JavaScript
Intro for Web Development
Lab 4

1 Executing a JavaScript program

There are a multitude of ways to execute a JavaScript program.

1. Using the Node.js runtime environment to directly execute a JavaScript program
2. Including JavaScript into a HTML document
3. Running JavaScript in console tab of Dev Tools
4. Using an online JavaScript playground
5. Using a framework that is based on JavaScript (e.g. Angular, React, Vue)

We have already seen the first approach in the previous lab sessions. We will look at some of the other approaches here.

1.1 Including JavaScript into a HTML document

There are 3 main ways to add JavaScript code to a web page:
https://www.tutorialrepublic.com/javascript-tutorial/javascript-get-started.php

The most common approach is to place the JavaScript code in a separate file with the .js extension and then load it within the page through the `src` attribute of the `<script>` tag.

Files to use:
```
demo-embed-javascript.html
demo-include-javascript.html
hi-there.js
demo-inline-javascript.html
```

For JavaScript referenced from within a HTML document that is loaded in the browser, there are 2 ways to perform simple text output for debugging purposes:

- Using `alert('message to display');` method of the Window global object which causes an alert box to appear with the message.
- Using `console.log('message to display');` where the message appears in the console tab of Dev tools

1.2 Placement of JavaScript tag

Recommendation for including in <head> tag along with `defer` and `async` attributes properly
https://stackoverflow.com/questions/436411/where-should-i-put-script-tags-in-html-markup

https://flaviocopes.com/javascript-async-defer/#just-tell-me-the-best-way

1.3 Typing JavaScript code directly into the console tab

https://developer.chrome.com/docs/devtools/console/javascript/

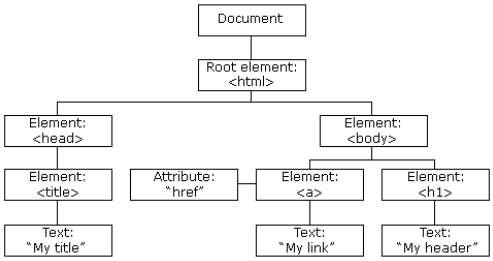
https://www.digitalocean.com/community/tutorials/how-to-use-the-javascript-developer-console

1.4 Using an online JavaScript playground

https://linuxhint.com/top-five-javascript-playgrounds/

2 JavaScript Document Object Model (DOM)

When a HTML document is loaded into the browser, it creates an internal representation of that document in memory known as a Document Object Model (DOM) tree. All the various items in the HTML (the elements, attributes and content between tags) are represented as objects / nodes in the this tree. The DOM is a hierarchical tree starting from the root element <html> extending all the way down to the leaf nodes, which are typically contents or attributes of various elements.



The DOM exposes a cross platform programming interface (API) that allows a program (typically written in JavaScript) to modify the relevant nodes in the tree in a specific way (for e.g. changing their properties, removing a node, adding new nodes, etc), thereby adding interactivity to the web page. This can be achieved using either basic JavaScript (vanilla JavaScript) or a popular JavaScript library or framework such as React or Angular.

https://www.javascripttutorial.net/javascript-dom/document-object-model-in-javascript/

https://www.w3schools.com/js/js_htmldom_document.asp

3 Selecting elements

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

3.1 getElementById

https://www.javascripttutorial.net/javascript-dom/javascript-getelementbyid/

Files to use: `demo-getElementById.html`

3.2 querySelector, querySelectorAll, Nodelist

https://www.javascripttutorial.net/javascript-dom/javascript-queryselector/

https://www.w3schools.com/jsref/met_document_queryselectorall.asp

The argument passed to the querySelector / querySelectorAll method is any one of the valid CSS selectors:

https://www.tutorialrepublic.com/css-tutorial/css-selectors.php
https://code.tutsplus.com/tutorials/the-30-css-selectors-you-must-memorize--net-16048

The value returned from a call to querySelectorAll is a NodeList
https://www.w3schools.com/jsref/dom_obj_html_nodelist.asp

Difference between NodeList and a HTMLCollection
https://www.w3schools.com/jsref/met_document_queryselectorall.asp

Each object in the NodeList is a HTML Element object. A summary of the properties and methods available on each object is shown here:
https://www.w3schools.com/jsref/dom_obj_all.asp

The complete list of properties and methods for the HTMLElement object can be found at the official MDN documentation:
https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement
https://developer.mozilla.org/en-US/docs/Web/API/Element

Files to use: `demo-querySelectorAll.html`

3.3 getElementsByName, getElementsByTagName, getElementsByClassName()

https://www.javascripttutorial.net/javascript-dom/javascript-getelementsbyname/
https://www.javascripttutorial.net/javascript-dom/javascript-getelementsbytagname/
https://www.javascripttutorial.net/javascript-dom/javascript-getelementsbyclassname/

Files to use: `demo-getElementsByName.html`

4 Traversing Elements

https://www.javascripttutorial.net/javascript-dom/javascript-siblings/

https://www.javascripttutorial.net/javascript-dom/javascript-get-parent-element-parentnode/

https://www.javascripttutorial.net/javascript-dom/javascript-get-child-element/

Files to use: `demo-traverseElements.html`

5 Working with attributes

https://www.javascripttutorial.net/javascript-dom/html-attributes-dom-object-properties/

https://www.w3schools.com/html/html_form_attributes.asp
https://www.tutorialrepublic.com/html-tutorial/html-attributes.php

https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement
https://www.w3schools.com/jsref/dom_obj_text.asp

https://www.javascripttutorial.net/javascript-dom/javascript-setattribute/
https://www.javascripttutorial.net/javascript-dom/javascript-getattribute/

Files to use: `demo-objectProperties.html`

6 Manipulating element's styles

https://www.javascripttutorial.net/javascript-dom/javascript-style/

https://www.w3schools.com/cssref/
https://www.tutorialrepublic.com/css-reference/css3-properties.php

Files to use: `demo-changeStyles.html`

7 Manipulating elements

https://www.w3schools.com/js/js_htmldom_html.asp

7.1 createElement, appendChild, innerHTML, textContent, innerText

https://www.javascripttutorial.net/javascript-dom/javascript-createelement/

https://www.javascripttutorial.net/javascript-dom/javascript-appendchild/

https://www.javascripttutorial.net/javascript-dom/javascript-textcontent/

https://www.javascripttutorial.net/javascript-dom/javascript-innerhtml/

Key differences between using innerHTML and createElement when creating new HTML content
https://www.javascripttutorial.net/javascript-dom/javascript-innerhtml-vs-createelement/

Files to use: `demo-create-inner.html`

7.2 insertBefore, insertAfter, append, prepend

https://www.javascripttutorial.net/javascript-dom/javascript-insertbefore/

https://www.javascripttutorial.net/javascript-dom/javascript-insertafter/

https://www.javascripttutorial.net/javascript-dom/javascript-append/

https://www.javascripttutorial.net/javascript-dom/javascript-prepend/

Files to use: `demo-insert.html`

7.3 replaceChild, cloneNode, removeChild

https://www.javascripttutorial.net/javascript-dom/javascript-replacechild/

https://www.javascripttutorial.net/javascript-dom/javascript-clonenode/

https://www.javascripttutorial.net/javascript-dom/javascript-removechild/

Files to use: `demo-replace-remove.html`

8 Working with events

https://www.javascripttutorial.net/javascript-dom/javascript-events/

https://www.javascripttutorial.net/javascript-dom/handling-events-in-javascript/

Files to use: `basic-event-handling.html`

8.1 Mouse events

https://www.javascripttutorial.net/javascript-dom/javascript-mouse-events/

For difference between the different location-related coordinates accessible from the event object:
https://stackoverflow.com/questions/6073505/what-is-the-difference-between-screenx-y-clientx-y-and-pagex-y
https://www.codetd.com/en/article/12514516

General outline for manipulating the DOM

- Select an event to listen for and an element on which that event can occur on
- Register an event handler to perform some action on the DOM when the event happens. Actions can involve:
  a) Selecting a particular element (Section 2. Selecting Elements)
  b) Manipulating the element in some way (Section 4. Manipulating elements)
  c) Changing the attributes of the element (Section 5. Working with Attributes)
  d) Changing the styling of the element (Section 6. Manipulating the Element's styles)

Files to use: `basic-mouse-handling.html`

8.2 Sample DOM manipulation with mouse events

Demonstration of more complex DOM manipulation based on mouse events

Files to use: `more-mouse-handling.html`

8.3 Keyboard events

https://www.javascripttutorial.net/javascript-dom/javascript-keyboard-events/

Files to use:
`basic-keyboard-handling.html`
`more-keyboard-handling.html`

A more detailed example that demonstrate extensive DOM manipulation in conjunction with processing of keyboard events:
Files to use:
`work-with-lists.html`
`lists.js`