Module 12: Day 1 - Introduction to jQuery

Lesson 1: jQuery Fundamentals

Introduction to jQuery

jQuery is a lightweight, fast, and feature-rich JavaScript library. It was built with the purpose of simplifying various tasks in JavaScript, such as HTML document traversal, event handling, and animations. Additionally, jQuery also provides capabilities for AJAX requests, paving the way for asynchronous web development.

What is jQuery?

Released in January 2006 by John Resig, jQuery was created to simplify the client-side scripting of HTML. Since its inception, it has grown to become one of the most prevalent JavaScript libraries available today. It offers an easy-to-use API that works seamlessly across a multitude of browsers and is completely open-source.

Why Learn jQuery?

- **Simplicity**: jQuery simplifies numerous complex tasks from JavaScript, such as AJAX calls and DOM manipulation.
- **Efficient Coding**: jQuery enables developers to write fewer lines of code for complex functionalities.
- **Cross-Browser Consistency**: jQuery effectively handles many cross-browser inconsistencies.
- **AJAX Support**: jQuery provides a simplified method of developing browser-friendly Ajax applications.
- **Large Community and Documentation**: jQuery has an extensive online documentation and a large community for support.

Getting Started

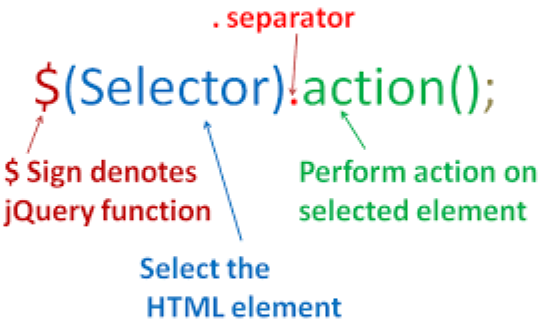
Incorporating jQuery into your project is simple and can be done in two main ways:

1. Download the jQuery library directly from jQuery.com
2. Include jQuery from a Content Delivery Network (CDN) ***(best option)***

jQuery Syntax Overview

jQuery statements typically start with the "$" sign and are often used to access an HTML element with a certain selector. For example: `$(selector).action()`

- **$**: The $ sign defines or accesses jQuery.
- **(selector)**: This is used to query HTML elements.
- **.action()**: These are the actions to be performed on the elements.


. separator
$(Selector).action();
$ Sign denotes jQuery function
Perform action on selected element
Select the HTML element

Consider the following example:

```
$(function() {

  $("h1").text("Hello from jQuery!");

});
```

In this example, we are using jQuery to select an `<h1>` element and change its text to "Hello from jQuery!" as soon as the document is ready. This is a simple demonstration of how jQuery can interact and manipulate DOM elements.

Including jQuery via CDN

A CDN (Content Delivery Network) provides a quick and simple method to start using jQuery. It requires no downloads or installations. To include jQuery in your HTML file through a CDN, use the following script tag:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

This script tag includes the minified version of jQuery, ensuring faster load times due to the smaller file size. It can be placed within the `<head>` tag or just before the closing `</body>` tag **(above any script link that depends on jQuery)** in your HTML document. Placing the script at the end of the body allows the page to fully load before loading jQuery, thus enhancing performance.

Additional Resources:

- **jQuery Official Website:** https://jquery.com/
- **jQuery API Documentation:** https://api.jquery.com/
- **Learning jQuery:** https://learn.jquery.com/
- **Introduction to jQuery:** https://www.freecodecamp.org/learn/front-end-development-libraries/jquery/
- **jQuery Tutorial - W3Schools:** https://www.w3schools.com/jquery/
- **jQuery AJAX Methods:** https://www.w3schools.com/jquery/jquery_ref_ajax.asp

Codepen Links:

Selecting Elements

https://codepen.io/shafferma08/pen/XWxNJGX

Lesson 2: jQuery DOM Manipulation

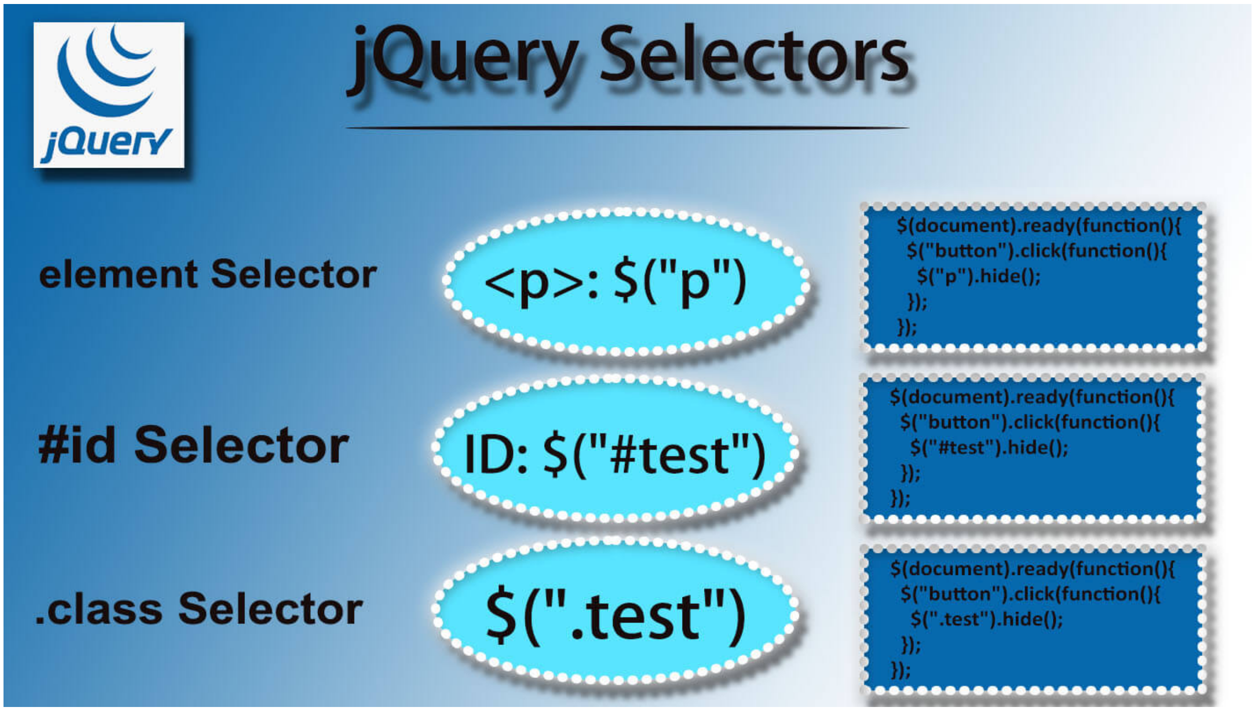
Introduction to jQuery DOM Manipulation

One of the reasons jQuery became so popular is because of its ability to efficiently traverse and manipulate the DOM (Document Object Model). The DOM represents a web page as a structured hierarchy of objects, allowing developers to interact with elements on a web page using JavaScript. jQuery makes DOM manipulation straightforward and intuitive.

DOM Manipulation with jQuery

Selecting DOM Elements

jQuery provides a powerful mechanism for selecting specific elements in a document, which is also known as the jQuery selectors. For instance, to select all paragraphs in an HTML document, you would use `$("p")`.



```
$(document).ready(function(){
 $("button").click(function(){
  $("p").hide();
 });
});
```

```
$(document).ready(function(){
 $("button").click(function(){
  $("#test").hide();
 });
});
```

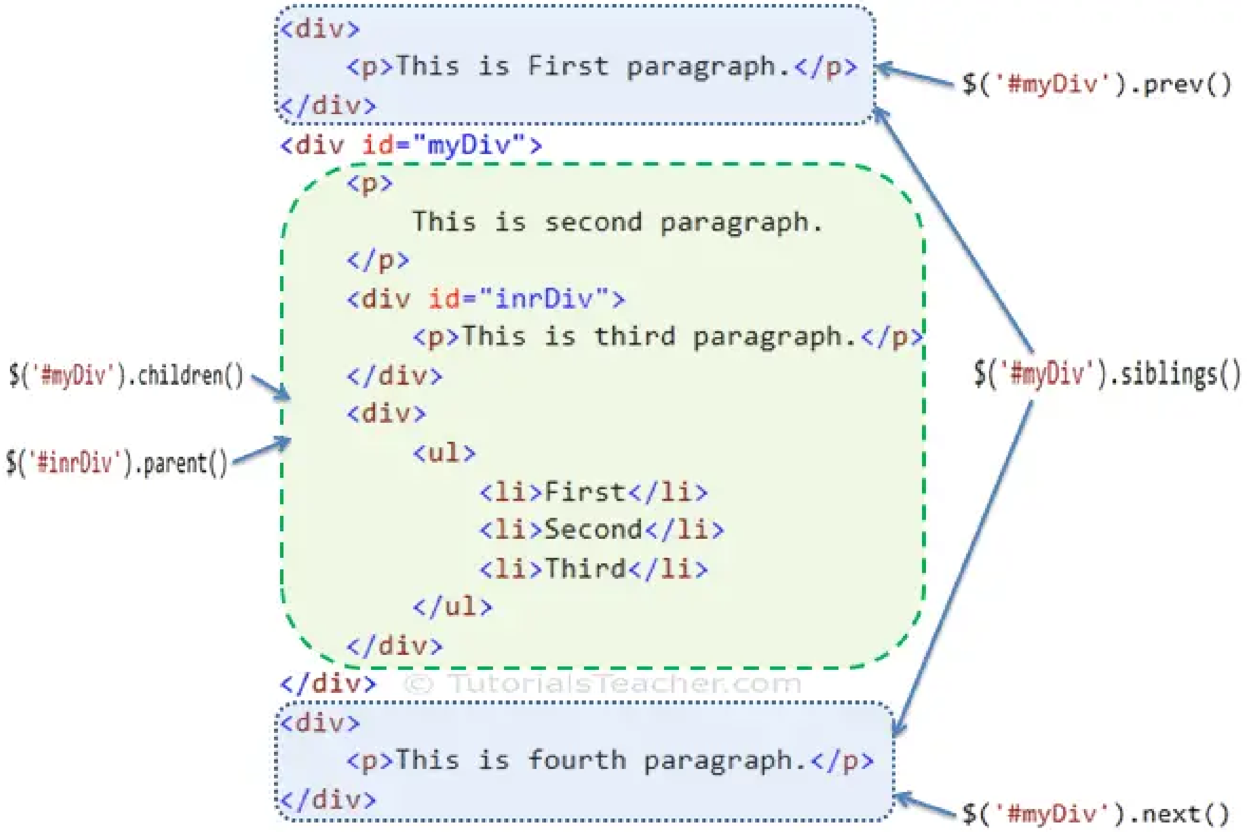
```
$(document).ready(function(){
 $("button").click(function(){
  $(".test").hide();
 });
});
```

Source: https://www.myprograming.com/jquery-selectors/

DOM Traversal

DOM traversal involves navigating around the DOM tree using jQuery. This can be achieved using methods such as `parent()`, `children()`, `siblings()`, `next()`, and `prev()`.

- `parent()`: Selects the direct parent of the selected element.
- `children()`: Selects all direct children of the selected element.
- `siblings()`: Selects all sibling elements of the selected element.
- `next()`: Selects the next sibling of the selected element.
- `prev()`: Selects the previous sibling of the selected element.



```
<div>
    <p>This is First paragraph.</p>
</div>
<div id="myDiv">
    <p>
        This is second paragraph.
    </p>
    <div id="inrDiv">
        <p>This is third paragraph.</p>
    </div>
    <div>
        <ul>
            <li>First</li>
            <li>Second</li>
            <li>Third</li>
        </ul>
    </div>
</div>
<div>
    <p>This is fourth paragraph.</p>
</div>
```

Codepen Link: https://codepen.io/shafferma08/pen/QWzJNaw

Source: https://www.tutorialsteacher.com/jquery/jquery-traversing-dom-elements

Modifying Styles and Attributes

jQuery provides methods to modify the styles and attributes of selected elements, such as `css()` and `attr()`.

- `css()`: Applies CSS styles to selected elements. You can either get the value of a style property (`css("propertyname")`) or set a new value (`css("propertyname", "value")`).

- `attr()`: Gets the attribute value for only the first element in the matched set. It returns undefined for values of elements that have not been set.

Adding and Removing DOM Elements

You can add new elements in the document or remove existing ones with jQuery using `append()`, `prepend()`, `remove()`, and `empty()`.

- `append()`: Inserts content at the end of the selected elements.
- `prepend()`: Inserts content at the beginning of the selected elements.
- `remove()`: Removes the selected element (and its child elements).
- `empty()`: Removes the child elements from the selected element.

Additional Resources:

- **jQuery DOM Manipulation (W3Schools):** https://www.w3schools.com/js/js_jquery_dom.asp
- **jQuery Traversing (W3Schools):**
- https://www.w3schools.com/jquery/jquery_traversing.asp https://www.w3schools.com/jquery/jquery_ref_traversing.asp
- **jQuery Manipulation (jQuery API Documentation):** https://api.jquery.com/category/manipulation/
- **jQuery Selectors (jQuery API Documentation):** https://api.jquery.com/category/selectors/

Lesson 3: jQuery Events and Animations

Introduction to jQuery Events and Animations

jQuery makes it easy to work with events and animations. Events are user interactions with the web page like a mouse click, key press, form submission, etc. Animations provide visual feedback and can enhance user interaction when used in moderation.

Working with Events in jQuery

Events are the backbone of any interactive web application. jQuery provides a robust and simple way to handle these events.

- `click()`: Triggers or attaches a function to handle click events.
- `mouseover()`: Triggers or attaches a function to handle mouseover events.
- `keydown()`: Triggers or attaches a function to handle keydown events.
- `focus() and blur()`: These are used to handle when an element gets focus and loses focus, respectively.
- `input()`: This event is fired synchronously when the value of an `<input>`, `<select>`, or `<textarea>` element is changed.
- `change()`: This event is fired for `<input>`, `<select>`, and `<textarea>` elements when an alteration to the element's value is committed by the user.
- `submit()`: This event is fired when a form is submitted.

jQuery Animations

Animations can help make your web pages more dynamic and interactive. Here's a quick overview of the most commonly used jQuery animation methods:

- `.fadeIn()`: Transitions the selected element to fully opaque (opacity: 1).
- `.fadeOut()`: Transitions the selected element to fully transparent (opacity: 0).
- `.fadeToggle()`: Fades the selected element in or out depending on its current opacity.
- `.show()`: Reveals the selected element by changing its display property.
- `.hide()`: Hides the selected element by changing its display property.
- `.slideUp()`: Hides the selected element with a slide-up motion by animating the height.
- `.slideDown()`: Shows the selected element with a slide-down motion by animating the height.

- `.slideToggle()`: Shows or hides the selected element with a slide-up or slide-down motion depending on its current state.
- `.animate()`: Defines custom animations on numeric CSS properties.

Remember to always manage your animations responsibly, as they can either enhance or deteriorate the user experience if not used appropriately.

Additional Resources:

- **jQuery Events (W3Schools):** https://www.w3schools.com/jquery/jquery_events.asp
- **jQuery Animation (W3Schools):** https://www.w3schools.com/jquery/jquery_animate.asp
- **jQuery Events (jQuery API Documentation):** https://api.jquery.com/category/events/
- **jQuery Effects (jQuery API Documentation):** https://api.jquery.com/category/effects/

Codepen Links:

Events and Listeners

https://codepen.io/shafferma08/pen/NWObPGY

jQuery Animations

https://codepen.io/shafferma08/pen/JjmbopK

Module 12 Collection:

https://codepen.io/collection/jbpjoB

AEL Hints / Guide

Codepen Link

Sandbox: https://codepen.io/shafferma08/pen/BaGPNjG

Hints and Tips for AEL 2.1

Selecting Elements with jQuery

Remember, to select an element with jQuery, you use the `$()` function, and the argument you pass into it is a string that describes what you want to select. The syntax is similar to CSS.

A. All the h1 elements on the page

- Hint: Think about how you'd select all `h1` elements in CSS.

B. The element with the ID navbar

- Hint: Recall that IDs in CSS are selected with a `#`.

C. All the input elements that are descendants of the element with the class form-group

- Hint: You can select descendants in jQuery just like in CSS, using a space.
- Hint: You can use the `find()` function to find descendants of a certain element.

D. All the li elements that are children of the element with the ID menu

- Hint: `children()` function can be used to select direct children elements.

E. All the elements with the class btn that are siblings of the element with the ID form

- Hint: The `siblings()` function can be used to select all sibling elements.

Selecting Specified Elements

In this section, you're given some HTML and asked to select certain elements. This requires you to understand the structure of the HTML and apply what you know about jQuery selection.

Modifying Styles and Attributes with jQuery

jQuery's `css()` function is used to get or set styles on selected elements. The `attr()` function is used similarly for attributes.

A. Set the background color of all div elements to blue

- Hint: You can use the `css()` function to set the `background-color` property.

B. Set the href attribute of all a elements to #

- Hint: You'll want to use the `attr()` function to change the `href`.

C. Set the src attribute of all img elements to a random image from Picsum

- Hint: `attr()` is also useful for changing `src`. The URL for a random image from Picsum is `https://picsum.photos/200`.

D. Set the font size of all p elements to 16px

- Hint: The `css()` function can be used to set `font-size`.

E. Get the value of the alt attribute of the first img element

- Hint: The `attr()` function without a second argument gets the value of the attribute.

Adding and Removing Elements

The `append()`, `prepend()`, and `html()` functions can be used to add elements to the DOM. `remove()` and `empty()` are used to remove elements.

A. Add a nav element with the following links: Home, About, Contact

- Hint: Consider using `append()`. Remember, you can create HTML as a string in jQuery.

B. Add a span element to the end of the h1 element with the text "(New)"

- Hint: Again, `append()` is useful here.

C. Remove the p element

- Hint: The `remove()` function is used to remove elements from the DOM.

Hints and Tips for AEL 3.1

In these exercises, you will need to combine your understanding of events and animations in jQuery. Remember, the general pattern is to first select the element you want to work with, then use jQuery methods to handle events and animations.

1. **Create a button that changes the background color of the page when clicked.**
    - Hint: You will need to create a button element and then add a `click` event listener to it. In the callback function of the `click` event, use the `css()` method to change the background color of the page.

2. **Create a form that displays a message when it is submitted.**
    - Hint: First, create a form element. You will then add a `submit` event listener to the form. In the callback function of the `submit` event, prevent the default form submission behavior using `event.preventDefault()`, and then use jQuery to display a message on the page.

3. **Create a div that slides up when clicked.**
    - Hint: You will need to create a div element, and then add a `click` event listener to it. In the callback function of the `click` event, use the `slideUp()` method to animate the div.

4. **Create a button that fades in an image when clicked. For the image you can use Unsplash, Picsum, or something along those lines.**
    - Hint: Start by creating a button and an image element. Initially, you could use the `hide()` method to make the image invisible. Then, add a `click` event listener to the button. In the callback function, use the `fadeIn()` method to make the image appear gradually when the button is clicked.

5. **Create a button that animates a div when clicked.**
    - Hint: Create a button and a div. Add a `click` event listener to the button. In the callback function, use the `animate()` method to animate some properties of the div. Remember that `animate()` only works with numeric properties.