8.5 jQuery Get and Set

flower girls by normal rockwell - answered by Dante Magbuhos Jr.

- One very important part of jQuery is the possibility to manipulate the DOM.
- jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

Getting element text and values

- Three simple, but useful, jQuery methods for DOM manipulation are `text()`, `html()`, and `val()`.

Method	Description	Sample code	Code explanation

`.text()`	• Gets the text content of selected elements	`<div id="my_div">This is some text inside a div element.</div>`
		
		`<p id="my_para">This is a paragraph with some text.</p>`
		
		`<button id="get_text">Get Text</button>`
		
		`$(document).ready(function () {`
		`  $("#get_text").click(function () {`
		`    let text = $("#my_div").text();`
		`    let para_text = $("#my_para").text();`
		
		`    alert("Text from #my_div: " + text + "\nText from #para_text");`
		`  });`
			• When the "Get Text" button is clicked, it retrieves the text content of the element with `id="my_div"` and stores it in the `text` variable.
			• It also retrieves the text content of the element with the `id="my_para"` and stores it in the

		`});`	`para_text` variable. - An alert dialog is then displayed, showing the text content from both `id="my_div"` and `id="my_para"` elements.

`.html()`	Gets the content of selected elements (including HTML markup)	```html
		<div id="my_div">
		<p>This is some text inside a div element.</p>
		
		   - Item 1
		   - Item 2
		
		</div>
		<button id="get_html">Get HTML</button>
		
		$(document).ready(function () {
		  $("#get_html").click(function () {
		    let html_content = $("#my_div").html();
		
		    alert("HTML content from #my_div: " + html_content);
		  });
		});
		``
			- When the "Get HTML" button is clicked, it retrieves the HTML content of the element with `id="my_div"`, including its nested elements and their structure. - The HTML content is stored in the `html_content` variable. - An alert dialog is then displayed, showing the HTML content from the element with `id="my_div"`, including the paragraph and the list items.

`.val()`	Gets the value of form fields	```html
		<input type="text" id="my_input" value="Initial Value">
		<button id="get_value">Get Value</button>
		<p id="output">Input value will be displayed here.</p>
		
		$(document).ready(function() {
		  $("#get_value").click(function() {
		``
			When the "Get Value" button is clicked, it retrieves the current value from the input field and displays it in the `<div>` element with `id="output"`.

| | ```
    let input_value =
$("#my_input").val();

    alert("Value from #my_input: " +
input_value);
  });
});
``` | • When the "Set Value" button is clicked, it changes the value of the input field and updates the `id="output"` to indicate that the value has been set to a specific text. |

Getting element properties

- The `prop()` method returns properties and values of the selected elements.

Method	Description	Sample code	Code explanation
| `.prop()` | • Used to get properties of HTML elements. | ```
<input type="checkbox" id="my_checkbox" checked>
<button id="get_checkbox">Get Checkbox Value</button>
<p id="output">Checkbox value will be displayed here.</p>

$(document).ready(function() {
  $("#get_checkbox").click(function() {
    let is_checked =
$("#my_checkbox").prop("checked");

    $("#output").text("Checkbox value from #my_checkbox: " + is_checked);
  });
});
``` | • When the "Get Checkbox Value" button is clicked, it uses the `.prop()` function to get the `checked` property of the checkbox with `id="my_checkbox"`.<br>• It then displays a message in the output paragraph indicating whether the checkbox is checked or not, based on the value of the `checked` property. |

Setting element text and values

Method	Description	Sample code	Code explanation
| `.text()` | • Sets the text content of | ```
<button id="change_text">Change Text</button>
``` | • When the button is clicked, it changes the text content of |

	selected elements	```html<p id="text_element">This is some initial text.</p><script> $(document).ready(function() { $("#change_text").click(function() { $("#text_element").text("Text has been changed!"); }); });</script>```	the `<p>` element with `id="text_element"` to "Text has been changed!".
`.html()`	• Sets the content of selected elements (including HTML markup)	```html<button id="change_html">Change HTML</button><div id="html_element"> <p>This is some initial HTML content.</p></div><script> $(document).ready(function() { $("#change_html").click(function() { $("#html_element").html("<p>HTML content has been changed!</p>"); }); });</script>```	• When the button is clicked, it replaces the HTML content inside the `<div>` with new HTML content that includes bold text.
`.val()`	• Sets the value of form fields	```html<input type="text" id="text_input" placeholder="Enter text here"><button id="get_value">Get Value</button><button id="set_value">Set Value</button><div id="output">Value will be displayed here.</div><script> $(document).ready(function() { $("#get_value").click(function() {```	• When the "Get Value" button is clicked, it retrieves the current value from the input field and displays it in the `<div>` element with `id="output"`. • When the "Set Value" button is clicked, it changes

```
            var inputValue =
$("#text_input").val();
                $("#output").text("Current value:
" + inputValue);
            });

        $("#set_value").click(function() {
            $("#text_input").val("New Value
Set");
                $("#output").text("Value set to
'New Value Set'");
            });
    });
</script>
```

the value of the input field and updates the `id="output"` to indicate that the value has been set to a specific text.

Setting element properties

- The jQuery `prop()` method sets properties and values of the selected elements.

Method	Description	Sample code	Code explanation
`.prop()`	• Used to get or set properties of HTML elements.	```<label for="checkbox">Check this box:</label><input type="checkbox" id="checkbox"><div id="output">Checkbox is unchecked.</div><button id="toggle_checkbox">Toggle Checkbox</button><script> $(document).ready(function() {$("toggle_checkbox").click(function() { let checkbox = $("#checkbox"); let is_checked = checkbox.prop("checked"); if (is_checked) { checkbox.prop("checked", false);```	• When the button is clicked, it checks the current state of the checkbox using `.prop()`, toggles it, and updates the text in the "output" `<div>` to indicate whether the checkbox is checked or unchecked.

```
                $("#output").text("Checkbox
is unchecked.");
                } else {
                    checkbox.prop("checked",
true);
                $("#output").text("Checkbox
is checked.");
                }
            });
        });
</script>
```

Accessing the event trigger element

- In the case that you want to access the triggering element, you can use the `this` keyword.
- This is useful for when you have multiple elements with a similar selector.

Method	Description	Sample code	Code explanation

`this`	• Refers to the current DOM element that is the target of a jQuery method or event	```<button class="clickable_button">Click Me 1</button>```
		```<button class="clickable_button">Click Me 2</button>```
		```<button class="clickable_button">Click Me 3</button>```
		
		```<div id="output">Button clicked: None</div>```
		
		```$(document).ready(function () {```
		``` $(".clickable_button").click(function () {```
		``` let button_text = $(this).text();```
		
		``` $("#output").text("Button clicked: " + button_text);```
		``` });```
		```});```
			• When any of the "Click Me" buttons is clicked, the code uses the `this` keyword to identify the specific button that was clicked, retrieves its text content, and updates the text of the `<div>` with `id="output"` to indicate which button was clicked.
			• When you click a button, the text inside the `<div>` with `id="output"` is updated to display the text of

Manipulating Element Classes

- jQuery has methods that enable dynamic manipulation and querying of CSS classes on HTML elements.

Method	Description	Sample code	Code explanation

`.hasClass()`	- Checks if the selected element has a specific CSS class	```html
		<div id="my_div" class="initial-class">This is a div element.</div>
		<button id="check_class">Check Class</button>
		<button id="add_class">Add Class</button>
		<button id="rem_class">Remove Class</button>
		
		$(document).ready(function () {
		  $("#check_class").click(function () {
		    if ($("#my_div").hasClass("initial-class")) {
		      alert("#my_div has the 'initial-class' class.");
		    } else {
		      alert("#my_div doesn't have the 'initial-class' class.");
		    }
		  });
		
		  $("#add_class").click(function () {
		    $("#my_div").addClass("new-class");
		  });
		
		  $("#rem_class").click(function () {
		    $("#my_div").removeClass("initial-class");
		  });
		});```
			- When you click the "Check Class" button, it uses `.hasClass()` to check if the `<div>` with `id="my_div"` has `class="initial-class"` and displays an alert message accordingly.
			
			- Clicking the "Add Class" button adds the class `new-class` to the `<div>` element, and clicking the "Remove Class" button removes the `initial-class` from the `<div>`.

`.addClass()`	- Adds one or more CSS classes to the selected elements		
`.removeClass()`	- Removes one or more CSS classes from the selected elements		

| `.toggleClass()` | • Toggles the presence of one or more CSS classes on the selected elements, adding the class if it's absent or removing it if it's present | ```<style>``` ```.initial_class {``` ``` color: #333;``` ``` background-color: #f0f0f0;``` ``` padding: 10px;``` ```}``` ```.active_class {``` ``` color: #fff;``` ``` background-color: #007bff;``` ```}``` ```</style>``` ```<div id="my_div" class="initial_class">This is a div element.</div>``` ```<button id="toggle_class">Toggle Class</button>``` ```$(document).ready(function () {``` ``` $("#toggle_class").click(function () {``` ```$("#my_div").toggleClass("active_class");``` ``` });``` ```});``` | • When the "Toggle Class" button is clicked, uses jQuery's `.toggleClass()` method to toggle the class of the `<div>`.
 • When the class is toggled, it changes the text and background colors of the `<div>` to give it a different appearance. |

Additional Material

- **Learn more**
  - W3Schools