COMP6080

🍓 Other - Outline

Lecture 1.1

Author(s): Hayden Smith

(Download as PDF)

🤷‍♀️ Why COMP6080?

Front-end is everywhere.

Most front-end is web or mobile.

- Stack Overflow Developer Survey 2025: In 2025, 69%~ of professional developers reported using JavaScript - the most popular use of a programming language.
- GitHub Octoverse 2024: JavaScript and Typescript are the #2 and #3 are the most active languages on the platform.
- 97.6% of all websites use JavaScript as their client-side programming language (according to W3Techs, 2023). This makes JavaScript an essential technology for building dynamic and interactive web pages.
- React.js (built on JavaScript) is the most popular front-end framework, used by around 40% of developers according to the Stack Overflow Developer Survey in 2024.

🤷‍♀️ Why COMP6080?

Within 10 weeks I want to make you all confident in building your own simple web apps, to build them quickly, and to build it to look how you want it to look and behave.

I want you confident using AI to build learn and then eventually to build.

🎨 What Is A Front-End Developer?

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

In most cases, they write code for web browsers or native apps. This is a course about helping you make web browsers do things.

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

They build things for *people* to use.

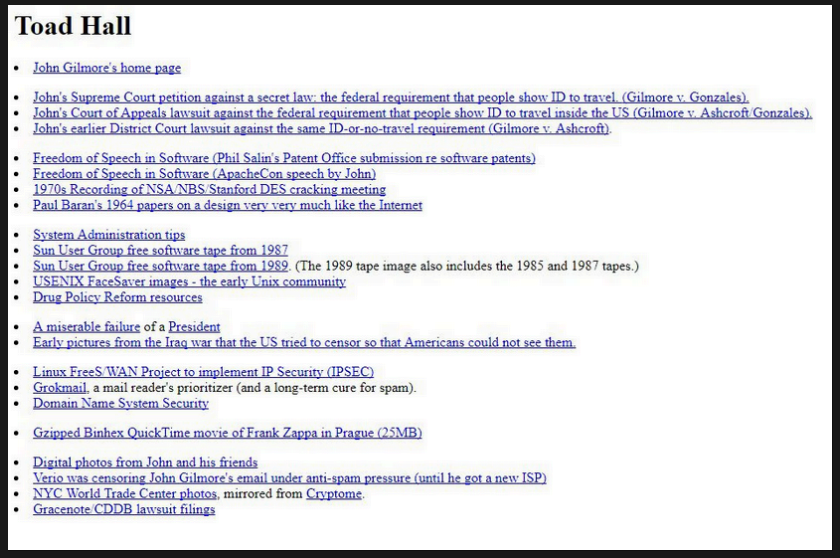
👵 A Bit Of History

How have the last 30 years unfolded?

 The Progress In Web

1️⃣ 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)

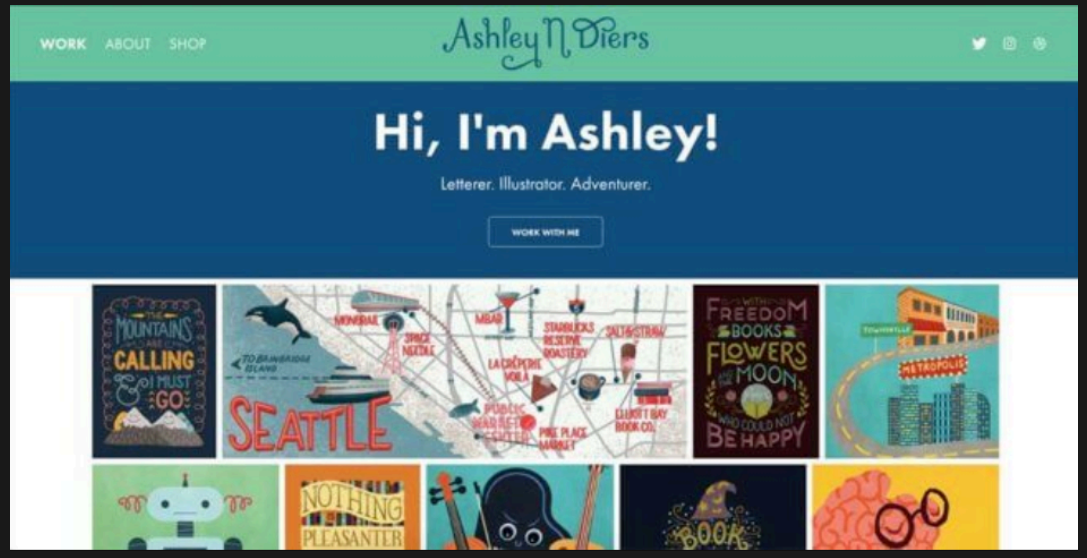
- Digital photos from John and his friends
- Verio was censoring John Gilmore's email under anti-spam pressure (until he got a new ISP)
- NYC World Trade Center photos, mirrored from Cryptome.
- Gracenote/CDDB lawsuit filings

HTML released 1993

🏃‍♀️ The Progress In Web

2️⃣ Basic CSS / Javascript

Now things don't like terrible

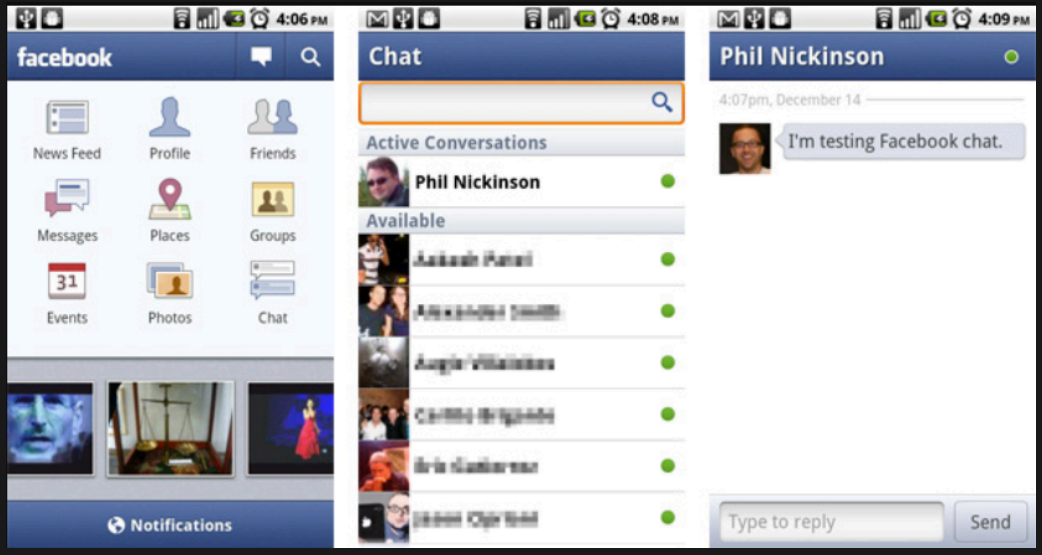


JS released 1995, CSS released 1996

 The Progress In Web

3️⃣ AJAX and background requests

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



AJAX appeared 2000~

🚶‍♀️ The Progress In Web

4️⃣ NodeJS & Typescript

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

NodeJS released 2009, TS released 2012

🏃‍♀️ The Progress In Web

5️⃣ Hybrid Apps, Electron

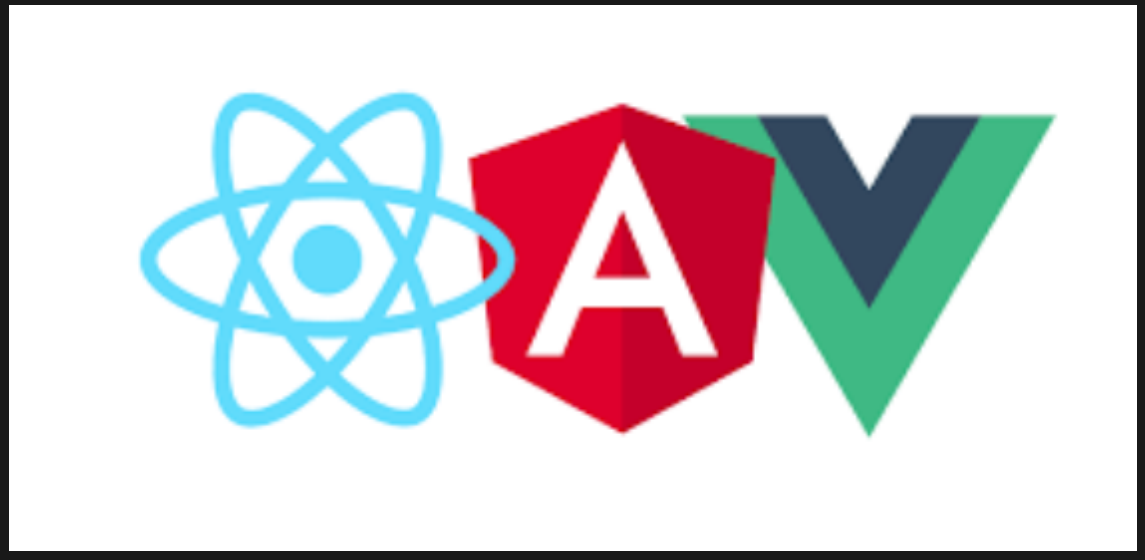
Type checking and common codebases - the beginning of heavy JS infrastructure

Electron, Apache Cordova

🏃‍♀️ The Progress In Web

6️⃣ 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

7️⃣ Mobile Native Apps & Hybrids

AI has vastly increase the rate at which web based software can be built

🏃‍♀️ The Progress In Web

7️⃣ Large language models

AI has vastly increase the rate at which web based software can be built

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.

📗 Summarised Learning Outcomes

- CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically
- CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation
- CLO3 : Use Javascript and CSS frameworks to allow more efficient integration of existing code and components into a final product
- CLO4 : Build stable applications that utilise concurrent programming through use of Javascript's asynchronous programming techniques
- CLO5 : Design and build interfaces that focus on best user experience and accessible design practices

🌻 Approach To Teaching

Firstly, this is a challenging level 6 course. It will be hard to perform in the HD bracket without hard work or natural intellect.

Secondly, learning front-end development will be a bit different from other languages you've learned:

- Front-end is a 'breadth' topic, which is unlike many other languages
- A lot less time is spent solving algorithmic or computational problems, and a lot more time is spent just trying to fine tune your visuals or find the right method/library/approach to get the expected behaviour
- You will still feel like you have a lot to learn by the end of the course

👀 Assessment Structure

There is no direct assessment associated with Lectures or Tutorials

Item	Due	Weighting
Assignment	Due week 3	20%
Quizzes	Throughout term	45%
Exam	Exam Period	35%

Teaching Methods

Let's explore our teaching methods

🟢 Lectures

- Avg 2 hours live lectures (ideally no new content, all demos, don't expect pre-prepared perfection)
- Avg 2 hours of pre-recorded per week (tightly broken up)
- Lectures are categorised into levels of importance for you.

🟡 Tutorials (Non-Quiz Weeks)

- You can attend any tutorial you want
- You can attend no tutorials if you wish
- All tutorials have been pre-recorded to learn at your own pace
- Tutorials are also your self-learning exercises
- Please respect your tutor probably needs to leave at the end

🟤 Tutorials (Quiz Weeks)

Let's review the "Quizzes" page together.

🔵 Help Sessions

- Chance to talk to tutors and get help on matters to do with tutorial exercises and assignments
- Help sessions will contain 1-5 tutors who will be split between assisting students with questions, and marking labs off
- Please pay attention to how many tutors are in a given help session - we do our best to predict demand and adjust but please attend help sessions being prepared to wait

🟣 Assignment - In-Depth Skills

Static HTML/CSS reproductions.

Programming.

4 days late penalty. 5% per day. 🔥

🧠 Assumed Knowledge

- Basics of GIT
- Basics of HTTP & Web Browsers
- High level understanding of scripting languages (either Javascript or Python)

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

If you lack this assumed knowledge, we have lectures to provide you all the help.

😍 Content We Cover

- HTML
- CSS
- Javascript (Vanilla)
- Javascript (Async)
- UI/UX/Accessibility
- Web Framework (ReactJS)

🥹 Content We Don't Cover (Much Or At All)

- Javascript classes
- Global state managers
- Advanced CSS (canvas, animation)
- CSS Compilers (.less, .scss)
- Native app development
- Web Assembly

💻 Course Site

Our course site is a custom content management system called Eckles. It focuses on flexible learning.

☃️ 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 ass1
- We will provide a demo of it's usage.
- There are git resources on Eckles.

Related Courses

- COMP3511: Human Computer Interaction
- COMP4511: User Interface Design and Construction
- COMP6443: Web Application Security & Testing

🙏🏾 Getting Help

If you need help with something, go here:

- Discourse (sidebar)
  - Note Hayden does not directly monitor
- Help Sessions
- cs6080@cse.unsw.edu.au

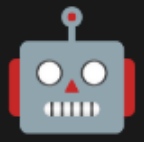
🤖 AI Usage Policy

Formally, AI is prohibited in assessments in this course.

In all assessments but one, it's not possible to use it anyway.

The final exam does not allow for the usage of LLMs.

Besides that, you are welcome to use them as assistants throughout the course.



AI Usage Policy

But there is more to talk about with AI...

👵 AI: So What's Even Going On?

Private companies have invested large amounts of money to develop language models that essentially mimic human thinking.

They train these models (tuning parameters in neural nets) with large amounts of information.

You then have a context window that you can provide to this model as input in order to get an output.

Model harnesses have improved, reasoning has been added, and MCPs are rampant. This makes them very powerful.

🔮 AI: Principles To Manage It

There are 3 important things to frame this AI change in

🔮 AI: Principles To Manage It

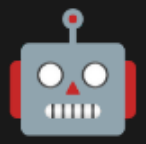
AI is a competing against humans, not against computers.

🔮 AI: Principles To Manage It

AI is fundamentally limited when it comes to warm body problems.

🔮 AI: Principles To Manage It

Education is unlikely to keep up (and it's not their fault).



Tips To Adapting To The AI World

Let's discuss some of my best advice on managing this AI era

🤖 1. Be Adaptable

Adaptability matters more than ever

New things matter every day

Old things that matter don't matter anymore every day

Don't be a slave to your pride

🤖 2. Writing Code Is Worthless (Kind Of)

Writing code = $0/hour

Reading code = Valauble (for now)

To read code, you need to write code

🤖 3. Build For People

Always start with the end user and work backwards from them

If you're solving their problems, and doing so economically, the rest will fall into place.

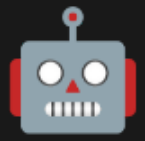
🤖 4. Domain Diversification

- Easier to other person person how to use AI than other way
- Two people, but one with CS skills, wins
- Drop into 4445 lectures or guest lectures
- Pick one domain this term and get good at it, suggestions would be:
  - Finance
  - Medicine
  - Business
  - Labor Market
  - (much more)

🤖 5. Be Top-Down With Your Approach

Try and solve your problems with AI first. Where it fails, use that as a guide to know to learn deeper.

(This is advice that won't always be useful in COMP6080)



6. Build-Analyse Cycle

Every time you use AI, ask yourself if you're using it right now as a **hammer** or a **tutor**. It should be a tutor until you're ready to use it for a hammer.

🤖 7. Learn The Basics And The Cutting Edge

- Learn the fundamentals
- Learn the cutting edge
- Plug the middle with the rest

🔮 The Future Of COMP6080

Let's do a thought experiment about where this course will be in 2 years so that you can put what we're doing today in context.

🔮 The Future Of COMP6080

No programming assignments

Assignment 1 exists because AI still sucks at it

🔮 The Future Of COMP6080

Probably significantly less time even looking at code, a much bigger focus on theory elements.

🔮 The Future Of COMP6080

Extra time free'd up introduces:

- Security
- Native mobile
- Product
- Business
- Cutting edge

⚒️ A Taste: "Making Web Browsers Do Things"

- Most things we do in this course are various combinations of:
  - **HTML**: Page structure
  - **CSS**: Page aesthetics
  - **Javascript**: Page dynamics
- Other topics we discuss are not about technology, but rather how to use it effectively:
  - Accessibility, product design, UI/UX, testing
- (Optional) Let's do a short demo to explore what is coming up in the first weeks

🦻 Feedback

Or go to the form here.