COMP6080
Web Front-End Programming

Week 1.1
Outline

What is a front-end developer?

Someone who writes code that is executed client-side, typically part of a end-user facing product.

They largely use HTML to structure pages, CSS to style them, and Javascript to make them dynamic.

They build things for *people* to use.

Why COMP6080?

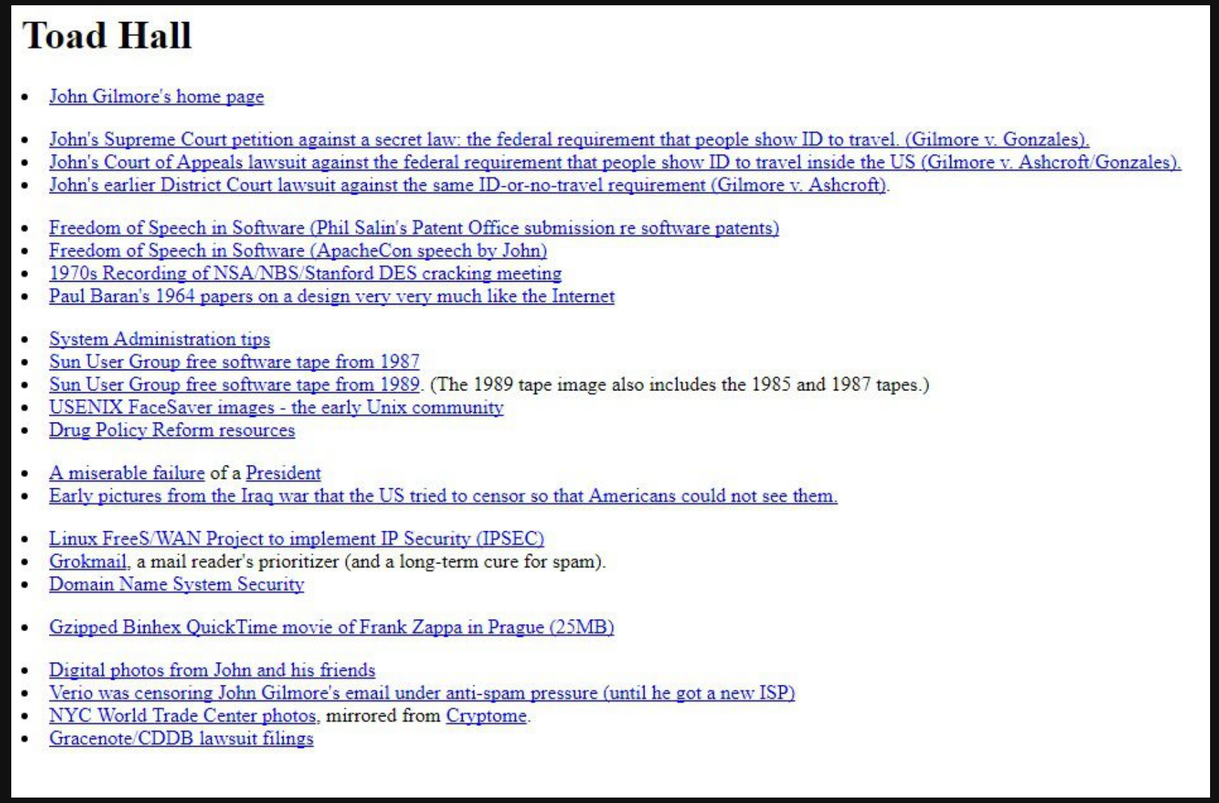
Web is exploding.

How did we get to this point?

The progress in web

1. HTML Static Pages

Beginning of websites



Toad Hall

- John Gilmore's home page

- John's Supreme Court petition against a secret law: the federal requirement that people show ID to travel. (Gilmore v. Gonzales).
- John's Court of Appeals lawsuit against the federal requirement that people show ID to travel inside the US (Gilmore v. Ashcroft/Gonzales).
- John's earlier District Court lawsuit against the same ID-or-no-travel requirement (Gilmore v. Ashcroft).

- Freedom of Speech in Software (Phil Salin's Patent Office submission re software patents)
- Freedom of Speech in Software (ApacheCon speech by John)
- 1970s Recording of NSA/NBS/Stanford DES cracking meeting
- Paul Baran's 1964 papers on a design very very much like the Internet

- System Administration tips
- Sun User Group free software tape from 1987
- Sun User Group free software tape from 1989. (The 1989 tape image also includes the 1985 and 1987 tapes.)
- USENIX FaceSaver images - the early Unix community
- Drug Policy Reform resources

- A miserable failure of a President
- Early pictures from the Iraq war that the US tried to censor so that Americans could not see them.

- Linux FreeS/WAN Project to implement IP Security (IPSEC)
- Grokmail, a mail reader's prioritizer (and a long-term cure for spam).
- Domain Name System Security

- Gzipped Binhex QuickTime movie of Frank Zappa in Prague (25MB)

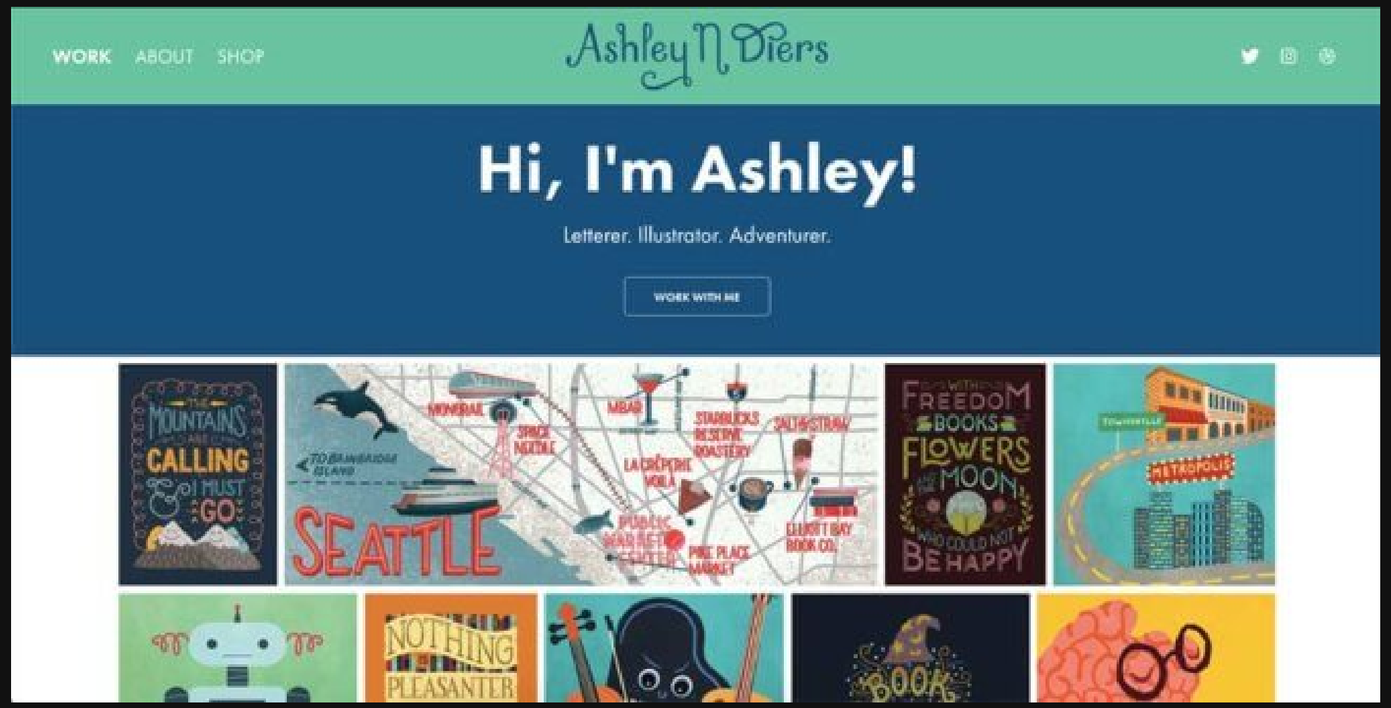
- Digital photos from John and his friends
- Verio was censoring John Gilmore's email under anti-spam pressure (until he got a new ISP)
- NYC World Trade Center photos, mirrored from Cryptome.
- Gracenote/CDDB lawsuit filings

HTML released 1993

The progress in web

2. Basic CSS / Javascript

Now things don't like terrible

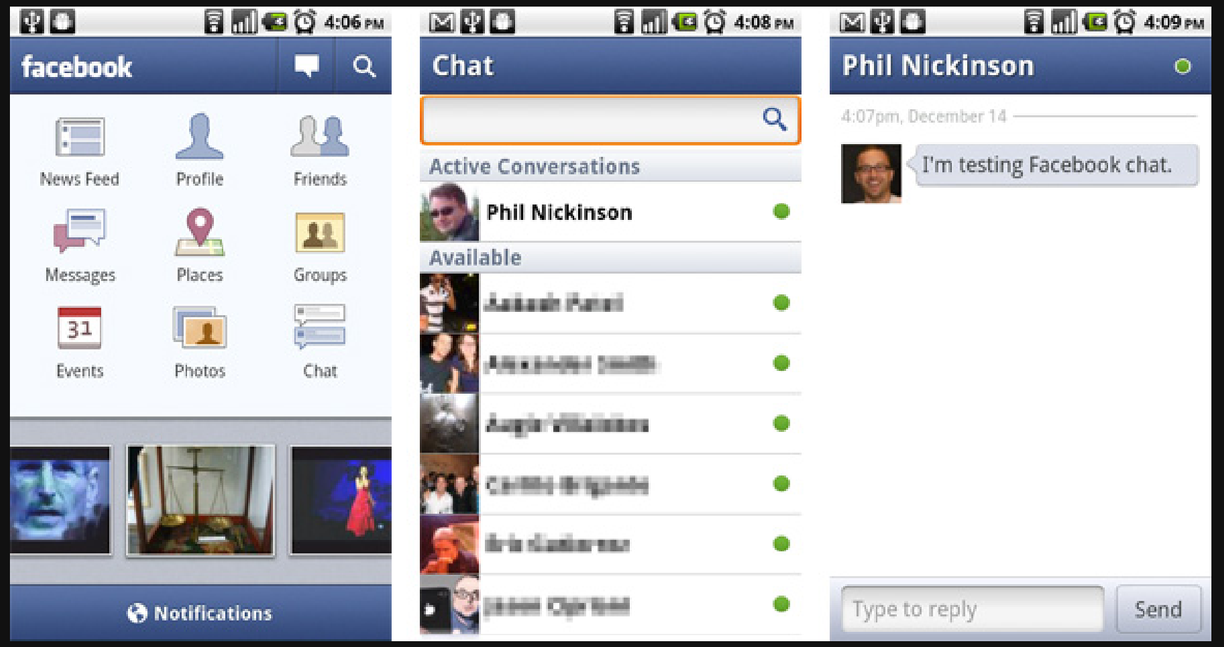


JS released 1995, CSS released 1996

The progress in web

3. AJAX and background requests

A whole new class of websites, without the need to refresh

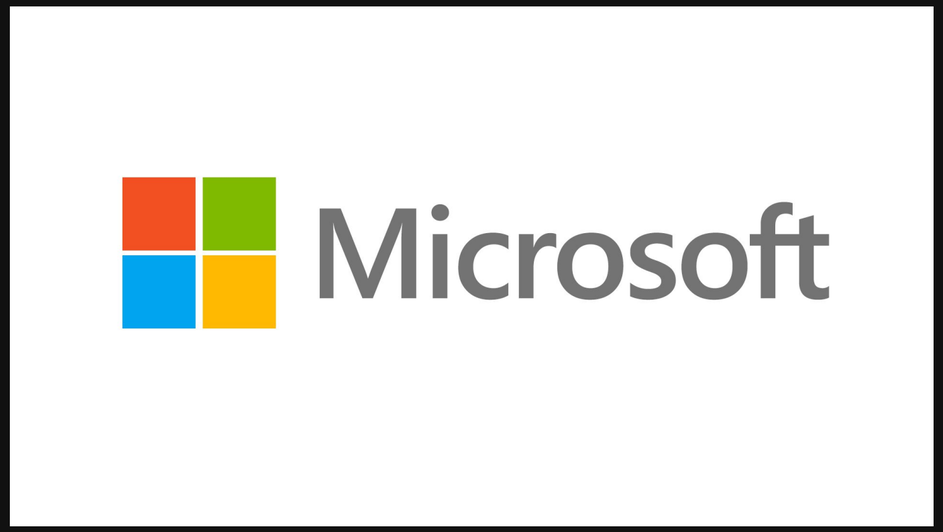


AJAX appeared 2000~

The progress in web

4. NodeJS & Typescript

Type checking and common codebases -
the beginning of heavy JS infrastructure



NodeJS released 2009, TS released 2012

The progress in web

5. Hybrid Apps, Electron

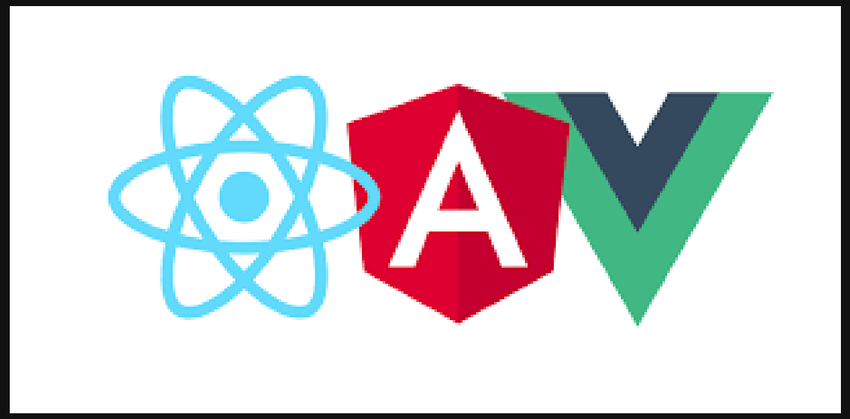
Type checking and common codebases -
the beginning of heavy JS infrastructure

Electron, Apache Cordova

The progress in web

6. Declarative Frameworks

Building of complex applications rapidly and in a scalable way. A web approach finally optimised for apps, rather than pages.



AngularJS released 2010, ReactJS released 2013, VueJS released 2014

The progress in web

In the vast majority of cases, your native desktop applications, native mobile applications, and web-apps or websites, can be built on a web-based, javascript based stack.

Course Overview

This is a new course, and we appreciate your understanding as we figure things out together.

- Assumed Knowledge
- Learning Outcomes
- Teaching approach

Assumed Knowledge

- Basics of GIT
- Basics of HTTP & Web Browsers
- High level understanding of scripting languages

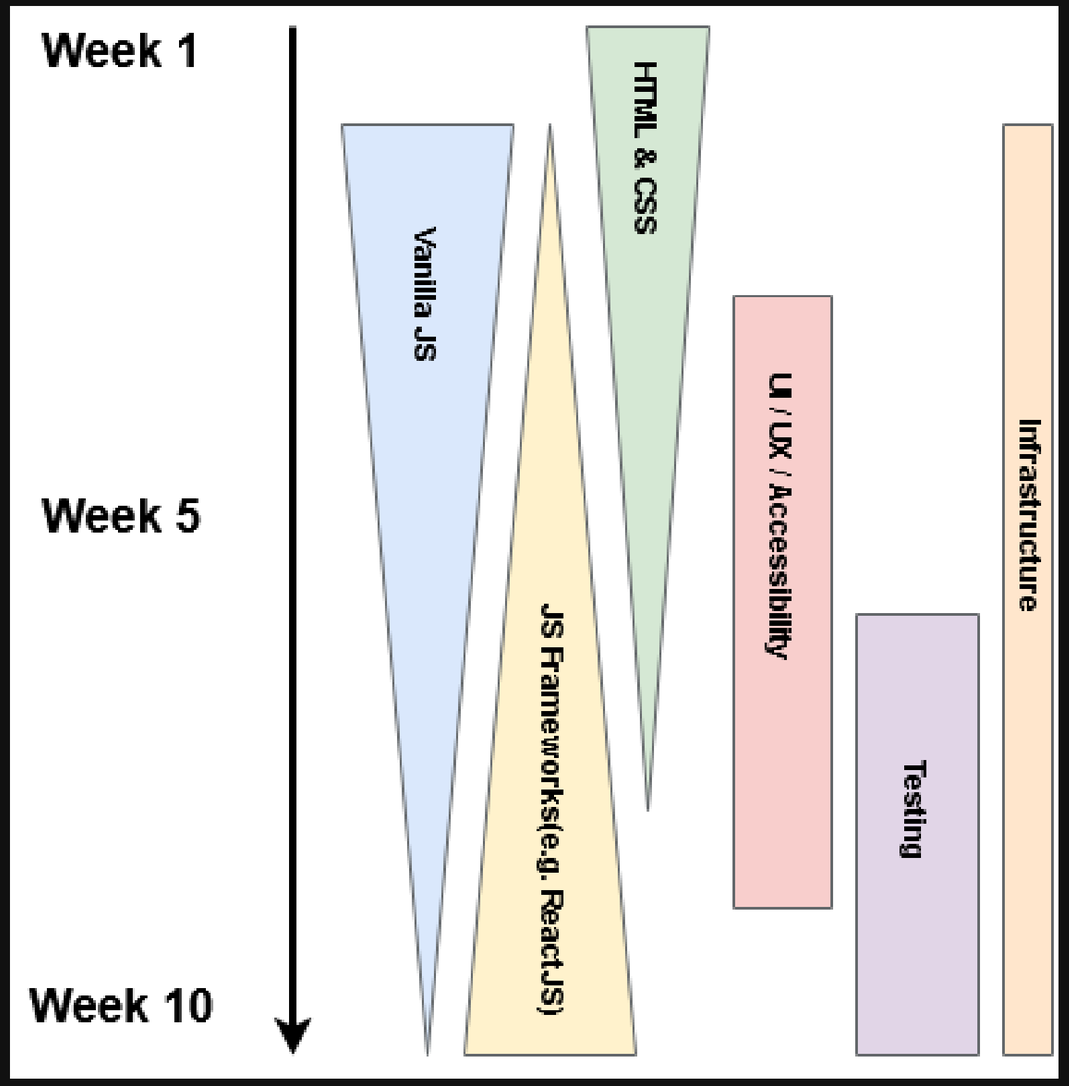
This course has no other assumed knowledge. Experienced web developers will likely find this course slow.

Learning Outcomes

Summarised

1. Fundamentals of Javascript to design, construct, test, debug code
2. Understanding HTML, CSS, DOM, to construct web pages
3. Modern Javascript & CSS frameworks, to build componentised apps
4. Understanding of asynchronous programming in the context of JS
5. Basic knowledge of front-end security
6. Awareness of UI/UX design, including accessibility

Approach to teaching front-end



A number of topics won't be included in the course. You can read more about those specific topics in our FAQ.

Teaching front-end comes with it's challenges, including:

- Rapidly developing tool sets
- The need to understand modern tools & fundamental knowledge
- The breadth of the tech stack on which web is built

The Team

We have a team of staff from UNSW, Canva,
and Google assisting with the course.

The Team

UNSW Team

- Hayden Smith
- Anto Lepejian
- Clarence Feng
- Daniel Latimer
- Emily Luxa
- Hoya Lee
- Justin Liang
- Kaiqi Liang
- Marina Reshetnikova
- Mariya Shmalko
- Melanie So
- Ravija Maheshwari
- Simon Haddad
- Soorria Saruva

Zain Afzal

Google

The Team

Creators & Presenters

Mark
Canva Pro Team

Anna
Presentations Team
UNSW Alum ('18)

Denis
App Store Extensions Team

Sam
Content Team

Christian
Frontend Infrastructure

Nic
Print Product Team

Sparky
Design Automation

Fahad
Onboarding Team

Tom
Ingredients Studio
UNSW Alum ('17)

The Team

With support from

Jess
University Liaison

Michael
Brand Designer

Muriel
Brand Designer

Peter
Video Download Team
UNSW Alum ('13)

Assessment Structure

There is no direct assessment associated with Lectures or Tutorials

Item	**Due**	**Weighting**
Assignments	Due weeks 3, 7, 10	75%
Exam	Exam Period	25%

Teaching Methods

Lectures:

- Avg 2 hours of pre-recorded per week (tightly broken up)
- Avg 2 hours live lectures (Mon & Thu 6pm-8pm) (often to reinforce)

Tutorials:

- Chance to further elaborate on content from lectures and discuss topics
- 1 hour per week with your tutor

Teaching Methods

Exercises / Help Sessions:

- Chance to talk to tutors and get help on matters to do with tutorials, exercises, and assignments
- Each week we provide you with a copy of exercises to complete. At the end of each week we will release solutions.
- Chance to practice skills discussed in lectures and tutes in breadth
- Help sessions will contain 2-4 tutors who will be split between assisting students with questions, and marking labs off

Assignments:

- A chance to practice skills in depth
- Ass1: Static HTML/CSS + Intro to DOM
- Ass2: Building an app with HTML/CSS/VanillaJS
- Ass3: Building and testing an app with ReactJS

Getting Help

If you need help with something, go here:

1. EdStem (sidebar on Webcms3)
2. Help Sessions
3. cs6080@cse.unsw.edu.au

Getting Setup

Running your code:

- In this course you'll need to use: NodeJS, NPM, Web Browser, HTML, CSS, ReactJS, and more.
- All of these tools can be easily run on your local machine (recommended) for OSX, Linux, or Windows.
- If you aren't comfortable with local installs, you can complete the course using gitlab (we have installed all relevant programs there).

Gitlab

- gitlab is the online tool we're using to manage our git repositories.
- We will provide a demo of it's usage.
- There are git resources on Webcms3.

Resources

- Web is a very well-resourced set of tools on the internet. Self-guided research will generally be adequate.
- The biggest issue will be finding the *appropriate* resources.
- Lectures may include resources on slides, and we also have a course-wide resources page here.

Style Guide

- Information about style guides for particular languages can be found on the style guides page.

Feedback

- Throughout term, you can leave anonymous feedback by clicking on the link in the Webcms3 sidebar "Feedback"

Getting Along

- Understand the expectations around student conduct.
- Create an inclusive learning environment.
- Let's all treat each other with respect and understanding.

Assignment 1

- Let's take a look at assignment 1 together

Your own personal copy of this repository has been deployed to your gitlab space.