- Shoutout to newbies

- **All Also's views, not Google**
- **Works with cloud dataproc**
  - Managed Big data clusters
    - Big data is SO MUCH DATA it can't be processed. Divvies up work between multiple machines.
    - Rent out servers as needed.Y
    - Configures clusters of machines.
  - open source project
  - public facing toolchains.
- **How did he get there?**
  - started in 2013
  - Got masters in CS
  - Worked for Blacksage
  - Big Data consulting at zData
  - Google siren call
    - build infrastructure that supports Big Data structures
- **Develop Patience**
  - You have to suck at something before you get sort of good at something.
  - There is so much to learn in CS.
    - You can never know everything.
    - Don't get burned out because you don't know something.
      - don't let it bother you
- **Manage Complexity**
  - Code gets complex fast.
  - One of the main skills of a Software Engineer is learning how to organise code.
  - How to organise and keep one concept simple w/in your scope
  - Be humble.
    - Humans are dumb in the scope of math in the world.
    - Keep cleaning your rooms
- **Find opportunities to celebrate.**
  - Going from beginner to expert is a slog
  - 10,000 hours of focused work on a thing to become an expert in something.
  - Takes a long time of steady progress.
  - Find stuff to get excited about.
    - There are so many paths in CS.
      - Find something that you think is rad

[Web design is a good place to start. Javascript components EVERYWHERE]

- **Help each other out**
  - SE is a team sport.
  - Cut out the ego.
    - you need to be able to work with people, no matter how smart you are.
    - everyone is at different levels.
      - Ven diagram of people's knowledge
    - Share, be open, have each other's backs.
- **Q&A**
  - Always a ramp up
    - First 3 months is mostly learning.
    - being able to ingest new tech is very good.
  - Interview practice
    - Drilled programming exercises.
    - Based on academic CS. Data structures etc.
    - Spent an hour a day on CS practice.
    - If you want to get good, practice.
    - Skills you use in interviewing not exactly what you will be doing in day to day.
      - Focuses on knowing CS fundamentals.
      - Knowing that you get what makes Software Engineering efficient.
  - How does the academic side of CS compare to the application
    - There are plenty of people who are v successful w/o formal education.
    - Started as a freelance web designer.
      - having the math background helped. Made it easy to get serious about programming.
    - Theoretical CS fundamentals very helpful for going really deep when you want to.
  - What are drones up to these days
    - flying drones in areas where airplanes also fly
      - Very dangerous, but cool pix
    - Stadiums also good filming shot.
      - If a drone runs out of battery while it's flying, it becomes a brick, can fall on people.

        - Could be used to drop grenades or disperse chemical weapons.
    - What's your favourite language?
        - Rust
        - Likes functional programming.
            - Not mainstreaming
        - Javascript, C++, python etc conform to similar concepts and syntax.
        - Purescript, useless , but cool high-level concepts
    - Where should we start to be able to succeed
        - Front end interface.
            - Being able to roll out a web page and have a UI is SUPER important.
        - No idea how to read the market.
            - Stuff that's hot right now may
            - Systems engineer.
                - writing kernals etc.
            - Staying on the front end.
                - Rapidly knocking out Websites and mobile interfaces.
    - Should beginners learn a lot at first?
        - Go narrow at first.
            - change focus if you find something you love
        - Knowledge Portfolio
            - When you learn something it's like buying stock
            - Coordinate your investments
                - Learn things that work well together.
    - What to do with someone with basics but no real world experience?
        - Process of learning is three fold
          1) You Learn a thing - what are the things
          2) Contemplation - How do I use a component?
          3) Mediation - practice a lot. Be bad at it first.
        - There is a point where you have to set down the learning and use what you've learned.
- Boy likes math
    - What classes to take?
        - Statistics is good to take.