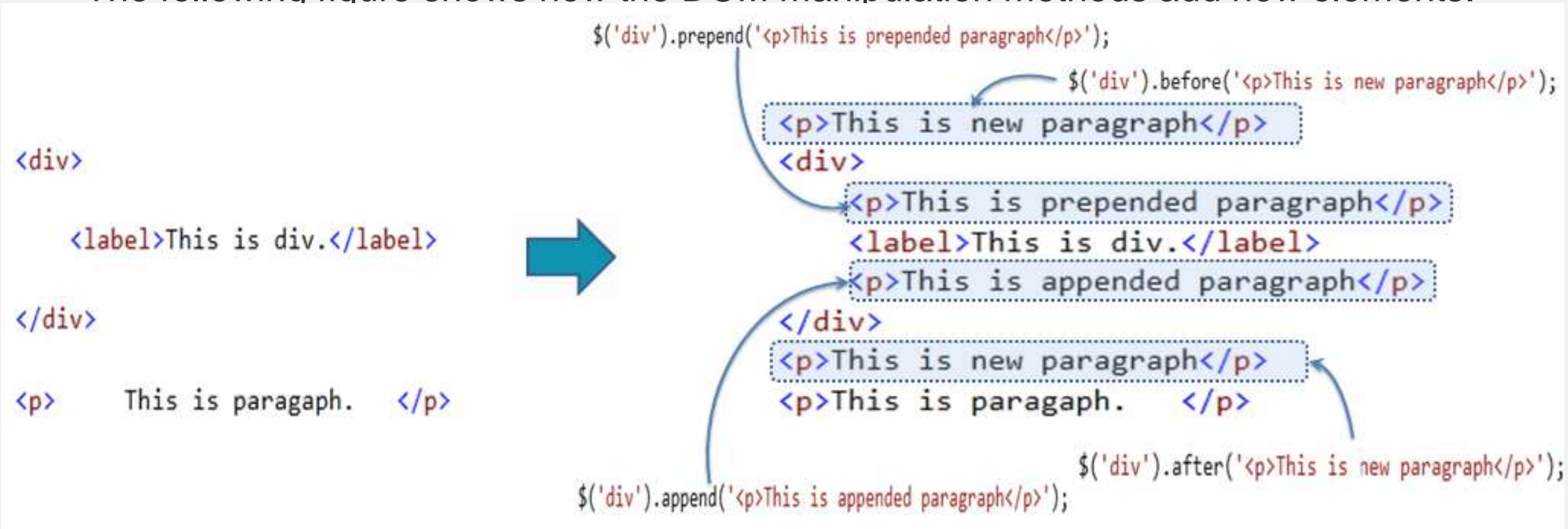- ➢ Wrapping content with-in an element

jQuery - DOM Manipulation

➢ jQuery provides various methods to add, edit or delete DOM element(s) in the HTML page.

➢ The following table lists some important methods to add/remove new DOM elements.

jQuery - DOM Manipulation

Method	Description
append()	Inserts content to the end of element(s) which is specified by a selector.
before()	Inserts content (new or existing DOM elements) before an element(s) which is specified by a selector.
after()	Inserts content (new or existing DOM elements) after an element(s) which is specified by a selector.
prepend()	Insert content at the beginning of an element(s) specified by a selector.
remove()	Removes element(s) from DOM which is specified by selector.
replaceAll()	Replace target element(s) with specified element.
wrap()	Wrap an HTML structure around each element which is specified by selector.

jQuery - DOM Manipulation

The following figure shows how the DOM manipulation methods add new elements.

```
$('div').prepend('<p>This is prepended paragraph</p>');

                                                    $('div').before('<p>This is new paragraph</p>');

                              <p>This is new paragraph</p>
<div>                         <div>
                                 <p>This is prepended paragraph</p>
  <label>This is div.</label>     <label>This is div.</label>
                                 <p>This is appended paragraph</p>
</div>                        </div>
                              <p>This is new paragraph</p>
<p>    This is paragaph.   </p>  <p>This is paragaph.    </p>

                                                    $('div').after('<p>This is new paragraph</p>');

$('div').append('<p>This is appended paragraph</p>');
```

jQuery after() Method

The jQuery after() method inserts content (new or existing DOM elements) after target element(s) which is specified by a selector.

Syntax:

```
$('selector expression').after('content');
```

Example: JQuery after() Method

Try the code and check the output here.

```
Example: jQuery after() Method

$('#div1').after('<div style="background-color:yellow"> New div </div>');

<div id="div1">div 1
</div>


<div id="div2">div 2
</div>
```

jQuery before() Method

The jQuery before() method inserts content (new or existing DOM elements) before target element(s) which is specified by a selector.

Syntax:

```
$('selector expression').before('content');
```

Example: JQuery before() Method

Try the code and check the output here.

```
$('#div1').before('<div style="background-color:yellow"> New div </div>');

<div id="div1">div 1
</div>


<div id="div2">div 2
</div>
```

jQuery append() Method

The jQuery append() method inserts content to the end of target element(s) which is specified by a selector.

Syntax:

```
$('selector expression').append('content');
```

Example: JQuery append() Method

Try the code and [check the output here.](#)

```
$('p').append('World!');


<p>Hello </p>
```

jQuery prepend() Method

The jQuery prepend() method inserts content at the beginning of an element(s) specified by a selector.

Syntax:

```
$('selector expression').prepend('content');
```

Example: JQuery prepend() Method

Try the code and check the output here.

```
$('div').prepend('<p>This is prepended paragraph</p>');

<div>
    <label>This is div.</label>
</div>
```

jQuery remove() Method

The jQuery remove() method removes element(s) as specified by a selector.

Syntax:

```
$('selector expression').remove();
```

Example: JQuery remove() Method

Try the code and check the output here.

```
$('label').remove();

<div>This is div.
    <label>This is label.</label>
</div>
```

jQuery replaceAll() Method

The jQuery replaceAll() method replaces all target elements with specified element(s).

Syntax:

```
$('content string').replaceAll('selector expression');
```

Example: JQuery replaceAll() Method

Try the code and <u>check the output here.</u>

```
$('<span>This is span</span>').replaceAll('p');


<div>
    <p>This is paragraph.</p>
</div>


<p>This is another paragraph.</p>
```

jQuery wrap() Method

The jQuery wrap() method wrap each target element with specified content element.

Syntax:

```
$('selector expression').wrap('content string');
```

Example: JQuery wrap() Method

Try the code and

```
$('span').wrap('<p></p>');

<div>

    <span>This is span.</span>
</div>
<span>This is span.</span>
```

Homework

Students will practice exercises of the topics covered today from this link:

JQuery DOM Practice Exercises

Learning Objectives

By the end of this session, the students have practised

✓ What is jQuery

✓ jQuery Syntax

✓ jQuery Selectors

✓ Manipulating HTML DOM with jQuery

Conclusion & Q/A

See you tomorrow!