Overview

- Ask lots of questions from class. As much interaction as possible.

Learnings from portfolio

Self-care

- Marathon mentality. Chip away day by day. All nighters are a last resort.
- When you're feeling stressed or stuck, take a break. *give example of riding bike home*
- When you're cognitively drained but still want to be productive, do something easier like listening to a podcast or reading medium articles.

Software development

- MVP philosophy: start small and iterate. Build up piece by piece. If you start big and it breaks, you don't know which part has broken.
- feature creep. Define early on what MVP is, and build that first. Write a list of nice-to-haves, and do those *after* the MVP is built and deployed. Donuts and sprinkles analogy
- Be selective with frameworks and other external resources. 1. if they break, you need to fix it 2. they're not created equal

WSL, servers and OS's

- What a server is (just a computer, usually running linux)
- OS structure, concept of a kernel
- Linux kernel, dominance of MacOS
- WSL and what it is/how it works

Programming intro

- Overview
    - Programming takes a while to get used to
    - The Ed challenges are just an intro. You need to do MORE. This is like learning a new language: practice, practice, practice
        - edabit
        - ask us for more exercises
    - There's heaps of new concepts and jargon here, plus a whole new way of thinking. Don't stress! Growth mindset!
    - two separate things we have to learn:
        - procedural/algorithmic thinking
        - ruby concepts and syntax, so we can translate algorithm into a language the computer can understand

- vegemite sandwich demo

- procedural thinking

    - what is programming
    - what is an algorithm

- breaking down a problem

    - flowchart
    - pseudocode
    - eg for making vegemite toast, with check for doneness (use draw.io)

Ruby review

- lots of jargon and concepts. It'll take a while to sink in
- What is Ruby?
    - Created in 1995 by Yukihiro “Matz” Matsumoto
    - designed to be human-friendly and intuitive
- why are we learning Ruby?
    - beginner-friendly language
    - Ruby on Rails is easiest framework to develop web applications
- irb (do in Ed, and recommend code-along)
- data types
    - int, float

    ```
    > 4.0 / 5
    => 0.8 #why?
    > 4 / 5
    => 0 #why? get everyone to try
    ```

    - arithmetic operators (+ - * / % **). Mention precedence
        - get everyone to try 4 + 5 * 3
    - variables (for fun, mention hiragana syntax)
        - reassignment
        - += and -=
        - constants (starts with capital, often uppercase). can change in Ruby, just a convention, but will warn
    - strings
        - concatenation
        - interpolation (show both simple variable, and full expression)
    - type coercion
    - boolean
    - nil
    - arrays
    - hash tables
    - symbols
- source code
- I/O
    - p vs puts vs print
    - a program that greets the user

    - a program that calculates BMI
- control flow
    - if else
        - a program that tell you to wear sunblock/raincoat (do flowchart first)
    - ternary
    - case

```
case var
when 0

when 1

else

end
```

- logical operators
    - and &&
    - or ||
    - not !
    - program that tells you to go to bed if tired && it's night
- truthy and falsey (only nil and false are false. 0 is weird in Ruby)
- loops (first show non-dry example with repetition)
    - while. program that does rocket ship countdown
    - for (only for collection in Ruby eg range)
    - next and break

Terminal app

- Have an idea for a basic terminal app. test to quiz ruby knowledge? cafe?
- Write its spec
- write its flow out formally in a flow chart
- translate the flow chart to pseudocode
- translate the pseudocode into functional Ruby

Schedule

Thu morning

overview

- Ruby kahoot
- portfolio learnings
- WSL overview
- Ruby review
- intro to programming: algorithms, procedural thinking

Thu afternoon

- Ruby install session

Challenges:

- program that tells you if you can vote, based on age

Stuff for weekly meeting

- Why is SRI in the portfolio assignment? It's too advanced for their level, creates a huge amount of busywork, and isn't widely used in industry.
- print out cheat sheets?

Fri morning

- Work through building terminal app

Fri arvo

- review and/or masterclass?