WEB COMPONENTS

EVERYWHERE

What are web components?

are a set of web platform APIs that allow you to create new custom, reusable, encapsulated HTML tags to use in web pages and web apps.

Custom components and widgets build on the Web Component standards, will work across modern browsers

can be used with any JavaScript library or framework that works with HTML

WEB COMPONENTS FEATURES:

- **HTML Templates** - an HTML fragment is not rendered, but stored until it is instantiated via JavaScript.
- **Shadow DOM** - Encapsulated DOM and styling, with composition.
- **Custom Elements** - APIs to define new HTML elements.
- ~~**HTML Imports** - Declarative methods of importing HTML documents into other documents.~~ (Replaced by **ES6 Imports**).

HTML templates

The <template> and <slot> elements enable you to write markup templates that are not displayed in the rendered page.

These can then be reused multiple times as the basis of a custom element's structure.

Shadow DOM

A set of JavaScript APIs for attaching an encapsulated "shadow" DOM tree to an element — which is rendered separately from the main document DOM — and controlling associated functionality.

In this way, you can keep an element's features private

so they can be scripted and styled without the fear of collision with other parts of the document

Shadow DOM allows hidden DOM trees to be attached to elements in the regular DOM tree — this shadow DOM tree starts with a shadow root, underneath which can be attached to any elements you want, in the same way as the normal DOM.

Shadow DOM

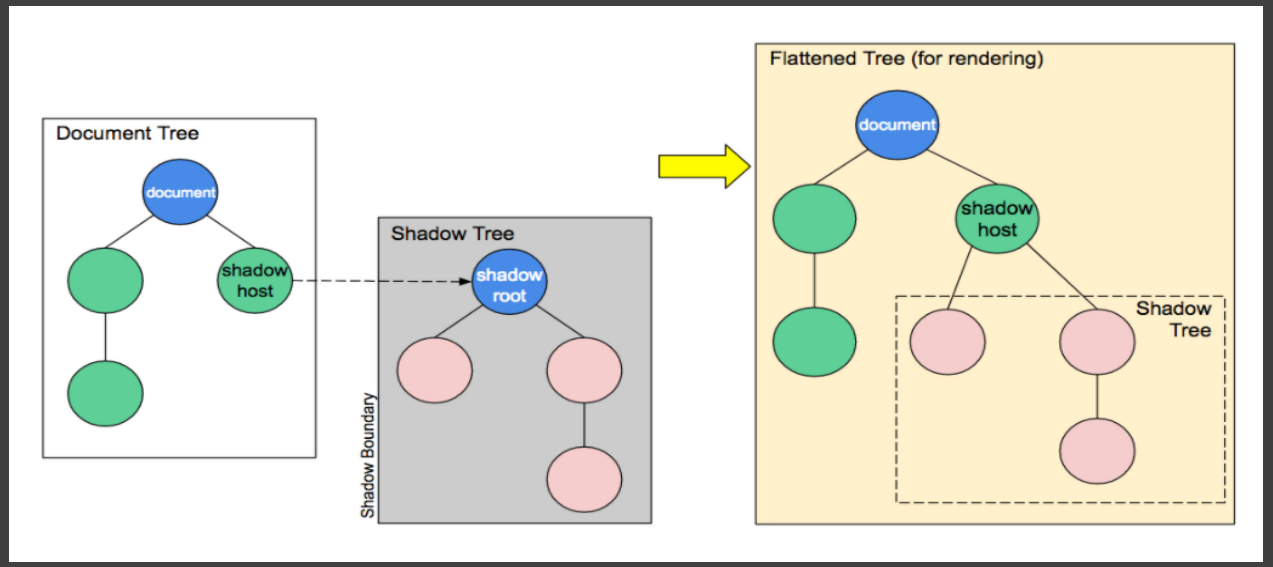
There are some bits of shadow DOM terminology to be aware of:

Shadow host: The regular DOM node that the shadow DOM is attached to.

Shadow tree: The DOM tree inside the shadow DOM.

Shadow boundary: the place where the shadow DOM ends, and the regular DOM begins.

Shadow root: The root node of the shadow tree.



Shadow DOM

You can attach a shadow root to any element using the Element.attachShadow() method.

This takes as its parameter an options object that contains one option – mode - with a value of open or closed

```
let shadow = elementRef.attachShadow({mode: 'open'});
let shadow = elementRef.attachShadow({mode: 'closed'});
```

open - can access the shadow DOM using JavaScript written in the main page context, for example using the Element.shadowRoot property

closed - you won't be able to access the shadow DOM from the outside

Approach for implementing a web component

Create a class in which you specify your web component functionality, using the ECMAScript 2015 class syntax

Register your new custom element using the customElements.define() method, passing it the element name to be defined, the class or function in which its functionality is specified

If required, attach a shadow DOM to the custom element using Element.attachShadow() method. Add child elements, event listeners, etc., to the shadow DOM using regular DOM methods.

If required, define an HTML template using <template> and <slot>. Again use regular DOM methods to clone the template and attach it to your shadow DOM.

Use your custom element wherever you like on your page, just like you would any regular HTML element.

CUSTOM ELEMENTS LIFECYCLE CALLBACKS

- **connectedCallback -** Invoked each time the custom element is appended into a document-connected element. This will happen each time the node is moved, and may happen before the element's contents have been fully parsed.
- **disconnectedCallback -** Invoked each time the custom element is disconnected from the document's DOM.
- **attributeChangedCallback -** Invoked each time one of the custom element's attributes is added, removed, or changed.
- **adoptedCallback -** Invoked each time the custom element is moved to a new document.

FRAMEWORKS

- **StencilJS -** a simple library for generating Web Components and progressive web apps (PWA).
  (built by the Ionic Framework team for its next generation of performant mobile and desktop Web Components)

- **Polymer -** library for creating web components.
  (built by Google and used by YouTube, Netflix, Google Earth and others)

- **SkateJS -** library providing functional abstraction over web components.

- **Angular Elements**

DEMO

THE NATIVE WAY