Module 4 Day 3

Document Object Model

What makes an application?

- Program Data
  - ✓ Variables & .NET Data Types
  - ✓ Arrays
  - ✓ More Collections (list, dictionary, stack, queue)
  - ✓ Classes and objects (OOP)
- Program Logic
  - ✓ Statements and expressions
  - ✓ Conditional logic (if)
  - ✓ Repeating logic (for, foreach, do, while)
  - ✓ Methods (functions / procedures)
  - ✓ Classes and objects (OOP)
  - ✓ Frameworks (MVC)
- Input / Output
  - User
    - ✓ Console read / write
    - ✓ HTML / CSS
    - ❖ Front-end frameworks (HTML / CSS / JavaScript)
  - Storage
    - ✓ File I/O
    - ✓ Relational database
    - ❑ APIs

Document Object Model (DOM)

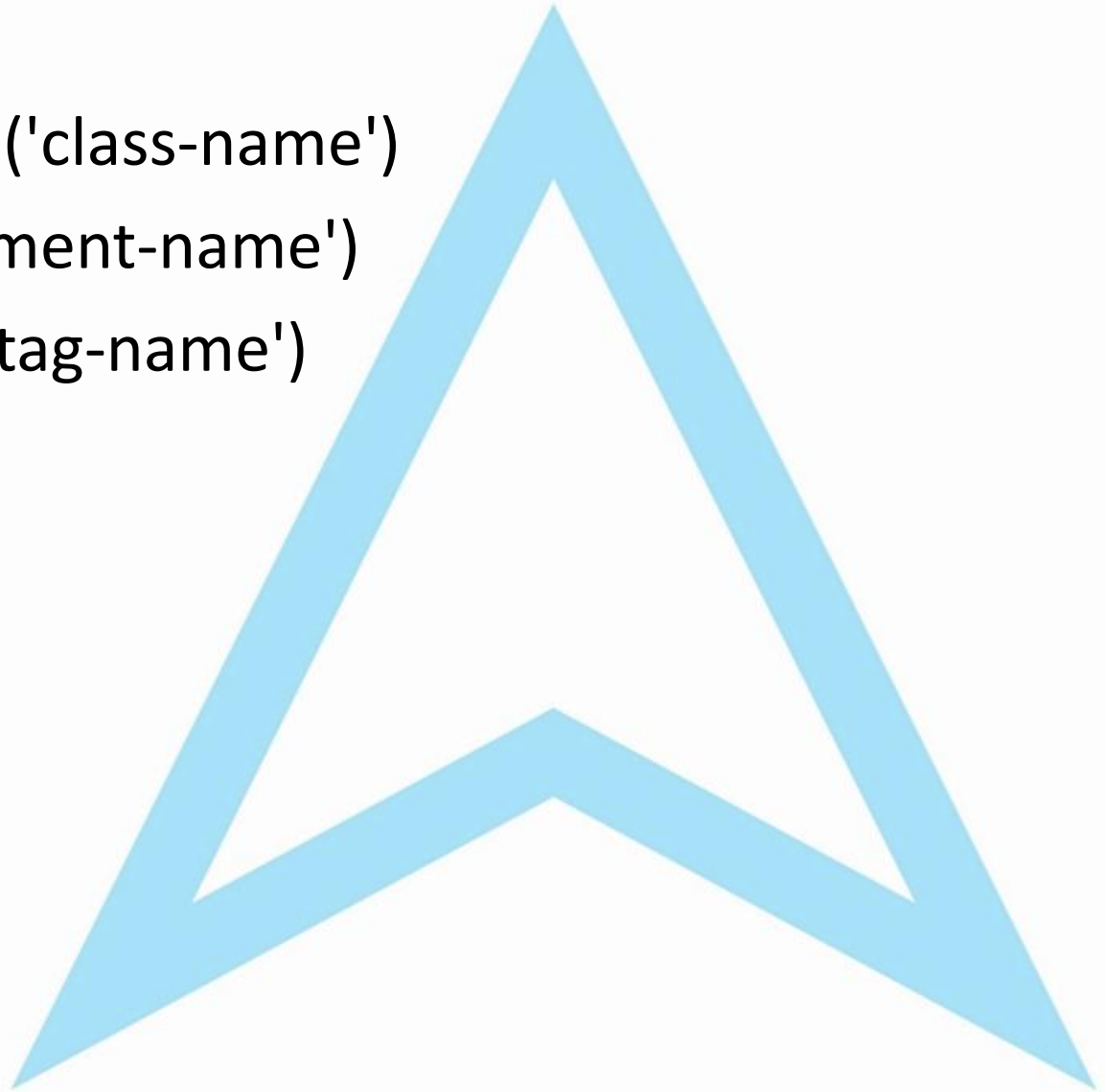
- The browser's internal representation of the HTML document
- CSS and JS target the DOM (not the original HTML)!
- *View Page Source* shows the original HTML
- F12 -> Elements (Inspect) shows the DOM
- *document* represents the HTML page (root object of the HTML doc)

DOM Get Methods

- document.getElementById('element-id')
  - Most efficient
  - Returns first (should be only) element matching id
  - Called only through document
- *element*.querySelector('css-selector-string')
  - Returns first element matching the selector
  - Called through any element to narrow the search
- *element*.querySelectorAll('css-selector-string')
  - Returns NodeList containing all matching elements
  - Called through any element to narrow the search

Other Get Methods

- element.getElementsByClassName('class-name')
- element.getElementsByName('element-name')
- element.getElementsByTagName('tag-name')
- These return 'live' lists



Modifying Elements – Properties

Property	Description
innerText / innerHTML	Gets or sets the text inside the node.innerText is safe; innerHTML is susceptible to injection attack.
value	Gets or sets the value of most input elements
checked	Gets or sets the Boolean state of a checkbox
classList	Gets a collection of the classes applied to the element. Use .add() or .remove() to change the classes on an element.
children / childNodes	Gets a collection of this element's child elements, or child nodes, respectively. children is *usually* what you want; childNodes include text, comments and other nodes that you are usually not interested in.
parentNode	Gets the element to which this element belongs (is in the parent's children collection)
nextElementSibling / previousElementSibling	Gets to the next/previous element with the same parent

Creating New Elements

```
// Create the element
let ele = document.createElement('tag-name');
// Set properties on the element
ele.id = 'element-id';
ele.innerText = 'text';       // etc.
// Find another relevant element in the tree
let parent = document.getElementById('parent-id');
// Insert the element into the DOM
parent.insertAdjacentElement('afterbegin', ele);
```

Removing Elements

```
// Find the element
let ele = document.querySelector('selector-string');

// Remove the element from its parent
ele.parentNode.removeChild(ele);
```