

# The Browser as a Canvas

Sean Park  
15-112 Optional Lecture!  
11/6/17

# Hi! I'm Sean.

- Some of my past work:
  - <https://seanpark.xyz>
  - <https://swipebuy.co>
  - <https://tokenbnk.com>

# What are we doing today?

- Basic
  - Client-Side: HTML / CSS / JS
  - HTML Tags
  - CSS Selectors
  - Recommended editors for Web Development

# What are we doing today?

- Intermediate
  - Javascript / jQuery
    - Asynchronous Event Handlers
  - CSS Frameworks
  - Debugging Front-end
  - Basic UI/UX Design

# What are we doing today?

- Advanced Demos
  - Animations
  - Advanced Interactions (sliders, modals, etc.)
- Mobile Responsive Design
- How can I learn more?

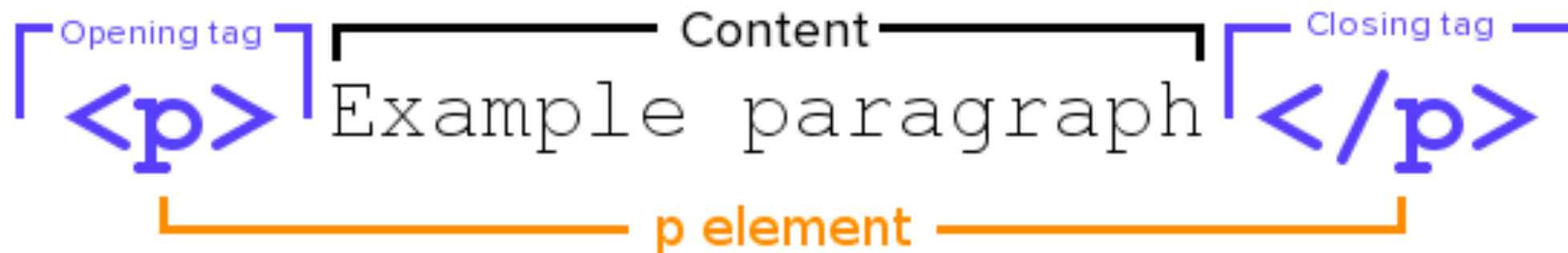
- Basic
  - Client-Side: HTML / CSS / JS
  - HTML Tags
  - CSS Selectors
  - Recommended editors for Web Development

# Overview - HTML / CSS / JS



- These are the languages browsers know how to run.

# HTML Tags



- This is how you tell your browser what to display!
- There are many different types of tags (h1, h2, ..., p, small, b, i, etc.)
- What is the difference between tags?
  - Mostly default styling and semantics



# HTML Tags

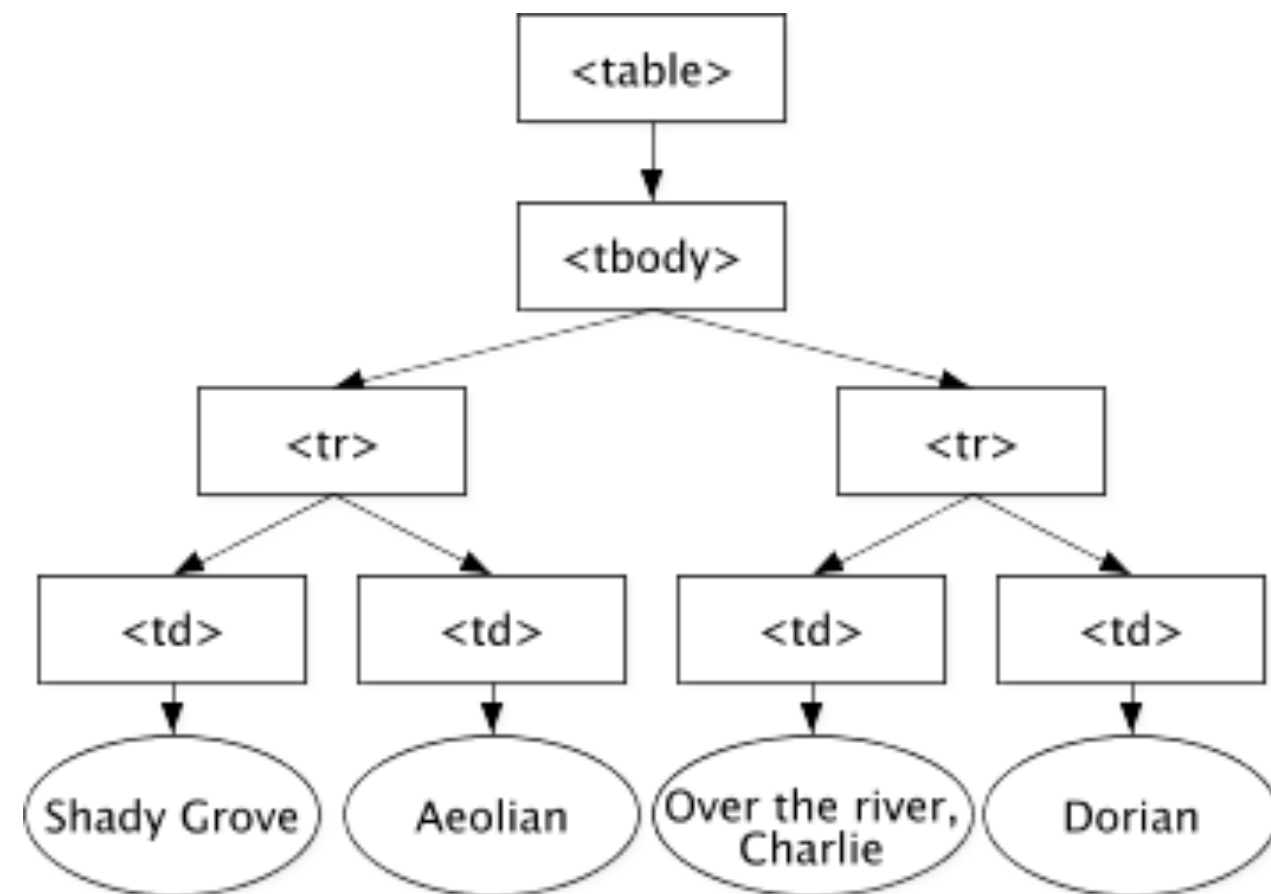
```
<footer>
  <div id="footer-left">
    
    <p>Swipebuy, Inc.</p>
  </div>
  <div id="footer-right">
    <table>
      <tr>
        <td><a href="">About</a></td>
        <td><a href="">Instagram</a></td>
      </tr>
      <tr>
        <td><a href="">Contact</a></td>
        <td><a href="">Snapchat</a></td>
      </tr>
      <tr>
        <td><a href="">Privacy</a></td>
      </tr>
    </table>
  </div>
</footer>
```

- Attributes describe the HTML tags
- What kind of different attributes do you see?

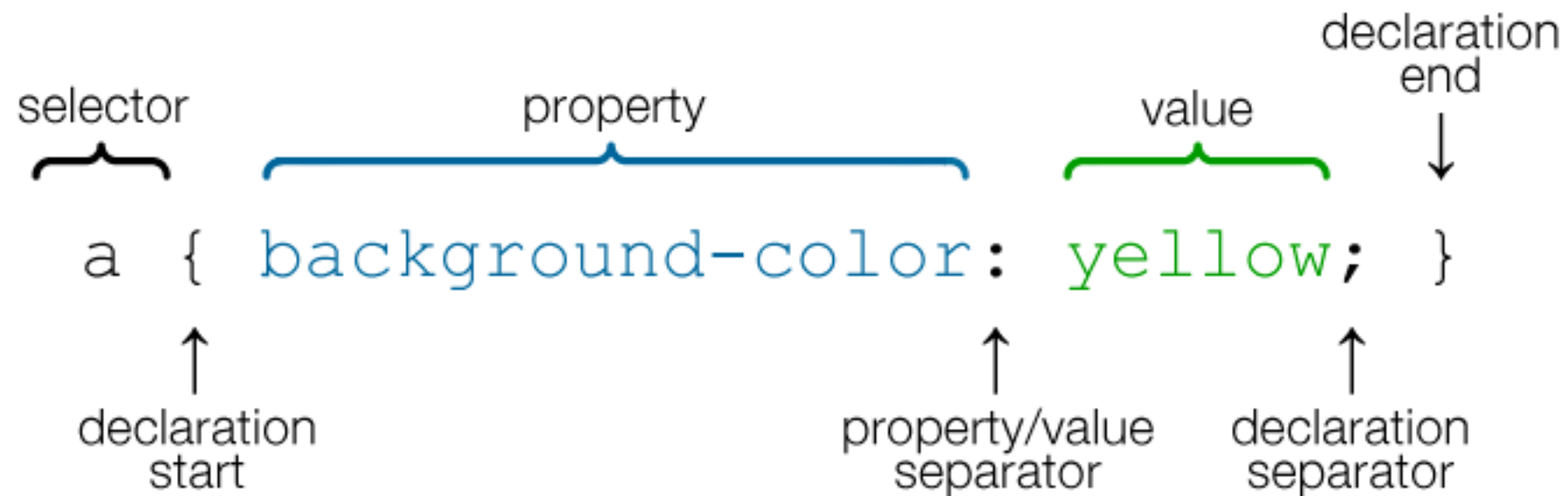
# DOM

## (Document Object Model)

```
<table>
<tbody>
  <tr>
    <td>Shady Grove</td>
    <td>Aeolian</td>
  </tr>
  <tr>
    <td>Over the river, Charlie</td>
    <td>Dorian</td>
  </tr>
</tbody>
</table>
```



# CSS Syntax



Each property has designated allowed values. It takes a while to get familiar with everything— Google is your best friend!

# CSS Selectors

```
1 | #danger { color: red; }  
2 | .warning { color: orange; }  
3 | p { color: green; }
```

- HTML elements are usually selected using their tag, id and/or class
- You can select using other attributes too, but those are not standard!

# CSS Selectors

```
<footer>
  <div id="footer-left">
    
    <p>Swipebuy, Inc.</p>
  </div>
  <div id="footer-right">
    <table>
      <tr>
        <td><a href="">About</a></td>
        <td><a href="">Instagram</a></td>
      </tr>
      <tr>
        <td><a href="">Contact</a></td>
        <td><a href="">Snapchat</a></td>
      </tr>
      <tr>
        <td><a href="">Privacy</a></td>
      </tr>
    </table>
  </div>
</footer>
```

- How can you select different elements inside this footer?

# HTML / CSS Resources

- There are tons of different HTML tags and CSS selectors. Check out these resources for decently comprehensive lists:
  - <https://www.w3schools.com/tags/>
  - [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

# Recommended Editors

- Atom: <https://atom.io/>
- Brackets: <http://brackets.io/>
- Sublime Text 3: <https://www.sublimetext.com/3>

- Intermediate
  - Javascript / jQuery
    - Asynchronous Event Handlers
  - CSS Frameworks
  - Debugging Front-end
  - Basic UI/UX Design



# Javascript in the DOM

```
document.addEventListener("DOMContentLoaded", function() {  
    document.getElementById("test").onclick = function() {  
        clickEvent(this)  
    };  
});
```

- Notice that the click event handler is inside the event listener for the event 'DOMContentLoaded'!

# jQuery Version

```
// jQuery Version

$(document).ready(function() {

    $('#test').click(function() {
        clickEvent(this);
    });

});
```

# How do we set up jQuery?

- jQuery provides Javascript wrappers around many functions that manipulate the DOM.
- Simply insert a script tag with the src set to the downloaded jQuery file or CDN link!
- CDN Links: <https://code.jquery.com/>

# CSS Frameworks

There are many CSS libraries that are already written, and can just be downloaded / linked just like jQuery! Frameworks help:

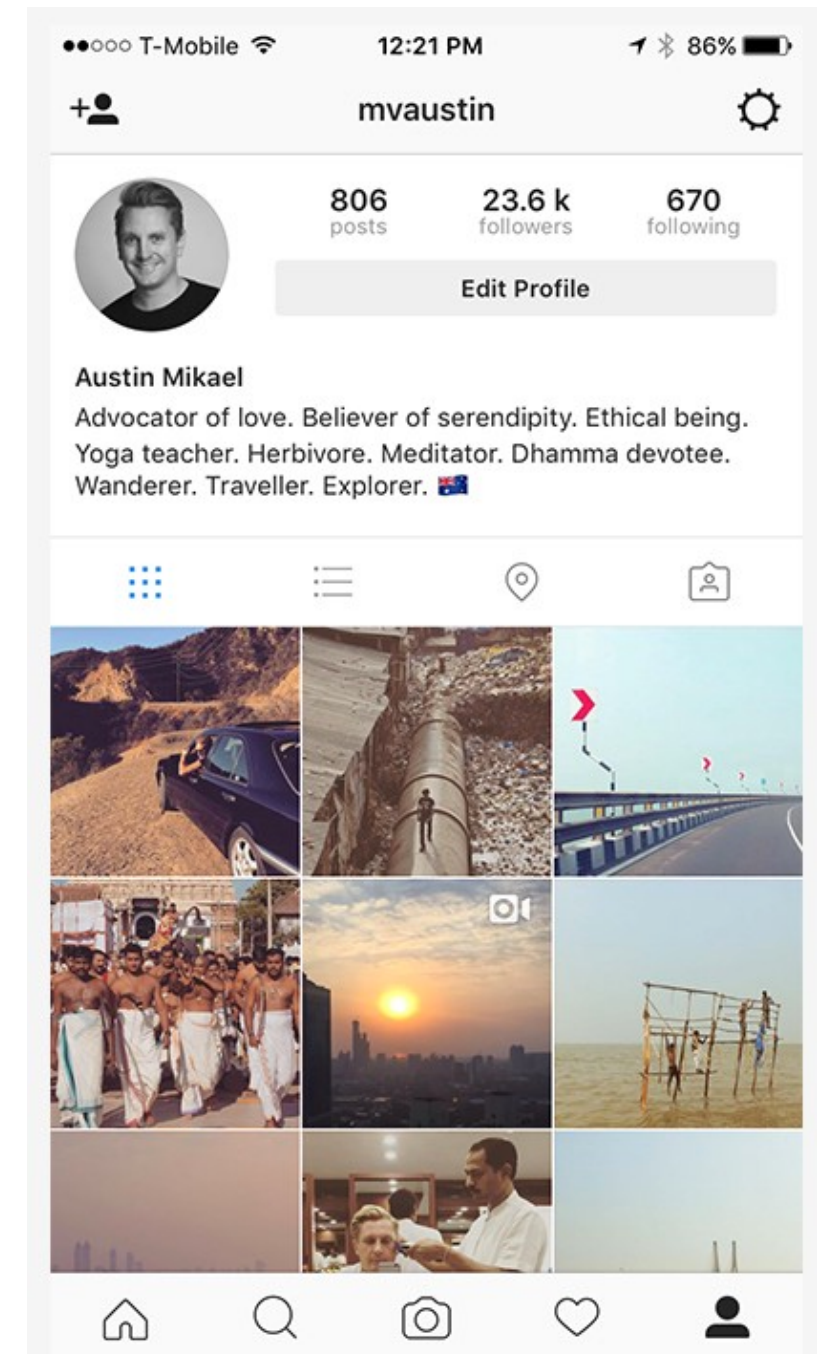
- Provide decent designs
- Maintain a visual theme
- Save time

# CSS Frameworks

- Materialize: <http://materializecss.com/>
- Bootstrap: <http://getbootstrap.com/>
- Foundation: <https://foundation.zurb.com/>
- UIKit: <https://getuikit.com/>

# Quick UI/UX Design Principles

- If in doubt: use a color palette!
  - <https://coolours.co/>
- Use grid layouts
- Maintain a good balance of white space and content
- What is most intuitive?



# This is all you need to make cool websites!

- My first ever website: <https://seanpark.xyz/prantls>
- Let's make a quick improvement to the Instaclone web app demo below.

<https://github.com/Ahezarkhani/instaclone>

- Advanced Demos
  - Animations
  - Advanced Interactions (sliders, modals, etc.)
- Mobile Responsive Design
- How can I learn more?



# How can I learn more?

- Take an online course! (Codecademy, Udemy, Lynda, Udacity, etc.)
- But no online course is going to cover all the little details. The best thing is to do is to get hacking!
- If you want to set up your personal website using GitHub Pages, check out my workshop walkthrough at [seanpark.xyz/yolo](https://seanpark.xyz/yolo)

# Thanks for coming!

Hit me up if you have any questions:

Sean Park  
[spark3@andrew.cmu.edu](mailto:spark3@andrew.cmu.edu)