Welcome to INFO6150

INFO6150: Web Design/User Experience

Why listen to me?

Brett Ritter

- He/him
- Web developer since 1995
    - This means I am old
- Multiple languages/frameworks/employers
- I am very funny, trust me

Not a trained educator!

- Teaching part time since 2017
- Help me improve

What does this course cover

- No experience assumed
- HTML
- CSS
- JavaScript to modify HTML
    - React intro
- Responsive/Adaptive design
- Some Common UI patterns
    - layouts, modals, spinners, forms, slide-in/out, menus, icons, transitions, etc
- Intro Accessibility (a11y)
- Intro, not exhaustive

What this course does NOT cover

- Internationaliztion (i18n)
- Backend Services
- Backend HTML generation
    - For these see the 6250 course
    - 6150 not a prereq for 6250, but will enhance
- Advanced React

How this course works

- Weekly assignments
    - Assigned and submitted via github
- Weekly short Canvas quizzes
- Midterm project
    - Just like assignments
    - 25% of grade!
- Final project
    - Just like assignments
    - 25% of grade!
- Slack support and discussion

Focus of this course

You will not be a designer, but you will be able to design web pages and be familiar with common practices.

Heavy emphasis on HTML and CSS, just a touch of JavaScript (JS)

Notice that JavaScript, despite the name, has no relationship to Java

Course Warnings

- Do *NOT* copy any work
    - classmate OR online
    - Penalties are severe!

Material I provide is fine, but *understand* it, don't just repeat it

Online information

Searching online is a skill you will never stop using

But there are pitfalls for Webdev

- Things changed quickly after a long time of not
    - Blame Internet Explorer for much of it

Vital lessons

- Do NOT use online sources older than 3 years
    - just too much outdated advice
- Do NOT copy answers
- Google "MDN (whatever)" is a good start

Livecast Attendance

- The school has legal obligations about physical attendance
- You are required to physically attend almost every class you aren't excused from
    - Otherwise you fail the course
- If requesting an excuse, _I_ trust you, but you should document for safety

COVID Overview

- I try to be as flexible as is workable anyway
- COVID sucks
    - We literally DO NOT KNOW the full impact
    - Even if "minimal", now is not the time to relax
- Thank you for keeping yourselves and others safe
- I will assist with this whenever possible

Important notes for this course

- Communication is *essential*.
    - confused, have doubts, curious?
    - need accommodations (temp or long-term)
    - please ask!
- Falling behind is a trap
    - address it early
    - hard work can't conjure time
- Most assignments require experimentation
    - it can be daunting

Frequently asked questions

- Windows or Mac?
    - Course: Either. I use Mac
    - Workplace: Depends on field
    - Command-line familiarity will be very helpful
- What IDE/editor?
    - Any
    - VSCode (no cost) is currently most popular
    - I use mostly vim

Frequently Asked Questions, part 2

- What browser?
    - Use: Anything modern
        - Firefox, Chrome, Safari, Edge
    - Graded: I teach and grade based on Chrome
- Classes recorded?
    - I will be trying to capture classes, but mistakes can happen

Slack

We communicate via Slack

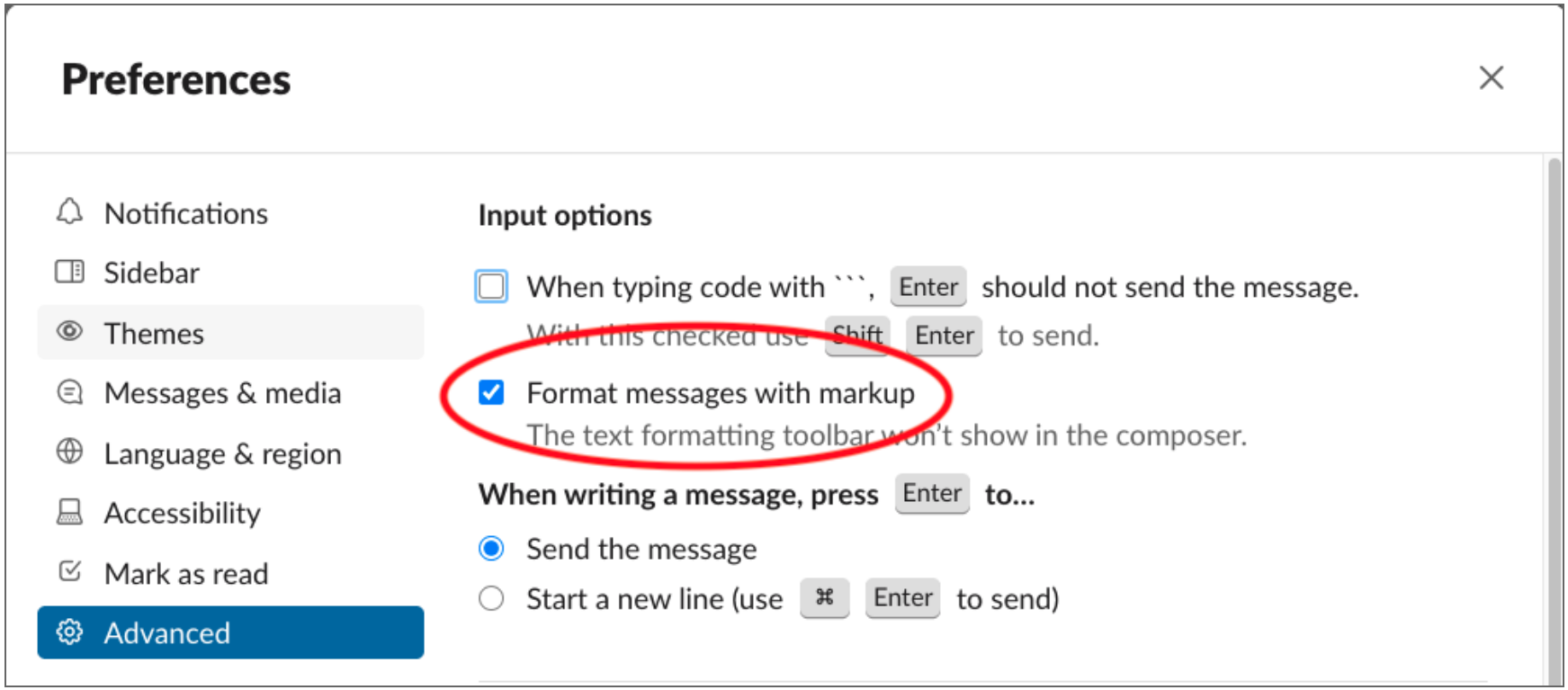
- **http://rebrand.ly/seainfo6150-slack**
- You can email me
    - but mail will be slower
    - Outlook is terrible to talk about code
- Use Slack on web, desktop, or phone
    - desktop/phone recommended for notifications
- Slack is a useful job skill

Using Slack

Respond to the question

Configuring Slack for code

Update your Slack Preferences:



Mentioning code in Slack

Send a message to **#questions** that says:

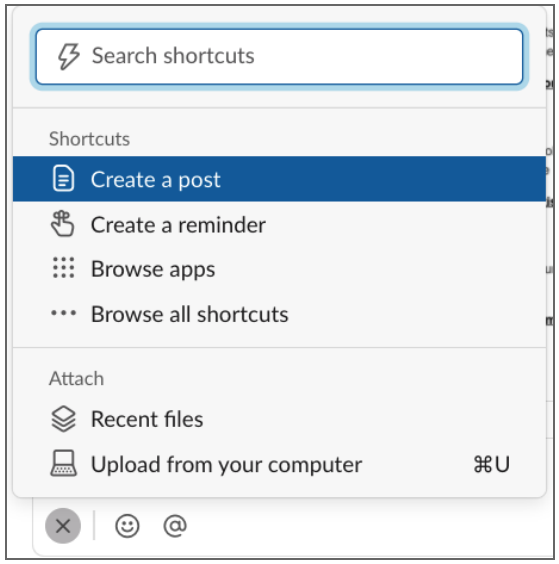
(notice the *backtick* characters)

`this has *bold*` but `` `this does *not*` ``

Code blocks in Slack

triple backticks ``` before and after your message

Or use "Create a post"



Screenshots can be hard to read if more than 1-2 lines

Class Github

We use git and github.com

Each of you get a personal repository

- **https://rebrand.ly/seainfo6150-github**

You must have/get a github.com account

Git vs Github

`git` is the source control/version control system.

- tracks files
- changes to files
- many devs can have repos
- can pass files/changes between many repos

`github.com` provides a central place for repos. It has competitors (example: gitlab.com)

github uses git, git does not require github or a github competitor, though we will use github.

Github flow

git is *decentralized*. github provides centralization(ish)

See `readings/basic-git.md` in your repo

- You make changes in a "feature branch" locally.
- You send that branch to github
- You create a Pull Request (PR) to merge your branch into main **on github**
- I/TA review and approve your request
    - We might request changes first
- I/TA merge your branch into main
    - On the job you will probably do this
- You update your local main

Other git-based flows

Other flows of changes and branches exist

- this one is the one we will use

Not all version control systems (VCS) are decentralized the way git is.

Github acts as a central point for communications

Local and remotes

"Local" means your computer (or anyone's)

When I give notes or assignments:

- I pull latest from github to my local copy
- I update my local copy
- I push the changes the github

You submit the same way:

- You pull changes from github to your local copy
- You make changes to your local copy
- You push the changes to github and create PR

Key Git Notes

- Always do work in the correct branch
- Always check `git status` before `git commit`
- Always check `git status` before `git push`
- When creating a PR, always check the file list
- Before creating a new feature branch
    - Always switch to `main` and pull latest

If you follow these instructions

- each assignment is distinct and will not conflict.

Summary - Class

Goal: Solid intro to web UI/UX

- both how to do it and why/why not

Not Goal: Everything web dev

- In particular, no backend/server work

Summary - Tools

- VSCode is most common Editor/IDE, anything allowed
    - On your own to fix issues though
- Demoed and graded using Chrome
    - But any modern browser *should* work
- Git and github.com to track and submit work
- Slack for in-class and out-of-class work
- Don't forget the `readings/` section of repo

Summary - Grades

Grades:

- Midterm and Final
    - Solo
    - Big chunk of course grade
- Test skills *from this course*
- Weekly assignments via repo on github
- Weekly quiz via Canvas
    - open book, not timed

Summary - Communication

- Always communicate!
    - Be careful about falling behind
- Slack at any time
    - if it is a bad time, I won't respond until later
- TA Office Hours (probably virtual) TBD
- Instructor Virtual Office Hours TBD

Summary - Warnings

- If you fall behind, speak up
- Don't trust older online info
- Many ways "work"
    - You are graded on material from course