

PROFESSIONAL & CONTINUING EDUCATION  
UNIVERSITY of WASHINGTON



# LESSON 01: MODERN JAVASCRIPT & DEBUGGING

---

## HTML 300



# OVERVIEW

---

1. Logistics
2. Package Manager
3. jQuery Review
4. Learn some modern JavaScript (ES6+).
5. Debugging, Working in Strange Codebases
6. Pull Requests, Code Reviews
7. Visual Design
8. Project Management 101
9. Intro to Course Project



PROFESSIONAL & CONTINUING EDUCATION

UNIVERSITY *of* WASHINGTON

# Hello!



# INTRODUCTION

---

- Aaron Katz
  - Full Stack Web Developer at Jobscan.co
  - 5 Years developing and 4 teaching (some overlap)
  - Really likes dogs
  - Try to make learning fun
  - Want to help you reach your career goals
- 
- Now tell me your name, why you are taking this course, and a useless super power you would like to have.



# THIS CLASS

---

- > Fast pace! Work hard to not fall behind, ask for help if you do.
- > Focus is on higher level skills: analysis, research, decision making. These are what let you grow to be more than just a “dev”, but do not take the place of writing good code!
- > Every class I will take a couple of minutes to answer any questions you might have about the life of a "dev".



# ASSIGNMENTS

---

- > There will be at least one assignment and one quiz each week.
- > Some lessons will have 2 graded assignments.
- > Some lessons build on each other, so keep up to date.
- > See Canvas website for due dates.
- > Late assignments lose 25% of possible points.



# CLASS CONVERSATIONS

---

- > We have class conversations that everyone will participate
- > Talking about technology topics is difficult, and essential.



# REQUIREMENTS TO PASS THE COURSE

---



- > Complete every assignment.
- > Miss no more than 2 class meetings.
- > Score an 80% overall on coursework.





# GOALS OF THE COURSE

---

- > To become a self-sufficient front-end developer
- > To continue to learn independently using resources such as:
  - [Stack Overflow](#)
  - [Google](#)
  - [YouTube](#)
  - [Mozilla Developer Docs](#)
- > Remember: if you have a question/problem, it is 99% likely someone else has already asked it & someone has answered it online



PROFESSIONAL & CONTINUING EDUCATION  
UNIVERSITY *of* WASHINGTON

# Package Manager



# PACKAGE MANAGER

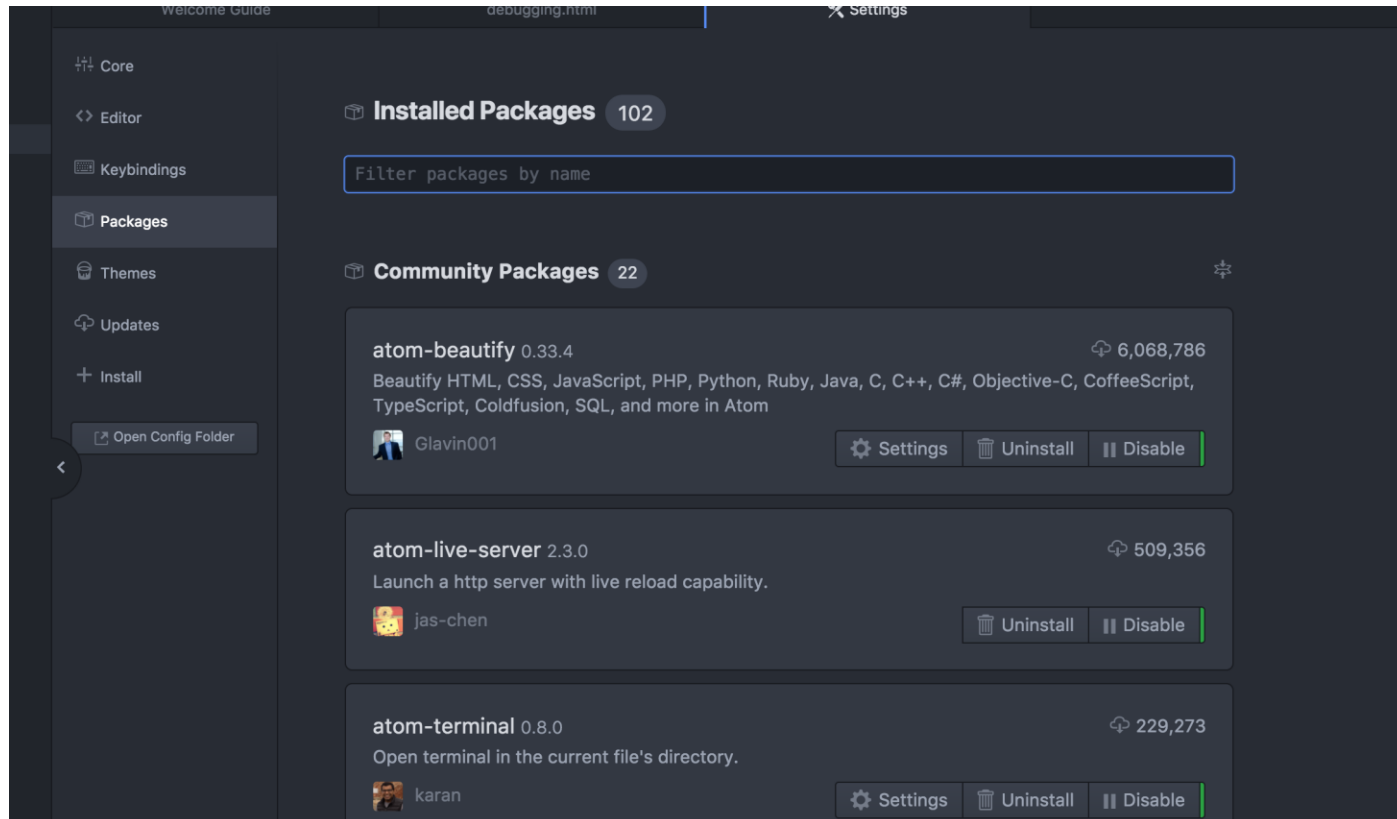
---

- There are three packages I would recommend installing in your code editor.
  - Emmet
  - Atom Beautify
  - Atom-live-server
- To get to the package manager in Atom
  - Click on the "Atom" option in the top menu bar and select "Preferences."
  - Then in the new option menu click on "Packages"



# PACKAGE MANAGER

- You should see this.



# PACKAGE MANAGER

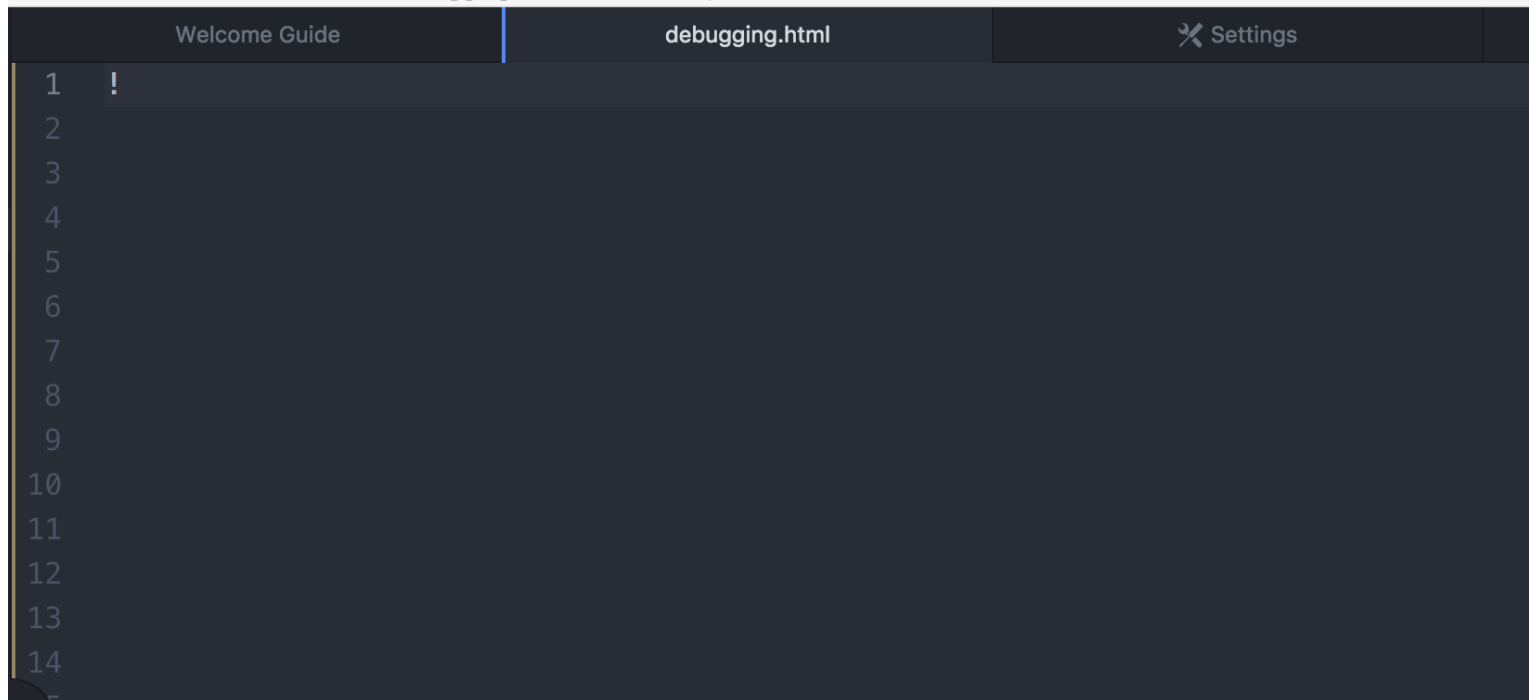
---

- Type the following one at a time in the input box
  - Emmet
  - Atom Beautify
  - Atom-live-server
- Click on the install button to install the package.



# EMMET

- Emmet will save you a lot of time by not making you type as much code.
- For example if you type "!", and then press tab.



A screenshot of a code editor interface with a dark theme. The top bar shows three tabs: 'Welcome Guide', 'debugging.html' (which is active), and 'Settings'. On the left side, there is a vertical line of numbers from 1 to 14, representing line numbers. The main editing area shows line 1 with an exclamation mark '!' followed by a cursor. The rest of the editor is empty.



# EMMET

- Then you get this.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
13
14
15
16
17
```



# jQuery Review





# JQUERY CDN

- I use Googles CDN it can be found at <https://developers.google.com/speed/libraries/>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <title>Document</title>
</head>
<body>

</body>
</html>
```



# USING JQUERY

---

- We are going to add another Script tag and insert our starter code for jQuery

```
<script>
  $(document).ready(function() {
    alert("test");
  });
</script>
```

- This will cause an alert with a message of "test" when you run the HTML page



# ADDING HTML

---

- In the body we are going to add an input box with an id of "getText" and a button with an id of "clicked".

```
<body>
  <input type="text" id="getText">
  <button id="clicked">Click Here</button>
</body>
```

- Then in our jQuery we are going to add a function that when the button is clicked alert to the screen the text in the input box.



# ADDING JQUERY

---

- Then is our jQuery we are going to add a function that when the button is clicked alert to the screen the text in the input box.

```
<script>
$(document).ready(function() {
  $("#clicked").click(function(event) {
    console.log($("#getText"));
    alert($("#getText").val());
  });
});
</script>
```



# RESULTS



- Any questions before we move on.
- You will need to use jQuery for your first assignment, so if you have a questions now is the time.

W

PROFESSIONAL & CONTINUING EDUCATION

UNIVERSITY *of* WASHINGTON

# ES6 JavaScript



# INTRODUCTION

---

- > Modern JavaScript syntax (ES6+) has several goals.
- > Our main objectives are to use
  - new syntax
  - array/object methods
  - string template literals
  - make writing JavaScript faster, cleaner, and with less code.



# ES6 FUNCTIONS

> All can be found at :

<https://github.com/UWFront-End-Cert/html300.git>. Go into lesson01, inclass, es6. Use jsbin.com as a sandbox

- |             |                         |
|-------------|-------------------------|
| 1. forEach  | 6. every                |
| 2. map      | 7. some                 |
| 3. filter   | 8. find                 |
| 4. reduce   | 9. string interpolation |
| 5. includes | 10. arrow functions     |





PROFESSIONAL & CONTINUING EDUCATION

UNIVERSITY *of* WASHINGTON

# WORKING IN NEW CODEBASES



# SET UP

---

> Fork:

<https://github.com/UWFront-End-Cert/html300.git>

- > Create a new branch with your last name and week\_one.
- > Add me as a collaborator, my username is akatz6.
- > If your fork needs to be updated go into your terminal and type the following:
  - `git remote add upstream https://github.com/UWFront-End-Cert/html300.git`
  - `git fetch upstream`
  - `git fetch master`
  - `git merge upstream/master`
  - `git push`



# THREE WAYS TO DEBUG

---

- > `alert("test");`
- > `console.log("test");`
- > `debugger;`



# DEBUGGING TOOLS

---

- > Browser Developer Consoles
- > Vue extension for a [Chrome](#)/[Firefox](#).
- > Google's [Lighthouse](#)/[PageSpeed Insights](#)
- > [Can I Use](#) (to verify if browser support is available)



# EXERCISE

---

- > After you have created a new branch open the debugging.html file.
- > This file is full of bugs and uses for loops.
- > Find the bugs using the debugging tools we reviewed and integrate the ES6 functions we spoke about to replace the strings and for loops.
- > Also, don't forget to add comments





# LET'S GET TO WORK



Click to add text

# W

# CODE REVIEWS

---

- > Find bugs
- > Improve or make consistent code, style, and architecture decisions
- > Share information about how the project works



# PULL REQUESTS

---

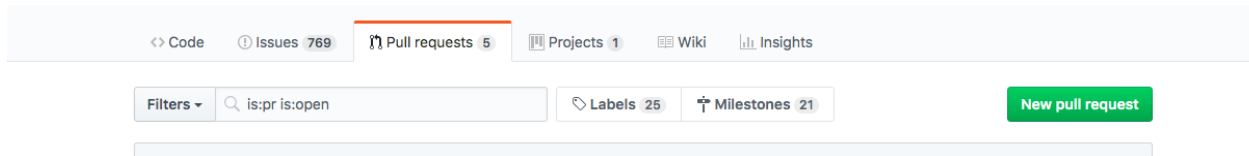
- > To have your code changes incorporated into a shared codebase, you make a pull request on GitHub. This allows collaborators to review your work.
- > You may have to add a collaborator to review your code. Directions on how to do this are at <https://help.github.com/en/articles/inviting-collaborators-to-a-personal-repository>.





# LET'S DO A PULL REQUEST

- > Commit and push your changes to your repo
- > Click “Pull Requests” tab
- > Click “New Pull Request” button



- > “Base” is my repo, master branch; “compare” is your repo, branch
- > Add your instructor as a reviewer.
- > Add a title and friendly, informative description
- > Push the big green button!



PROFESSIONAL & CONTINUING EDUCATION

UNIVERSITY *of* WASHINGTON

# BAD WEB SITES



# IT HURTS MY EYES

---

Let's look at these sites and why they are bad

- > <https://blinke.com/>
- > <https://www.lingscars.com/about>
- > <http://lifeactionrevival.org/>
- > <http://www.arngren.net/>
- > <http://www.libertyvan.com/>
- > <http://www.roverp6cars.com/>



# GOOD WEB SITES



# I WANT TO EXPLORE AND MAYBE BUY SOMETHING

---

What makes these sites good?

- > <https://www.gregorypacks.com/home>
- > <http://www.espn.com/>
- > <https://www.airbnb.com/>
- > <https://www.google.com/>
- > <http://www.timelessliving.mu/>



# ACTIVITY

---

There is no silver bullet for what makes a great web site. Some people like to scroll, while others want to click. Some like to interact while others want easily find what they are looking for.

- > We are going to build a list of what we think is important in a web site design.
- > Explain four reasons why you like a web site and four reasons why you hate a web site. For each reason you will need to provide a web site that explains your reasoning.



# WHAT I LIKE IN A WEB SITE

---

**W**

# WHAT I HATE IN A WEB SITE

---

W



# VISUAL DESIGN BASICS

---

You've already identified some of the fundamentals of design

- > Space – similar items together
- > Consistency
  - consistent layouts provide predictability
  - break consistency thoughtful
- > Reduce mental strain
  - keep copy (text) minimal
  - make it clear what you want users to do
- > Use color thoughtfully
  - Ensure colors have high contrast for readability and accessibility
- > Typography should be functional



# THE BEST ARTISTS BORROW

- > Browse these themes and re-create them with your own flair
  - [WordPress popular themes](#)
  - [WordPress showcase](#)
  - [awwwards](#)
- > Want to read more?
  - [The non-designers design book by Robin Williams](#)
  - [Learning from Lego \(modular web design\)](#)

## The Best Artists...Borrow

Most of you are not designers. It's totally fine to base some or most of your projects visual design on someone else's site. Start with imitation, and then introduce your own ideas.



PROFESSIONAL & CONTINUING EDUCATION

UNIVERSITY *of* WASHINGTON

# PROJECT MANAGEMENT 101



# AGILE METHODOLOGY

---

- > Collaborative
- > Adaptive
- > Continual Improvement
- > Continual Delivery



# VOCAB: ITERATE OR ITERATION

---

- > Do something over and over!
- > It's acceptable to ship a product or feature that is “good enough” right now, get feedback, then **iterate** on it to improve it.



# VOCAB: SPRINT

---

- > A sprint is a work period containing planning, execution and review phases.
- > Most companies use two-week sprints.
- > A sprint planning meeting is held where stakeholders talk about goals and time to complete those goals.
- > At the end of those two weeks the stakeholders review the results of the sprint planning meeting.
- > If you feel that you cannot complete your goals in the allotted time, speak up and explain why.



# VOCAB: STAND UP MEETING

---

- > Very brief status meeting, usually with all participants standing to keep it quick. Sometimes is a team, sometimes whole company. Often held daily, first thing in the morning.



# VOCAB: STAKEHOLDER

---

- > A **stakeholder** is someone who will be affected by the outcome. A software feature's stakeholders might be the sales team, the engineering team, and the project manager. These are the people who will have a voice in decision making, as outcomes will affect them.





# WHAT IS THE GOAL

---

- > Project Management is about getting work done efficiently. There are countless strategies, but bottom line is almost always to have the clearest possible **communication** between stakeholders and developers who are doing the work.



# DO A GUT CHECK

---

DO YOU KNOW...

1. What you need to do to pass this course?
2. How to use modern JavaScript (ES6+)?
3. How to debug code?
4. How to identify good design?
5. How to use the basics of project management?
6. How to complete the course assignments?



# ANY QUESTIONS?

---

As always feel free to contact me through Canvas if you have any questions. I do have a full-time job, so I might not get back to you immediately.

If you don't hear back from me in 24 hours, please ping me again.

