Javascipt

* Associates a program with the webpage
* A program/script is downloaded and executed in the browser.

Why use JavaScript?

* Other web technologies like html and css are not very interactive instead they are passive.

Associating javascript:

a. Externally linked (can be placed anywhere)

e.g. <script type = “text/javascript” src = “script.js”> </script>

b. Embedded within html

<script>

console.log (‘embedded script…’)

</script>

c. Inline script

e.g.

<button onclick = console.log (“inline script…”); )

Click me </button>

Code defined outside of a function = global or top level

Code defined inside of a function = function body

Attributes:

1. Defer Attribute – do not execute until document has been rendered.

- fetching does not block the rendering.

2. Async Attribute – fetch the script, render the page, then execute even the if rendering is not yet done.

* Some browsers do not support defer and async, in this case, placing will be the key
* For browsers with no scripting support or with scripting support disabled:

<noscript> No Scripting Support. </noscript>

Some browsers disable javascript for security measure purposes.

Window Object – represents a browser environment

\*to manipulate a document:

window.document or document

\*to return a reference by element:

document.getElementById(‘x’)

\*to set an attribute:

h.setAttribute(‘title’)

\*to manipulate the styling:

h.style = “color: blue”

\*to retrieve all elements of a familiar style

[p, p, p, p]

querySelector – argument will be a CSS selector

- if there are more than one match, it will choose the first.

querySelector All – will return all matches except pseudo elements

e.g. document.querySelector All (‘body p’)

Document Object Model API

<p> hello </p>

p – element

Node

hello – text

document.childNodes

document.children

document.hasChildNode

document.body

document.head.childNodes – all elements

document.head.children – specific elements

document.body.firstChild – can be a text node

document.body.firstElementChild

document.body.firstChild.nextSibling

document.body.firstChild.previousSibling