

AR/VR Workshop

Part 3: **Foundations**

Phillip G. Bradford
University of Connecticut

Learning plan

Introduction	Overview – outline goals Setting up google cardboard with glitch.com & github
A-frame basics	Simple 3D a-frame examples Work with glitch.com
Foundations	JavaScript, DOM, events, Web-Components
A-frame components	Defining A-frame components
Three.JS and A-frame	Basics of Three.js for A-frame
Entity component architecture (ECA)	Three.js and ECA with standard OO paradigm
A-frame and planets	Complex 3D a-frame, work with complex a-frame detail and basic planetary math; illustrate ECA, geometries, controls, etc.
A-frame and animations	Goal: show how to do basic animation
Conclusion	Goal: review our learning

Outline

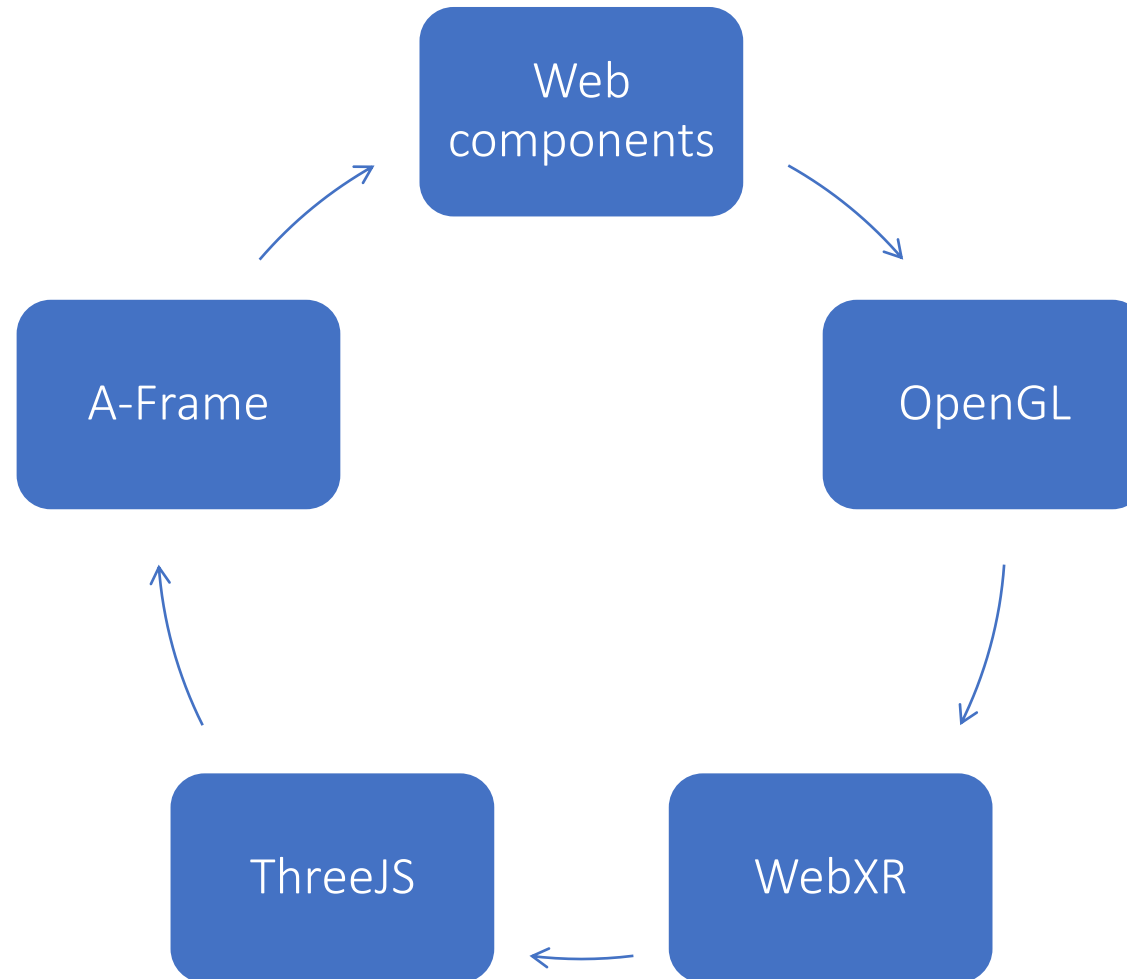
JavaScript, DOM and HTML

What are web components?

Really, HTML and VR?

Examples

High level view and learning path



What are web components?

Reusable and sharable parts for web pages – used in web frameworks

HTML – Hypertext Markup Language

DOM – document object model

DOM elements as HTML tags

Examples from: Mozilla Dev Network

<https://github.com/mdn/web-components-examples>

See also: Ben Farrell: Web components in action, Manning, 2019

Really, HTML, CSS and VR?

Ultimately, VR is presented on a screen

HTML/JS/CSS do a lot for screen presentation

- Hook to APIs

- Interactive

- Good separation of duties

Can leverage sophisticated libraries, e.g., OpenGL

DOM is a tree

```
<document>
  <html>
    <header>
    </header>
    <body>

    </body>
  </html>
```



Shadow DOM

Challenges with CSS over-riding your components?

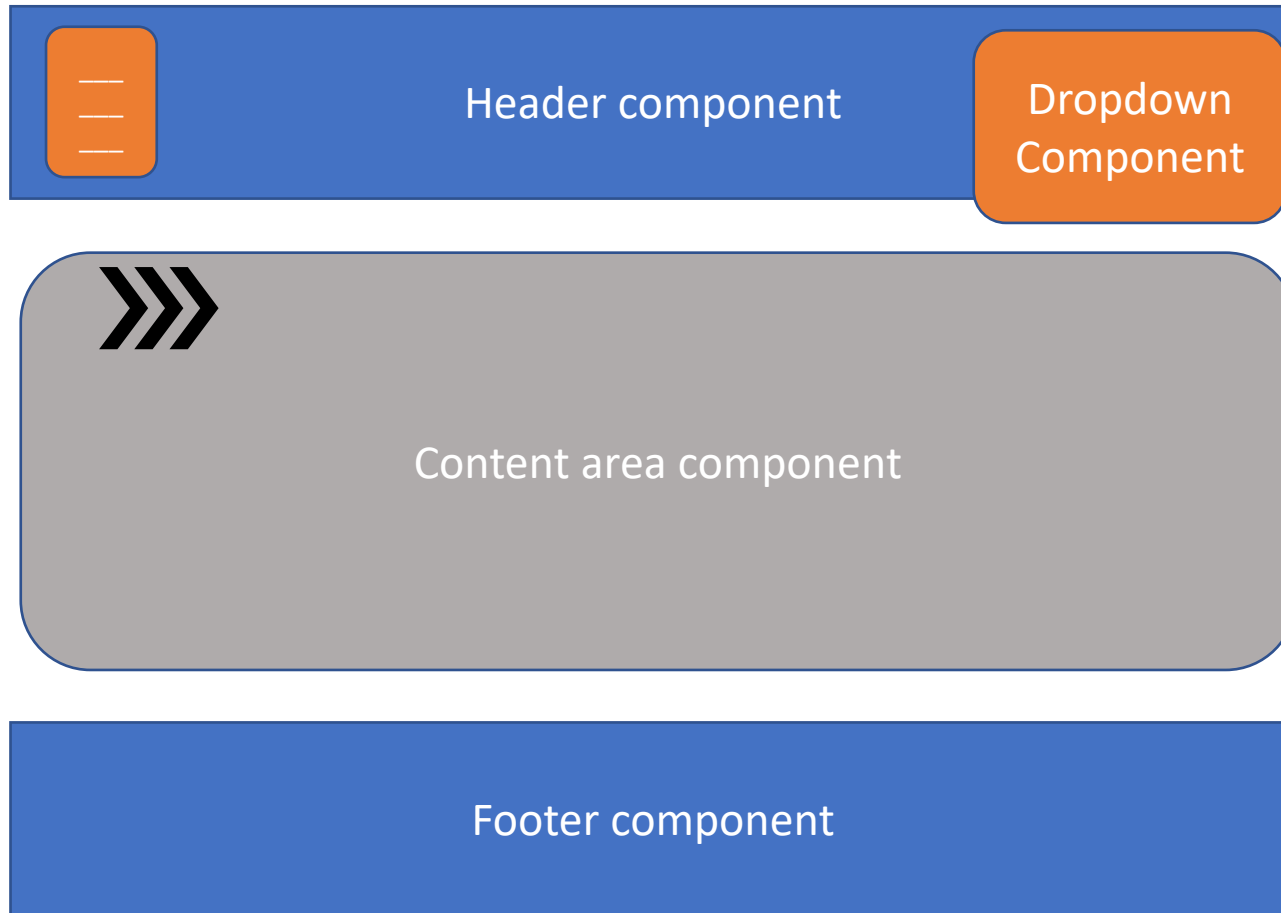
Inheritance in the DOM tree

Do we want CSS or other info inheriting down the DOM tree?

Always?

`Element.attachShadow()`

Component based development



Object oriented-ness of web components

```
customElements.define('a-new-frame', class extends HTMLElement {});

if (!customElements.get('a-new-frame')) {
    customElements.define('a-new-frame',
        class extends HTMLElement {});
}
```

MDN Word count example

Computer> git clone <https://github.com/mdn/web-components-examples.git>

Let's explore their WordCount subclass of HTMLParagraphElement

web-components-examples/edit-word

index.html

main.js

Two important attribute sets

Standard attributes – class, etc.

https://www.w3schools.com/tags/ref_standardattributes.asp

Event attributes – window, mouse, keyboard, media, etc.

https://www.w3schools.com/tags/ref_eventattributes.asp

Objective

Put up a component that counts words in the parent element

Word count rating widget

Hello! Count these words.

Words: 4

index.html

```
<html>
  <head>
    <meta charset="utf-8">
    <title>MDN Simple word count web component</title>
  </head>
  <body>
    <h1>Word count rating widget</h1>
    <article contenteditable="">
      <h2>Sample heading</h2>

      <p> three </p>
      <p> four    five </p>

      <p is="word-count"></p> // is – global attribute saying this HTML TAG should behave like a custom object
    </article>
    <script src="main.js"></script>
  </body>
</html>
```

main.js

```
class WordCount extends HTMLParagraphElement { // HTMLElement!?
  constructor() {
    super();

    const wcParent = this.parentNode;

    function countWords(node) { // method
      const text = node.innerText;
      return text.trim().split(/\s+/g).filter(a => a.trim().length > 0).length;
    }
    const count = `Words: ${countWords(wcParent)}`;
    const shadow = this.attachShadow({mode: 'open'});
    const text = document.createElement('span');
    text.textContent = count;
    shadow.appendChild(text);
  }
}
```

main.js

```
setInterval(function() {  
    const count = `Words: ${countWords(wcParent)}`;  
    text.textContent = count;  
}, 200);
```

```
customElements.define('word-count', WordCount, { extends: 'p' });
```

```
setTimer( fcn, 1000 );
```

<https://developer.mozilla.org/en-US/docs/Web/API/setInterval>

Web Components Lifecycle

```
<html>
<head>
  <meta charset="utf-8">
  <title>MDN Life cycle callbacks test</title>
  <style>
    custom-square { margin: 20px; }
  </style>
  <script defer src="main.js"></script>
</head>
<body>
  <h1>Life cycle callbacks test</h1>

  <div>
    <button class="add">Add custom-square to DOM</button>
    <button class="update">Update attributes</button>
    <button class="remove">Remove custom-square from DOM</button>
  </div>

</body>
</html>
```

main.js

See main.js in

<https://github.com/mdn/web-components-examples/tree/main/life-cycle-callbacks>