After INFO6150, now what?

- You have been web developers for weeks
- Always more to learn
- Easy to forget details
- Practice!
- Expand
- Tap into info sources

Practice

Best way to improve as a coder is to code

Make a public repo

- Contribute over time
- **https://blog.patricktriest.com/host-webapps-free/**

Contributions over time

- Teach you more
- Give you an interview topic
- Show value to employers

Showing off

Everyone has a resume and claims

Have something VISUAL

- Demo site(s)
- Show off SPECIFIC skills

You never know which visual trick excites them

Portfolio tips

- Don't show off assignment code
    - Lots of people can copy code from class
    - Your audience lacks context
- Build your HTML/CSS by hand
    - SASS/etc okay, but include your original
- Be explicit about what you will show
    - "Parallax background", "Adaptive mobile hamburger menu"
    - Assume they may not actually look at it
    - Just having it has value
        - Plus practice!

Expanding UI/UX knowledge

- More CSS
- More A11y
- More Interaction (JS/CSS)
- Advanced UI Patterns
- UI Prototyping Tools
- Available Libraries

CSS is central to Web UI/UX!

Two parts to learning

- Broader Understanding
    - More properties
    - More effects
- Deeper Understanding
    - Know what to expect
    - Know how to achieve effect

This class was a deep intro

- But still an intro

Also More CSS Systems and Organization

- CSS modules
- Tailwind for utility classes

More A11y

I expect these skills to be increasingly in demand

- And you get to feel good about your work

Goals:

- Use a screen reader
    - Use a second one
- Follow/read/listen to a11y/disabled users
    - Learn what counts for UX
        - Ex: Skiplinks used more by sighted users

Advanced UI Patterns

Learn to create

- And their effect on the UX
- Also: Learn their names
- **http://ui-patterns.com/**
- **https://careerfoundry.com/en/blog/ui-design/ui-element-glossary/**

More Interaction

JS + CSS allows for more than just CSS

- Many effects to learn how to do
- And what the UX impact of effects are
    - Ex: "tactile" feel from visual
    - Ex: Flat vs Depth from visuals
    - More ways to feel "intuitive"

Common tactic: Try to recreate effects you see

- WITHOUT copying their source!
- See: "CSS Battles"

Prototyping Tools

Software (web or desktop) that creates prototypes

- Mix of Photoshop and web-page generation
- Output is Prototype site/page
    - Not a "real" site
    - See Mockups/etc discussion
- **Figma**
    - Current industry favorite
        - Bought by Adobe, future uncertain
- Proto.io
    - An alternative to Figma

Learn some Libraries

- Libraries make your dev time more efficient
    - But introduce limitations
- Learn the concepts, before or after
    - No library lasts forever
        - Rarely lasts long, in web dev
- Ex: Form handling (Ex: Formik)
- Ex: Specific UI Patterns (Ex: date-picker)
- Ex: Routing + deeplinking (Ex: @tanstack/router)
- Ex: CSS preprocessor (Ex: Sass)

Learn other HTML-generation options

- Non-React library/framework
    - Ex: Vue, Svelte
- Non-Create React App boilerplate generation
    - Ex: Vite
- Server Side Generation/Rendering
    - Ex: NextJS, Remix

Expand Web Dev knowledge

Web UX Engineer = Web Design + Front End Web Dev

Front End Web is increasingly using services

- Gives Front End access to changing server data

Options to improve:

- Learn service calls
- Learn state management
- Learn back end development
- Learn TypeScript
- Learn Automated Testing

Information Sources

Web Tech changes rapidly

- How do you learn about new options?
- How do you learn new best practices?

My experience: You have to CHASE this

- Easy to get bubbled into what you do now
- New devs generate a lot of intro "noise"
    - Great for newbie perspective
    - Often contradictory
    - Sometimes bad advice

Finding Good Sources

- Sometimes good people get celebrity
- Other times you have to find them
    - Both tech and search engines have biases!
    - Keep your eyes open
- When you find someone good
    - How will you know their next bit of info?
    - Subscribe, follow, whatever
- "Curate" your sources often
    - Sources will get stale over time

Twitter! Oh no!

Previous best source was Twitter

- Now in meltdown/exodus
- See where people go

Samples from my list:

```
@geekgalgroks  @cassidoo
@zeldman       @laurieontech
@jaffathecake  @JoshWComeau
@rmurphey      @b0rk
@meyerweb      @TerribleMia
@jen4web       @darkpatterns
```

Blogs and RSS

Longform content is still important

I use RSS to keep track of blogs

- **https://overreacted.io/**
- **https://www.joshwcomeau.com/**
- **https://css-tricks.com/**
- **https://adrianroselli.com/**

Videos, Podcasts, and Streams

I'm a sucker for text, but others prefer other formats

- Same principles: Filter, Follow, and Curate

Samples:

- HTTP 203 Podcast
- Ladybug Podcast (paused, but good backlog)
- **https://www.youtube.com/kevinpowell**
- **https://youtu.be/wPcv9Rp4lHk**

Conferences

- A few days dedicated a common topic
    - Great sources of info and inspiration
    - So many topics!
    - So good for extroverts!
        - Networking and friends
- Many conferences stream their talks
- Many conferences later post them online for free
    - Even for those that didn't attend in any way
    - So good for introverts!

Bumps Ahead

- Imposter Syndrome
- Gatekeeping
- Web Hate
- Burnout

Imposter Syndrome is very common in WebDev

You are a fraud. When they find out your career and reputation are ruined

Remember

- Everyone knows only a tiny slice
- The more you know, the more aware of ignorance
- Arrogant people can't be trusted
- You've done a lot
    - You aren't done

Keep the joy of learning

Gatekeeping

Unfortunate reality of people

- "HTML/CSS isn't a real language/real programming"
- "You aren't cut out for programming"
- "you just make it look pretty"

...but not reality of actual reality

- Why was Web successful? Why not replaced?
- Why do they want to gatekeep you out?

Web Hate

A particular brand of gatekeeping

- "web dev is trash programming"
- "JS? Haha!"

This can really tear into you

Caring makes you a great dev

- Caring makes you vulnerable

Burnout is very common in the industry

- Not: I need a vacation
- Instead: I'm mentally traumatized
    - Physical + mental symptoms
    - Can take months/years to recover (if at all)
        - Recovery requires lack of triggers
- The more you care about the work
- The less control you have over the priorities
- The more personally you take any imperfection
    - The more it piles on
- Can rob your ability to function

Be Nice

It can be nasty out there

- Particularly for non-men, non-whites
    - No one is totally safe
- Protect yourself
- Never think it is deserved
- Don't add to it
- Stand up for others
    - Particularly if you're not marginalized