More JavaScript and the DOM

Day 2

JavaScript Continued

- JavaScript is technically just an implementation of the ECMAScript Language Standard - ECMA-262
    - ECMA is ***European*** Computer Manufacturer's Assoc
    - Just like the C is an implementation of ISO (International Standards Organization) 9899:<version>
- ES is often used as the abbreviation for **E**CMA**S**cript
- You can think of JS and ES as roughly the same thing, but since ES gets *versioned* (ES5, ES6, ES2018, ES.Next) you'll often see it used when some *code* uses some new feature of a version.

ECMAScript Versions

- ES3 - 1999 - The Dark Ages
- ES5 - 2009 - Adds strict mode, JSON support
- ES6/ES2015 - The Renaissance - Adds classes, let/const declarations, modules, block scope, arrow functions, promises, destructuring assignment, Map, Set, yield, iterators, for..of, and more…
- ES7/ES2016 - Exponentiation operator, Array#includes()
- ES8/ES2017 - async/await, atomics and shared memory (for concurrency features with worker threads)

The Problem with Cutting Edge Language Features

- ES6 was the significant, feature-rich milestone in June 2015
  - Sadly, browsers implement features at varying rates
    - Arrow functions came to Edge July 2015, Chrome September 2015, Internet Explorer *Never*.
  - If a browser doesn’t support a feature your code depends on, your user’s experience is broken for that user.
- ES Compatibility: https://kangax.github.io/compat-table/es6/
- CanIUse: https://caniuse.com/#feat=arrow-functions

The Rise of the Transpiler

- Navigating browser compatibility was really terrible
  - Either you break your project for some % of I.E. users
  - Or, you stick with the old feature set of JavaScript ☠️
- Computer scientists had a better idea:
  - Write a compiler that translates comfy, nice, modern code into older, simpler, more compatible code.
  - Compilers translating source code to source code, or a “source-to-source” compiler, are called transpilers.

Popular Transpilers

- 2009 - Google Closure - Command-line program written in Java to check and compile JS to simpler, more compact JS
- 2010/2011 - RequireJS/Browserify - Multi-file modules for growing JavaScript projects.
- 2012 - TypeScript - Adds static type annotations and type checking.
- 2013 - Facebook React's JSX - Extended JavaScript language to mix-in HTML tags.
- 2015 - Babel - Transpile ES6(!) to older versions of ES
  - Demo: https://babeljs.io/repl

Document Object Model

HTML Describes an Object Tree

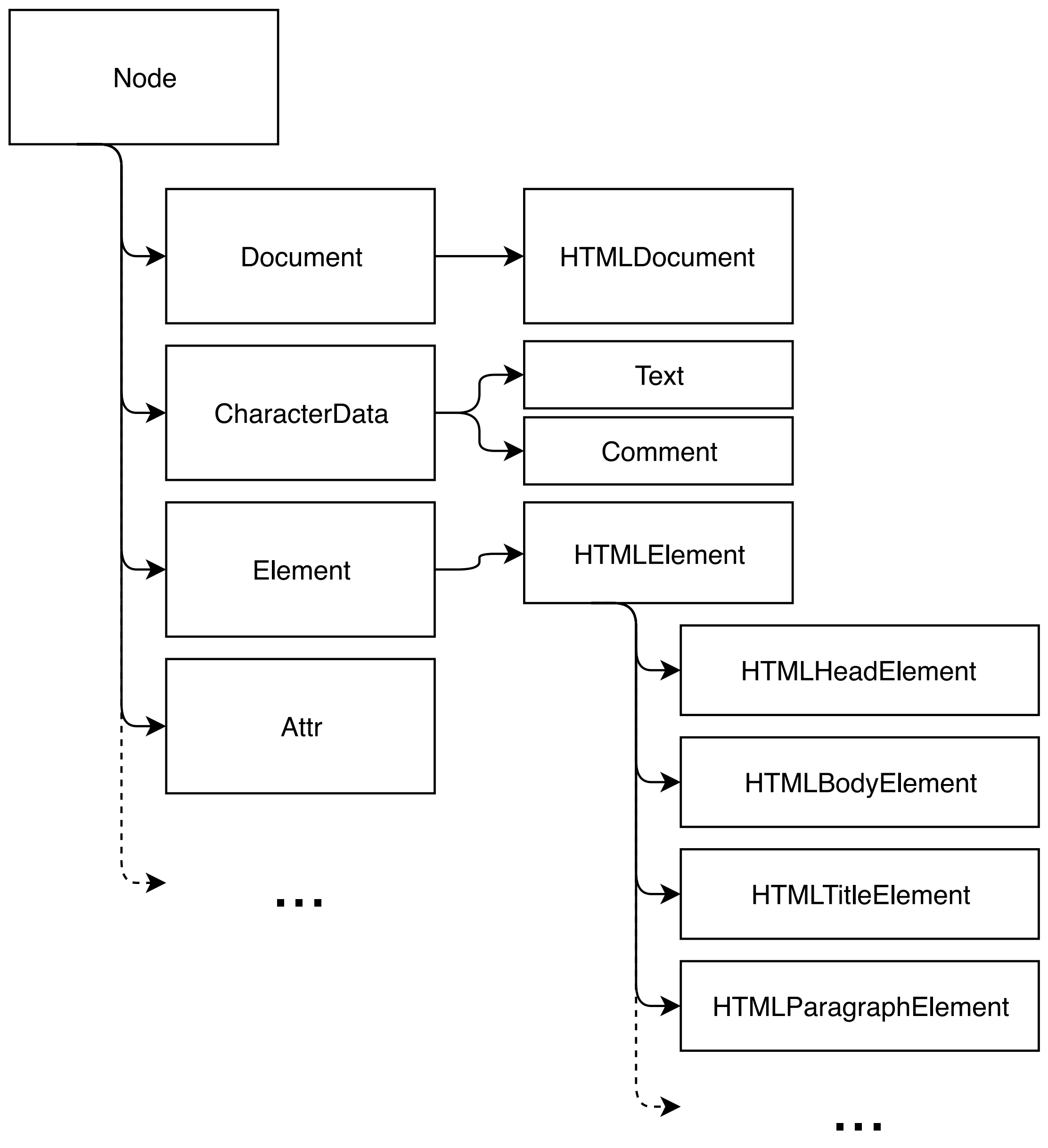
- The **Document Object Model (DOM)** specifies the structure, classes, and interfaces of objects in the tree representing a page.

- Each ***Tag, Attribute, Text Block, Comment***, and so on, describes a type of **Node** object.
  - Node is the superclass of every node in the DOM

- When a browser downloads HTML for a page, it parses the HTML text and constructs an equivalent tree of objects in memory.

DOM Class Hierarchy

- DOM classes extend from a common superclass: **Node**
- Of course, each subclass can introduce properties and methods specific to it.
- Most of the DOM work you'll do will be with objects whose types descend from HTMLElement.



Exploring DOM's *Node*

- childNodes
- nodeName
- nodeType
- appendChild
- removeChild
- For more reference: https://javascript.info/dom-navigation

Challenge

- Count the number of Nodes (using childNodes) on ESPN.com
- Try: Opening ESPN.com in your browser.

Exploring DOM’s *Element*

- attributes
- className
- id
- querySelector()
- querySelectorAll()
- setAttribute()

Exploring DOM's *HTMLElement*

- style
- focus()
- blur()
- click()

Exploring *EventTarget* Interface

- Many DOM classes implement the *EventTarget* Interface including…
    - Element
    - Document
    - Window