School of Infocomm

C219 Front-end Web Development

Lesson 7

JavaScript and jQuery

Recap

Lesson 6

What is JavaScript?

JavaScript is a programming language that allows you to implement complex features on web pages. It is the standard programming language for the web.

JavaScript can update and change both HTML and CSS and is commonly used to calculate, manipulate and validate data.

Over 97% of websites use it for client-side web page behaviour.

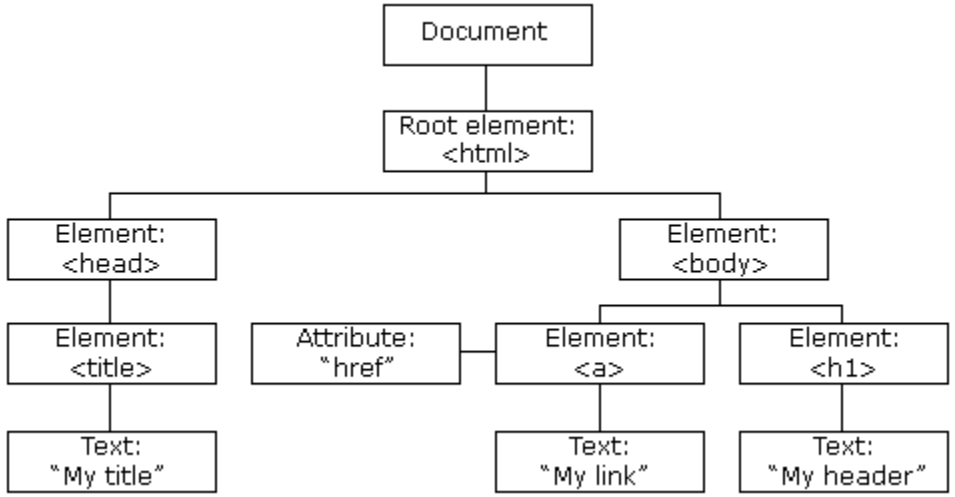
HTML DOM

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.

In other words, the HTML DOM is a standard for how to get, change, add, or delete HTML elements.

The HTML DOM model is constructed as a Tree of Objects.

The HTML DOM Tree of Objects Example



DOM and JavaScript

With the object model, JavaScript gets all the power it needs to create dynamic HTML.

JavaScript can use the DOM to:

- Change the HTML elements and attributes in the page
- Change the CSS styles in the page
- Remove existing HTML elements and attributes
- Add new HTML elements and attributes
- React to HTML events in the page
- Create new HTML events in the page

Using JavaScript

JavaScript code is inserted between the `<script></script>` tags in a web page, usually right before the `</body>` tag. To use an external script, put the name of the script file in the `src` attribute of a `<script>` tag.

```
<body>
    <p id="demo">Hello.</p>
    <button type="button" onclick="myFunction()">Click Me</button>

    <script>
        function myFunction() {
            document.getElementById("demo").innerHTML = "Bye!";
        }
    </script>
</body>
```

JavaScript Syntax

```
<button type="button" onclick="myFunction()">Click Me</button>
    <div id="demo"></div>

    <script>
        function myFunction() {
            var x = "Hello";
            let y = 2;
            const z = 3;

            document.getElementById("demo").innerHTML = x + " " + (y + z);
        }
    </script>
```

An event attribute which executes a script when the button is clicked

```
function name(parameter1, parameter2)
{
    // code to be executed
}
```

Var can be updated and re-declared
Let can be updated but not re-declared
Const cannot be updated or re-declared

getElementById() returns the element that has the ID attribute with the specified value

The innerHTML property sets or gets the HTML content of an element

HTML Events

HTML has the ability to let events trigger actions in a browser from user interactions. Here are some common events used:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeyup	The user releases a keyboard key
onload	The browser has finished loading the page

Reference: https://www.w3schools.com/jsref/dom_obj_event.asp

Accessing HTML Elements

Here are some methods you can use to access HTML elements and manipulate them.

Finding HTML Elements

Method	Description	Example
document.getElementById(id)	Find an element by element id	document.getElementById("title")
document.getElementsByClassName(class)	Find elements by class name	document.getElementsByClassName("a")
document.querySelector(selector)	Find elements by CSS selector	document.querySelector(":checked")

Changing HTML Elements

Property	Description
element.innerHTML = new html content	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.style.property = new style	Change the style of an HTML element

Reference: https://www.w3schools.com/js/js_htmldom_elements.asp

Accessing HTML Elements

The **getElementsByClassName** method returns an array-like object of all child elements which have all of the given class name(s).

To get the first element with a class of **demo4**, we need to add **[0]**.

```
<div class="demo4" onclick="myFunction4()">Click me!</div>

    <script>
        function myFunction4() {
            document.getElementsByClassName("demo4")[0].innerHTML = "Thanks!";
        }
    </script>
```

Exercise 1

Create a form that would display the user inputs in the same page when submitted.

Requirements:

- Two text fields - Name and Email
- Two radio buttons for Gender - Male and Female
- Submit button
- When the Submit button is clicked, display all form data below
- Use exercise1.html in the Resources folder

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.

jQuery greatly simplifies JavaScript programming.

https://jquery.com

Using jQuery

The fastest and easiest way to include jQuery in your project is by using a content delivery network (CDN).

Insert this before **</body>:**

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

ⓘ All your JavaScript should be written below the jQuery link.

Reference: https://jquery.com/download

jQuery Selectors

jQuery selectors allow you to select and manipulate HTML elements.

jQuery selectors are used to find HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It is based on the existing CSS Selectors, and in addition, it has some of its own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: **$()**.

```
$("p") //selects all <p> elements
$(this) //selects the current element
$(".test") //selects all elements with class="test"
$(":button") //selects all elements with type="button"
```

View the full list of jQuery selectors here and try them out.

jQuery Events

jQuery events are methods used to register behaviours that take effect when the user interacts with the browser, and to further manipulate those registered behaviours.

Examples of events are, clicking on an element, selecting a radio button and moving the mouse over an element.

```
$(".box").click(function() {
  alert("You just clicked the box.");
});
```

View the full list of jQuery events here and try them out.

jQuery Effects

The jQuery library provides several techniques for adding effects or animation to a web page. These include simple, standard animations that are frequently used, and the ability to craft sophisticated custom effects.

```
$(".box").click(function() {
  $(this).fadeOut();
});

$(".box").click(function() {
  $(this).fadeOut().fadeIn(5000); //you can chain multiple effects
});
```

View the full list of jQuery effects here and try them out.

jQuery HTML and CSS Methods

You can easily manipulate HTML elements using the jQuery HTML and CSS methods. These methods allow you to get, set, copy, add or remove content, elements and attributes.

```
$(".box").click(function () {
  $("p")
    .html("Hello")
    .css({
      "font-weight": "bold",
      "font-style": "italic"
  });
});
```

View the full list of jQuery HTML and CSS methods here and try them out.

jQuery and JavaScript

Since jQuery is based on JavaScript, you can include JavaScript operators, declarations and conditional statements within jQuery.

```
let count = 0;

$(".box").click(function () {
  count++;
  $("p").html("Click counter: " + count);

  if (count == 10) {
    alert("Congrats, you clicked me 10 times.");
  }
});
```

jQuery UI

jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery library. To use jQuery UI, you need to include the JS and CSS files here.

```
//jQuery UI autocomplete widget
<input type="text" class="cars">

<script>
  var tags = ["Audi", "Tesla", "Toyota", "Porsche", "Volvo"];

  $(".cars").autocomplete({
    source: tags
  });
</script>
```

View all jQuery UI features here.

Exercise 2

Using only jQuery, code the same form submit feature in Exercise 1. In addition, code the enhancements below.

Requirements:

- Fade in the form data on submit
- Using jQuery CSS methods, change the font to Arial, add a border surrounding the form data, and add a background colour.
- Slide up the form data on mouseover

Exercise 3

Complete the jQuery and Bootstrap exercises in exercise3.html.

Requirements:

- Show a tooltip that says “This is a button” when you mouse over the button
- Using the jQuery toggleClass method, toggle between blue and green for the circle when clicked
- Using the jQuery clone and appendTo methods, duplicate the square into the grey area when you click Add.

Deliverables

Individual Submission:

- Exercise 1
- Exercise 2
- Exercise 3
- L07 Problem Statement

Submit all deliverables by 2359 today

Others:

- Quiz 3:30 to 4:00pm