The Document Object Model

- The Document Object Model (DOM) allows you to access elements of a web page and enable interaction with the page by adding and removing elements, changing the order, content and attributes of elements, and even altering how they are styled.
- The DOM provides several methods that allow us to access any element on a

  Page:
  - Document.body returns the body element of a web page.
  - Document.images returns a node list of all the images in the document.
  - Document.links returns a node list of all the <a> elements and <area> elements that have an href attribute.
  - Document.anchors returns a node list of all the <a> elements that have a name attribute.
  - Document.forms returns a node list of all the forms in the document.
  - const h1 = document.getElementById('title');
  - const listItems = document.getElementsByTagName('li');
  - const heroes = document.getElementsByClassName('hero');
  - heroes.length;
- The document.querySelector() method allows you to use CSS notation to find the first element in the document that matches that matches a CSS selector provided as an argument.
- The document.querySelectorAll() method also uses CSS notation but returns a node list of *all* the elements in the document that match the CSS query selector.
- The DOM has numerous getter and setter methods that can be used to view, add, remove or modify the value of any of these attributes:
  wonderWoman.getAttribute('class');
  wonderWoman.setAttribute('class', 'villain');

  Classes Of An Element

- **The** className **Property**: There is also a className property that allows the class of an element to be set directly.

- The classList Property: a list of all the classes an element has.
    - The add method can be used to add a class to an element without overwriting any
      classes that already exist.
    - The remove method will remove a specific class from an element.
    - The contains method will check to see if an element has a particular class

- createElement(): method that takes a tag name as a parameter and returns that element.

- document.createTextNode(): text node can be created using this function.

- innerHTML: The innerHTML element property was standardized as part of the HTML5.

- The Document Object Model is a way of representing a page of HTML as a tree
of nodes.

- The document.getElementById(), document.getElementsByClassName(), document.getElementsByTagNames() and document.querySelector() can be used to access elements on a page.

- The parentNode(), previousSibling(), nextSibling(), childNodes() and children() methods can be used to navigate around the DOM tree.

- An element's attributes can be accessed using the getAttribute() method, and updated using the setAttribute() method.

- The createElement() and createTextNode() methods can be used to create dynamic markup on the fly.

- Markup can be added to the page using the appendChild() and insertBefore() methods.

- Elements can be replaced using the replaceChild() method, and removed using the removeChild() method.

- The innerHTML property can be used to insert raw HTML directly into the DOM.
- The CSS properties of an element can be changed by accessing the style property.