

# W1- The Dev Workflow

# AGENDA

Intro

Curriculum Overview

Approach to lectures

Tools

Version Control

Incremental development

# Know your Instructor

Sandeep Chopra

Development Lead - Frontend / Full-Stack

Xerris Inc. (Calgary)

Part-time Instructor- Web Flex, Lighthouse Labs



# Quirky Fact

# Introductions

Your name, city, and brief about your background and quirky fact

# Curriculum Overview

# Approach to Lectures

## Lecture over Zoom:

- Instructor will provide a Zoom link in your slack channels 10-15 minutes before lecture. You're welcome to join to just chill and have a chat.

**Expect Lectures to be 2- 2.5 hours**

# Approach to Lectures

Lecture over Zoom:

- Try to have your camera turned on. We would like the lecture to be engaging!
- Lecture notes, code, and video recording are going to be sent out after lecture.



# Approach to Lectures

- Mix a theory and practice, more practice.
- Provide context and explain why.
- More code demonstration (like pair programming).
- Focused on the approach
  - Problem Solving
  - Step by step incremental development
  - Error driven development

# Approach to Lectures

## Questions

To ask a question:

- Raise your hand (ALT-Y) or use the chat
- Please, leave the chat to the instructor for questions
- You can ask questions during office hours

# Approach to Lectures

What lectures are NOT:

- Coding along session
- Do your daily activities at the same time

# Tools

- Shortcuts (Learn your shortcuts!! Don't use the mouse!)

VS Code Cheat Sheet:

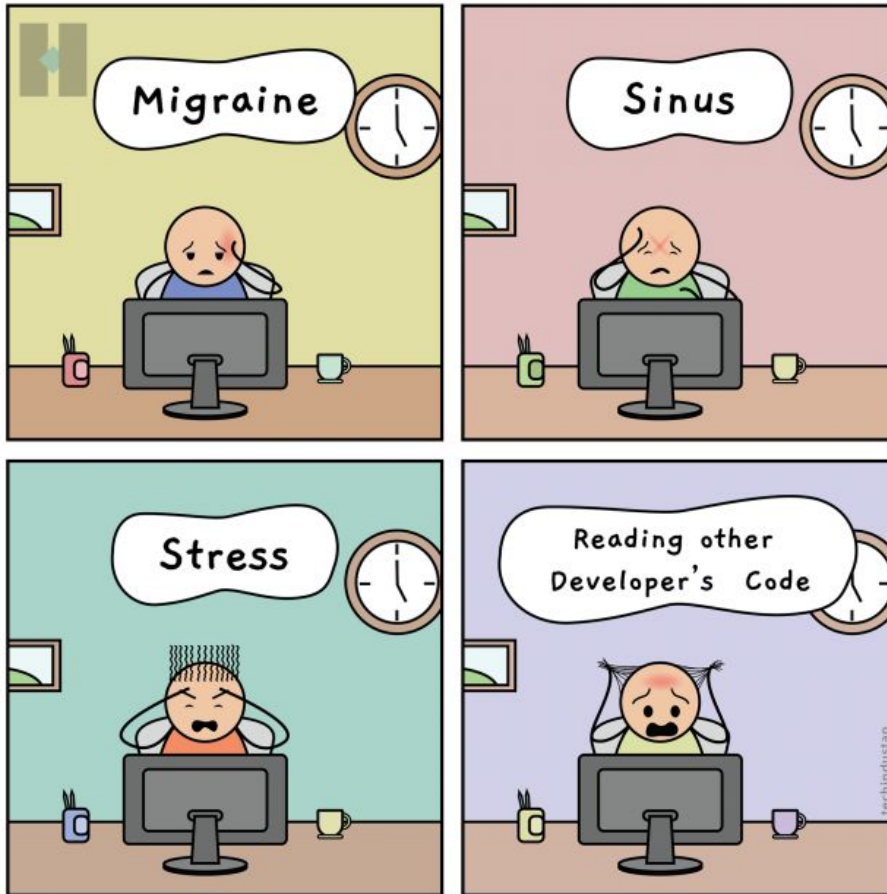
- <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-macos.pdf>
- <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf>
- <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-linux.pdf>
- Useful Add-Ons
  - ESLint
  - Bracket Matching
  - Prettier (but not for first few weeks)
- Google
  - Good habit to search for a solution (Stack Overflow)

# Incremental development

How to approach problem solving

- List the steps in order to solve a problem. Not thinking about the syntax.
- Step-by-step process:
  01. State the hypothesis
  02. Verify the hypothesis
  03. Make changes

## Types Of Headaches



- As developers, we express ourselves through code much like an author writing a book.
- Much like an author, we are writing code for others to understand.

# Version Control - GIT

What, Why git?

- Repositories (one repo per projects)
- Save milestones
- Keeps an history of your code (commits)
- Backup copy on github
- Work better as teams, branches
- Do use git
- You **will have to use** git in team projects

# Version Control - GIT

- GIT Workflow (add files to staging area, commit changes, update github)
- GIT Commands:
  - `git status`
  - `git add .`
  - `git commit -m "message"`
  - `git remote -v` (or add origin, rm origin)
  - `git push`
  - `git pull`
  - `git log`



# DEMO

Write a node program that takes in an unlimited number of command line arguments, goes through each and prints out the sum of them. If any argument is not a whole number, skip it. Do support negative numbers though. If any argument is not a number, output an error message. We need at least 2 arguments.

# Questions?