 # DOM APIs

Recall the basics of JavaScript that we have already covered. They are mostly very general and abstract. They do not specifically reference the HTML elements on a web page.

JavaScript:
- Data types
  - Strings
    - "Hello world"
  - Numbers
    - 5
    - -12
    - 0
    - 0.125
  - Booleans
    - true
    - false
  - Arrays
    - .map()
    - .forEach()
    - .find()
    - .length
  - Objects
  - Functions
    - Function declarations
    - Arrow functions
  - Classes
  - undefined
  - null
  - NaN
  - Infinity
- Built-in global objects
  - Math
    - .random()
    - .round(num)
    - .ceil(num)
    - .floor(num)
    - .max(num1, num2)
    - .min(num1, num2)
    - .abs(num)
    - .sqrt(num)
    - .pow(num, exponent)
  - Date

```
                - .now()
            - Promise
                - new Promise(callbackFunction)
                    - (resolve, reject)
                - promise.then()
                - promise.catch()
                - Promise.all()
    - Control flow
            - If/Else
            - Loops
```

What is the DOM API?

- Document Object Model Application Programming Interface
- Not part of JavaScript but gets added as additional functionality by web browser
- Provides access to the current state of the rendered tree of HTML elements as well as other properties of the user's browser

What is `window`?

- The global object
- Provides access to various browser-specific functionalities including the DOM API
- Represents the browser at a high level, including the browser application's resizable/scrollable container
- Does not need to specified, but helps for clarity

What properties exist on `window`?

- .innerWidth
- .innerHeight
- .location
    - .href
- .console
    - .log("message")
- .history
    - .back()
- .navigator
- .scrollX
- .scrollY

What methods exist on `window`?

- .addEventListener("scroll")
- .addEventListener("resize")
- .removeEventListener("eventType", callbackFunction)
- .alert("message")
- .setTimeout(callbackFunction, interval)
- .requestAnimationFrame(callbackFunction)
- .fetch(url, options)
- .scrollTo(x, y)
- .matchMedia("css media query")
    - .addEventListener("change", callbackFunction)

What is `window.document`?

- Represents a web page loaded in the browser
    - Web pages used to be called "documents" because the early web was primarily designed to display research papers
- Holds the element tree

What methods exist on `window.document`?

- .createElement("element-name")
- .querySelector("css selector")
- .querySelectorAll("css selector")

What properties exist on `document`?

- .documentElement => <html> Element
- .activeElement => Element that is currently focused
- .cookie => get and set cookies that will be sent to a server

How is an HTML element represented?

- An element in the DOM uses the Element interface

What methods exist on the HTML Element interface?

- .querySelector("css selector")
- .querySelectorAll("css selector")
- .closest("css selector")
- .matches("css selector")
- .contains(element)
- .append(element)
- .prepend(element)
- .remove()

- .getAttribute("attribute-name")
- .setAttribute("attribute-name", "value")
- .addEventListener("eventType", callbackFunction)
    - "click"
    - "change"
    - "submit"
- .removeEventListener("eventType", callbackFunction)
- .getBoundingClientRect()
    - .height
    - .width
    - .top
    - .left
- .scrollIntoView()
- .focus()

What properties exist on the HTML Element interface?

- .innerText
- .children => HTMLCollection
- .parentElement => Element
- .nextElementSibling => Element
- .style
    - .setProperty("cssProperty", "cssValue")
    - .getPropertyValue("cssProperty")
- .classList
    - .contains("className") => Boolean
    - .add("className")
    - .remove("className")
    - .toggle("className")

What are live HTMLCollections and NodeLists?

- Array-like representations of the DOM
- Update as the DOM changes
- Wrap in `Array.from()` to get a real array that won't change

What is the `event` argument in an event listener callback function?

- Provides properties and methods for the event that is being fired inside the callback function of .addEventListener()

What methods exist on the Event interface?

- .preventDefault()

- .stopPropagation()

What properties exist on the Event interface?

- .currentTarget
    - The element that the event listener was added to.
- .target
    - The element that is currently responding to the event listener callback
    - May be firing on the element that the event listener was added to, or may be firing on one of its ancestors or descendants.
- Other properties depend on the type of the Event

What is the difference between event bubbling and capturing?

- Some events will only fire on one element (the .currentTarget)
- Others "bubble" meaning the callback is fired on all ancestors and descendants of the .currentTarget element
- There are two paradigms for propagating events:
- Bubble mode (Bottom -> Top)
    - The default modea
    - Fires on the innermost child element first
    - Then fires on every ancestor element
- Capture mode (Top -> Bottom)
    - Specified as an extra argument in .addEventListener()
    - Fires on the top-most ancestor element first
    - Then fires on every descendant element
- To only fire on a specific element, guard by checking if event.target === event.currentTarget
- To prevent further bubbling/capturing at any point, call event.stopPropagation()

What is event delegation?

- You could add an event listener to many objects that all exist inside the same ancestor element...
- Or you could add one event listener to the common ancestor element, knowing the event will bubble and its callback function will fire on all the children.
- To only respond to specific children, guard by checking if event.target.matches("css selector")

What is the FormData API?

- new FormData(formElement)
    - .entries() provides an array of arrays with key/values

What is the interface for converting between an object and an array?

- Use the entries API
    - { key1: value1, key2, value2 }
    - [[key1, value1], [key2, value2]]
- object.entries()
- Object.fromEntries(array)