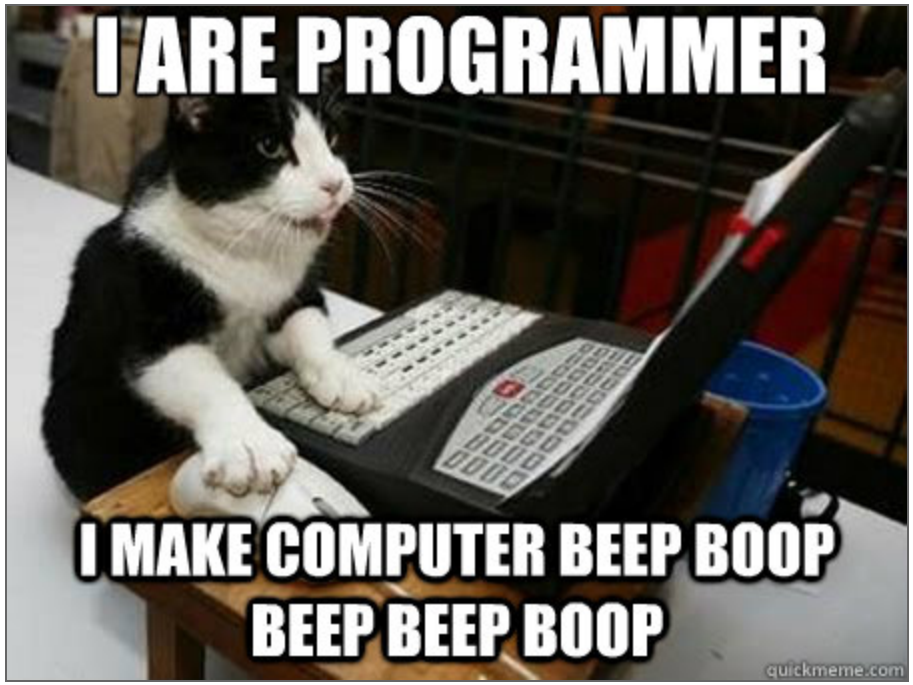
# Welcome to INFO6250

Web Development Tools and Methods



# Why the instructor is credible

Brett Ritter <b.ritter@neu.edu>

- first job as WebDev in 1995
- webdev ever since
- multiple languages, frameworks, platforms
- both frontend and backend
- I've been a full-time webdev, part-time teacher
    - tell me where to improve my teaching

# Not Perfect

I have a truly terrible memory.

**Terrible**

You have my permission to remind me, and keep reminding me, until something is done or I explicitly say "stop".

I am partially tone-deaf.

If I ask you to repeat yourself, my poor hearing is why.

# Funny

I tell jokes.

Fortunately, they are all hilarious and you will laugh.

Out loud.

Try it now.

# Get Better

We will keep practicing on that

# Web Development Tools and Methods

- The Hows and Whys of the Web
- Building a Multiple Page Web Application
    - NodeJS backend
    - HTML
    - CSS
    - JS frontend
- Building REST-based services (NodeJS backend)
- Building a Single Page Application (using React)
- Basic Best Practices for the Web

# Class is hard, worthwhile

- We cover a lot of material
- You must learn to apply new concepts
- A lot of time and work
- Goal is maximum preparation

# Cannot teach it all

Too much to cover

- You end empowered to continue your education

# Teaching Fundamentals

Pros:

- What you build from
- Fewer "surprise" gaps

Cons:

- Longer path to "wow!"
- Very rushed class

# Ask Questions

Everyone learns differently

I have terrible memory

Ask Questions

# Trying things

Best answers can be found by trying things

I set you up to experiment

# It Depends

Most common answer: 'It Depends'

**IS NOT**: Does not matter

**IS**: Depends on what?

**MAY BE**: We are still figuring that out

# Assignments

Assignments will NOT be "copy what I did"

Instead, will be "apply the skills in a new way"

Leave time to do work!

# Slack

We communicate via Slack

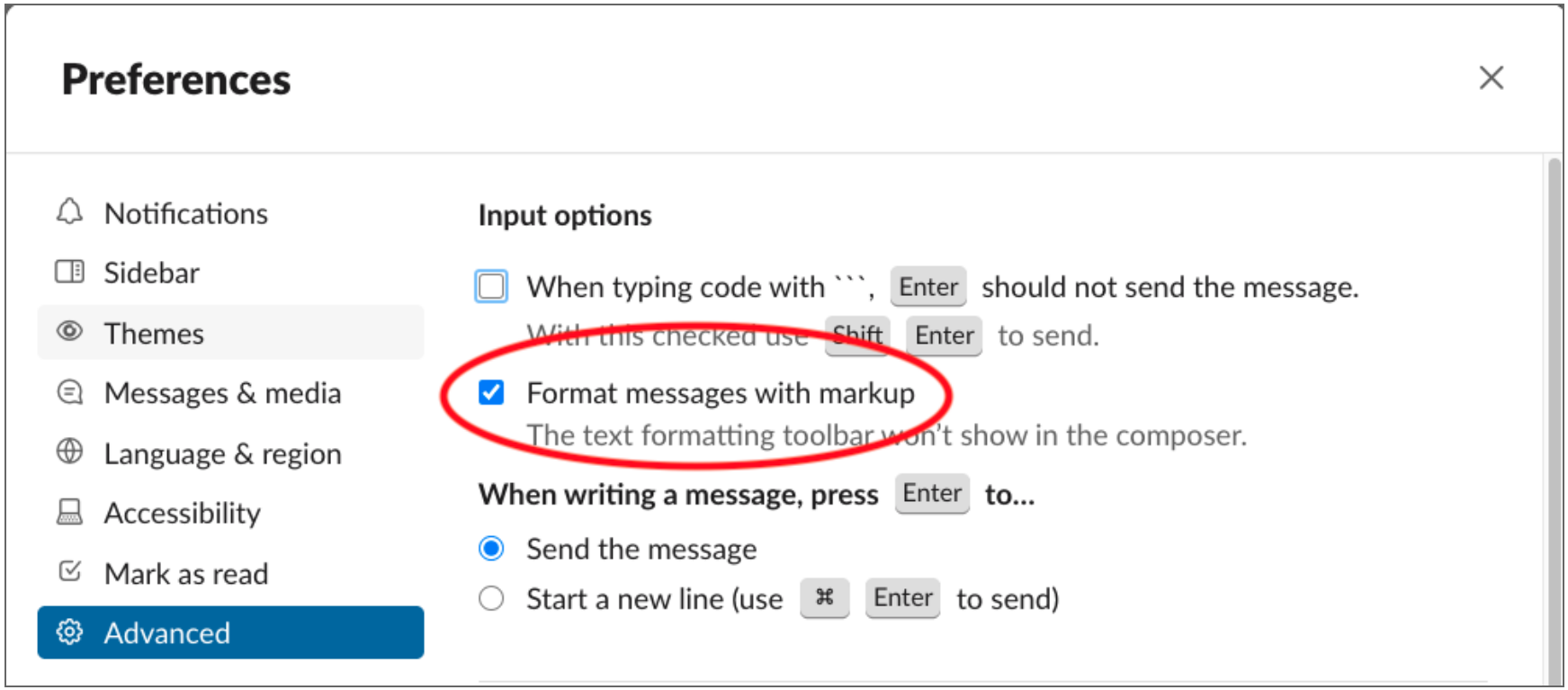
- **http://rebrand.ly/seainfo6250-slack**
- You can email me
    - but it will be slower
    - and it is terrible to talk about code
- Use it web, desktop, or phone
    - desktop/phone recommended for notifications
- This is a useful job skill

# 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:

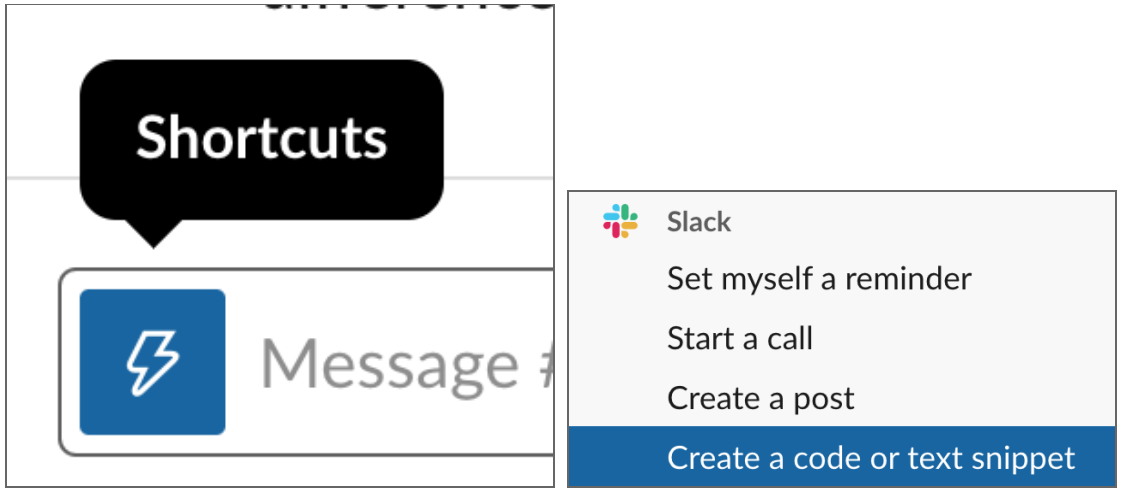
(notice the *backtick* characters)

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

# Code blocks in Slack

triple backticks ``` before and after your message

Or use "Create a code or text snippet"



# Class Github

We use git and github.com

Each of you get a personal repository

- **https://rebrand.ly/seainfo6250-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.

# How the Class works

3 Sections

- Assignment each class
- Quiz each class

Midterm project

Final Project at end of Semester

# How a Class works

- Lectures, sometimes labs (ungraded)
    - Will check progress via Slack
    - Ask questions via Slack
- Bio break 1/class
- Recorded online (if it works!)
- Slides as PDF added to repos
    - Possibly before, definitely after
- Sometimes samples to repos
- Assignment after each class

# How a Section works

- Classes and assignments
- Quizzes
- Exam at end of section
- 3 Sections
    - Knowledge builds on previous

# How Assignments work

- Build from skills
    - Leave time!
- Roughly 1 week to submit
- Added to repos under `/work` (see README)
- Each assignment is a different subdirectory
- Submitted via github Pull Request
- TA/I will review and merge
    - May request changes
- No changes unless requested
- Lowest score dropped

# How Quizzes work

In Canvas

- Multiple choice or short answer
- Covers materials from the class
- You are welcome to consult notes, recordings, etc
    - Do NOT copy from other people though (incl online)

# How Projects work

- Like assignments
    - Pull request, Due date
    - Done at home, not in class
- In repos under `/midterm` or `/final`
- Big chunk of grade each

# How the Final Project works

- Solo work
- Guidelines given
- Full React SPA + REST services
    - Minimal outside libraries
- Potential Showcase for NEU
- Submitted as Pull Request in repo
    - in `/final`
- Limited time!
- Chance to raise grade

# Other Details

- No video requirement
- Probably no breakout rooms
- Don't really use Zoom Chat
- Still limited Canvas use
- Slack preferred over Email
- No set Office Hours for Instructor yet
    - But can always request!
    - Quick Slack chats are common!
- TA Virtual Office Hours TBD

# Important Details

- These are not normal times, I understand
- Request Assignment extensions, no excuse required
    - But request IN ADVANCE!
- DO NOT COPY WORK YOU SUBMIT

Final Projects do NOT get extensions barring emergencies

# Do and Do Not

- DO ask questions
- DO not worry about bothering me
    - I will tell you BEFORE that happens
- DO NOT worry about looking uninformed
    - You are literally students
- DO NOT expect to catch up by working harder
    - Good attitude, but…
    - Time, not you, is the problem
    - Easier to fix earlier rather than later
    - Your employer will follow the same rules

# A Unique Warning about the Web

You WILL be expected to learn a lot of detail online

BUT a **lot** of info about web tech is outdated

- Don't use any sources older than 3 years ago

Really. **3 years max**. Or extra work and wrong work.

# Common Questions

- Do I need to know HTML/CSS?
- Do I need to know programming?
- Do I need to know JS?
- Can I use another language?
- Can I use this outside library?
- Can I use this other framework?
- Can I use this other IDE?
- Can the instructor send class materials out?

# Class Prerequisites

- Do I need to know HTML/CSS?
  - You are expected to know the basics
  - The 6150 class is great and recommended
  - You CAN learn alongside class (but **harder**)
  - **https://developer.mozilla.org/en-US/docs/Learn**

# Class Prerequisites (cont)

- Do I need to know programming?
    - Basics (variables, conditionals, looping, functions) are expected
    - Be ready to learn how things are different
- Do I need to know JS (Javascript)?
    - No expectation

# Using other tools

- Can I use another language?
    - No, we use only JS for my sanity
    - Lessons are general!
- Can I use Typescript?
    - No, we rely on JS fundamentals
    - TS is fine, just not part of the class

# Using other tools (cont)

- Can I use this outside library?
    - (example: Bootstrap)
    - No, we need to exercise the fundamentals
    - Exceptions are explicit per assignment
- Can I use this other framework?
    - (example: Angular/Vue)
    - No, we use React for core principles
        - and my sanity
- Can I use this other IDE?
    - Yes, but I don't know that IDE

# Class Materials

## Can the instructor send class materials out?

Slides and code samples will be added to your repositories after class.

- `git checkout main`
- `git pull origin main`

If you feel something is missing, ask for it via Slack.

BUT - anything that comes in the moment may not be saved to send out. Missing class is risky.

# What to expect

Section 1:

- Lots of fundamental concepts
- Rushing HTML/CSS/JS

Section 2:

- Network and services
- Asynchronous code

Section 3:

- Modern React
- Architectural concepts
- Job skills

# Summary - You

- will **remind me** if you need something
- can and will **use and check** our Slack
- have your own **github repo**
- have a **local copy** of the repo
- have installed and done **config** changes
- know how to **get** and **submit** work
- have **submitted a PR** that provides your info
- will **not use old** online information
- will **read** material
- will **ask questions**