Intro to Junior Phase

Pair Programming

- 2 programmers working collaboratively
  - **Driver**
    - writes the code
  - **Navigator**
    - reviews code
    - researches possible solutions
      - (e.g. array prototypal methods, proper syntax, etc)
  - **Both**
    - brainstorm how to solve problems
- Learning how to
  - manage frustration
  - rally after failures
  - form effective partnerships

Why Pairing?

- **Economics**
  - 15% more time on programs than individuals **but** the resulting code has 15% fewer defects.
  - Defects are costly both in time and money
- **Design Quality —** More diverse solutions
  - variety of experience
  - interpretation of code is different
  - roles differ
- **Satisfaction**
  - Average increased enjoyment in work
  - Average reported increase in confidence of code
- **Learning —** Invaluable feedback, tips and overall design skill
- **Team Building, Communication and Transparency**

Debugging

- **Read error messages**
  - Refer to line of code error was thrown at
  - Research this error online
- **`Console.log`**
  - Add text to each log so you know when it is being printed
- **`Debugger`**
  - chrome dev tools allows you to walk through the code line by line
- **Ask for help…**

Asking for Help

- **During workshops -** Help Desk
  - Make a descriptive help ticket
    - e.g. We are on X part of the workshop. The prototypal method we created is running, but it won't stop, so we are exceeding the maximum call stack.
- **Outside of workshop hours**
  - Use slack to message your instructors and fellow!
  - Request office hours from staff directly
- **With non-technical issues**
  - Incident Report Form
  - Feedback and check-ins

Getting Started

- Git Pairing Tips!
- Environment Setup - Toolbox Workshop
- Your cohort repo
- Junior Phase Outline