DOM manipulation using jQuery:

1. Changing Text Content and Styling:

This example shows how to change the text content and apply CSS styles to an HTML element using jQuery.

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>DOM Manipulation Example</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
    // Change text content
    $('#myParagraph').text('New Text Content');

    // Apply CSS styles
    $('#myParagraph').css({
        'color': 'blue',
        'font-size': '18px',
        'font-weight': 'bold'
    });
});
</script>
</head>
<body>

<p id="myParagraph">Original Text Content</p>

</body>
</html>
```

In this example:
- We use jQuery to select the `<p>` element with the id `myParagraph`.
- We use the `.text()` method to change the text content of the element.
- We use the `.css()` method to apply CSS styles to the element.

2. Handling Click Event:

This example demonstrates how to handle a click event on a button using jQuery.

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>DOM Manipulation Example</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
    // Handle click event
    $('#myButton').click(function() {
        alert('Button Clicked!');
    });
});
</script>
</head>
<body>

<button id="myButton">Click Me</button>

</body>
</html>
```

In this example:
- We use jQuery to select the button element with the id `myButton`.
- We use the `.click()` method to attach a click event handler to the button.
- When the button is clicked, the anonymous function specified as the argument to `.click()` is executed, displaying an alert with the message "Button Clicked!".

Certainly! Here are two more examples of DOM manipulation using jQuery:

3. Adding and Removing Elements:

This example demonstrates how to dynamically add and remove HTML elements using jQuery.

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>DOM Manipulation Example</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

```
<script>
$(document).ready(function() {
    // Add element
    $('#addButton').click(function() {
        $('<li>New Item</li>').appendTo('#myList');
    });

    // Remove element
    $('#removeButton').click(function() {
        $('#myList li:last-child').remove();
    });
});
</script>
</head>
<body>

<ul id="myList">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>

<button id="addButton">Add Item</button>
<button id="removeButton">Remove Last Item</button>

</body>
</html>
```

In this example:
- We use jQuery to select the buttons with the ids `addButton` and `removeButton`.
- When the "Add Item" button is clicked, a new `<li>` element with the text "New Item" is appended to the `<ul>` element with the id `myList`.
- When the "Remove Last Item" button is clicked, the last `<li>` element inside the `<ul>` is removed.

4. Toggling Classes:

This example shows how to toggle a CSS class on an element using jQuery.

```html
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<title>DOM Manipulation Example</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
    // Toggle class
    $('#toggleClassButton').click(function() {
        $('#myElement').toggleClass('highlight');
    });
});
</script>
<style>
.highlight {
    background-color: yellow;
}
</style>
</head>
<body>

<div id="myElement">Toggle Class Example</div>

<button id="toggleClassButton">Toggle Class</button>

</body>
</html>
```

In this example:
- We use jQuery to select the button with the id `toggleClassButton`.
- When the button is clicked, the `.toggleClass()` method is used to toggle the `highlight` class on the `<div>` element with the id `myElement`.
- The `highlight` class applies a CSS rule that changes the background color of the element to yellow.

These examples demonstrate more ways in which jQuery can be used to manipulate the DOM, including adding/removing elements and toggling classes, making web development tasks easier and more efficient.