

# Software Archaeology and Anthropology

17-313 Fall 2024

Foundations of Software Engineering

<https://cmu-17313q.github.io>

Eduardo Feo Flushing

# Administrivia

- Slack
  - Please add a profile picture.
  - Ask questions in #general or #technical-questions.
    - Please use threads.
    - Use the search tool
- Office hours can be found on the course home page:  
<http://cmu-17313q.github.io>

# Smoking Section

- Last two rows



# Team Formation Survey due Thursday

- Form groups based on schedule availability.
  - This is ridiculously important.
  - Identify experience and working styles.
  - Participation point
- Google Form, posted on slack

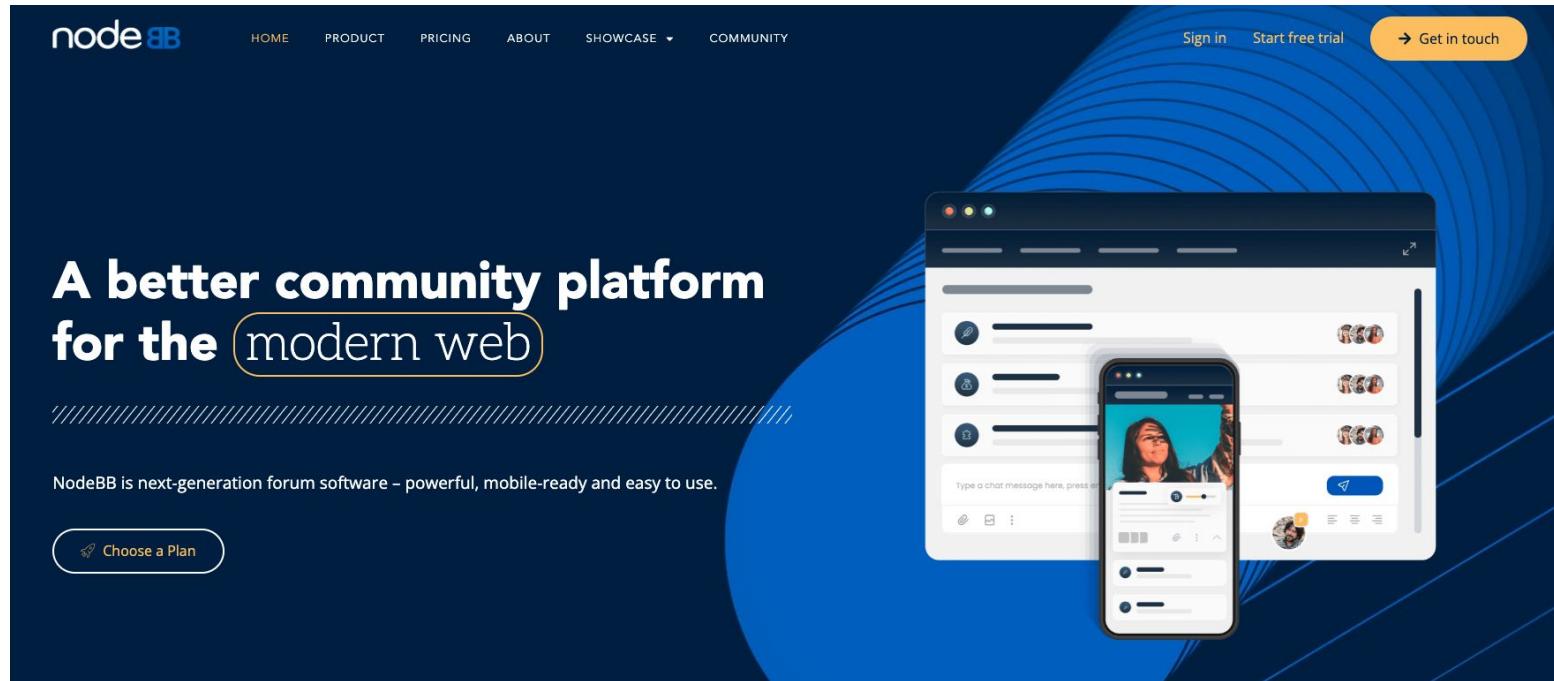
# Project P1



- **P1A:** Checkpoint due next Sunday (September 1<sup>st</sup>)
  - Only 5% of total P1 points – meant to ensure you start on time
- **P1B:** Due Thursday next week, September 5<sup>th</sup>)
  - Refactor a javascript file to improve its quality
  - It will be posted tomorrow
  - Start early

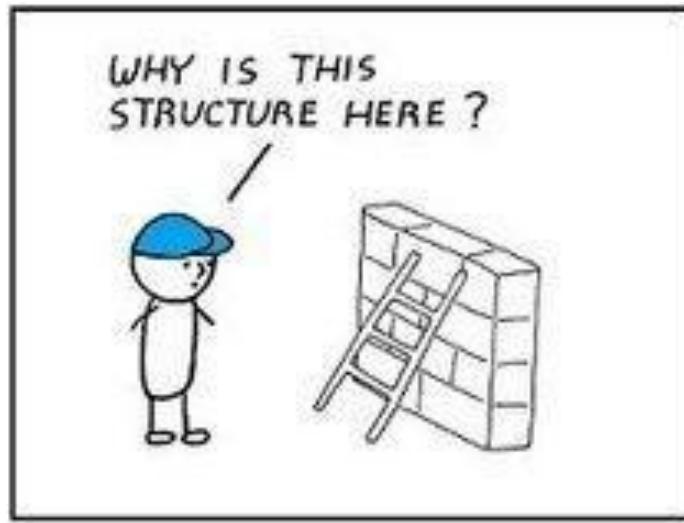
# Context: big old pile of code

- ... do something with it!



You will never  
understand the  
entire system!

# Challenge: How do I tackle this codebase?



# Challenge: How do I tackle this codebase?

- Leverage your previous experiences (languages, technologies, patterns)
- Consult documentation, whitepapers
- Talk to experts, code owners
- Follow best practices to build a working model of the system

# Bad news: There are few helpful resources!

- **Working Effectively with Legacy Code.**

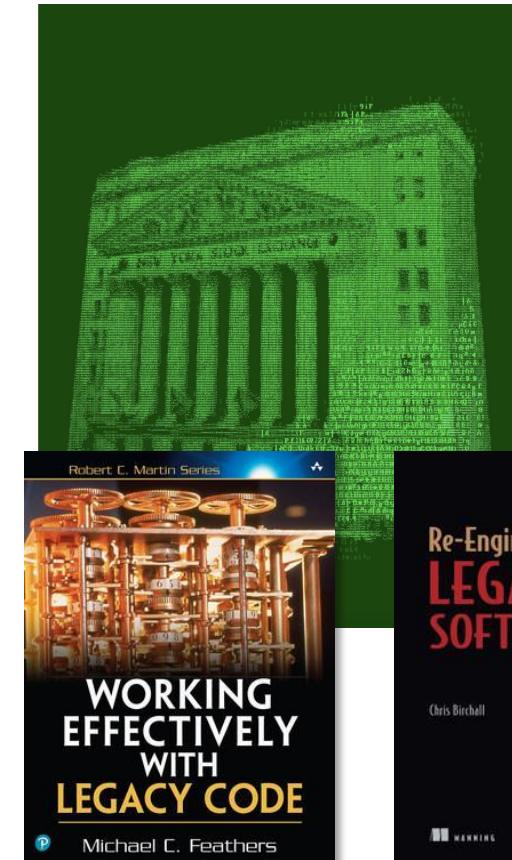
Michael C. Feathers. 2004.

- **Re-Engineering Legacy Software.**

Chris Birchall. 2016.

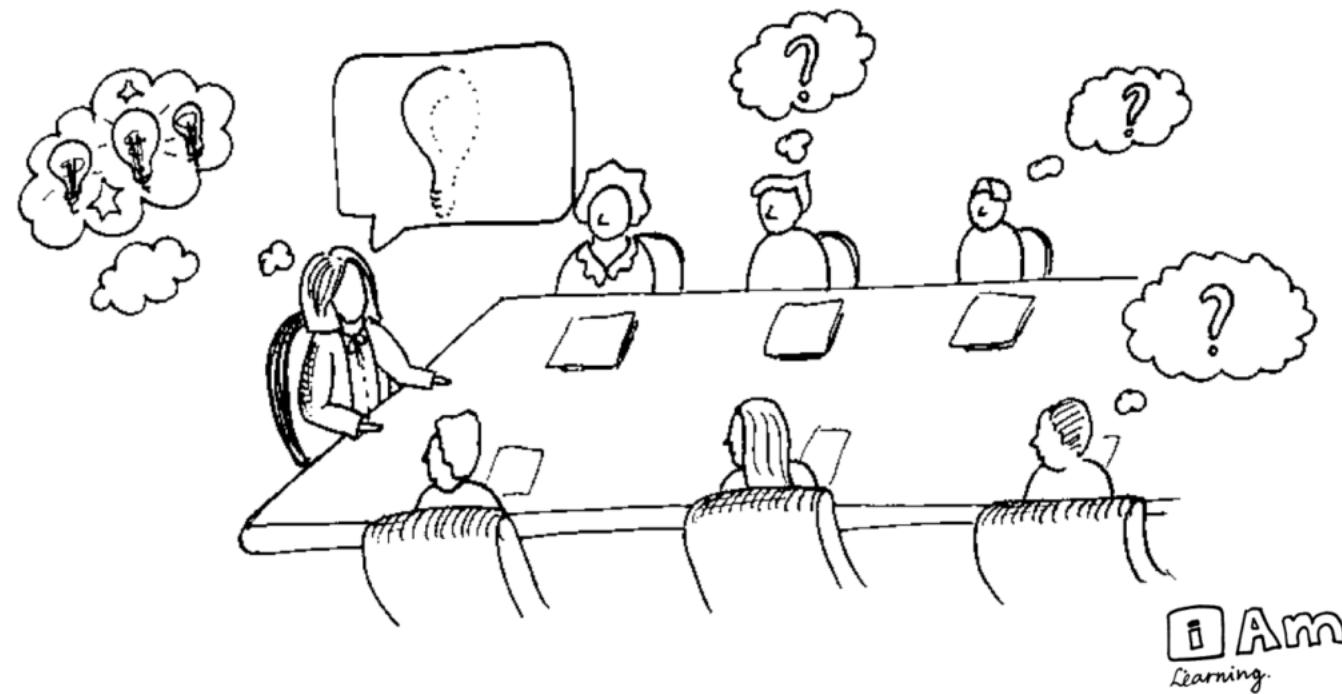
- **The Legacy Code Programmer's Toolbox.**

Jonathan Boccara. 2019.



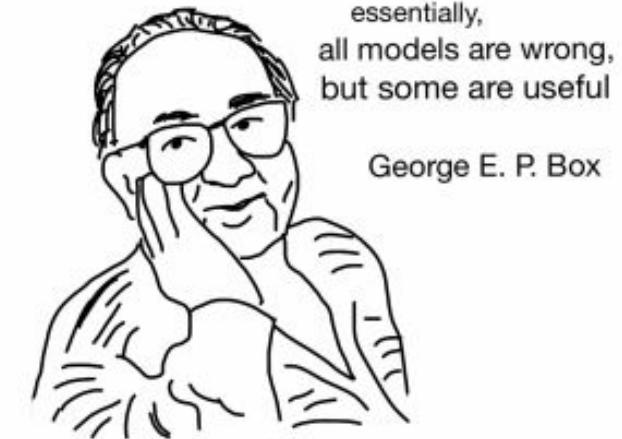
A screenshot of the Wealthsimple Magazine website. At the top right, there are links for 'START INVESTING', 'Wealthsimple Magazine', and 'MENU'. Below the header, there is a section titled 'MONEY &amp; THE WORLD' with an article thumbnail for 'The Code That Controls Your Money'. The main content area contains text about COBOL, followed by a section titled 'THE LEGACY CODE PROGRAMMER'S TOOLBOX' with a sub-section 'PRACTICAL SKILLS FOR DEVELOPERS WORKING WITH LEGACY CODE'. The page also includes a sidebar with various links and a footer with the author's name, 'JONATHAN BOCCARA', and 'FOREWORD BY KEVIN HENNEY'.

# Why? Because of Tacit Knowledge



# Today: How to tackle codebases

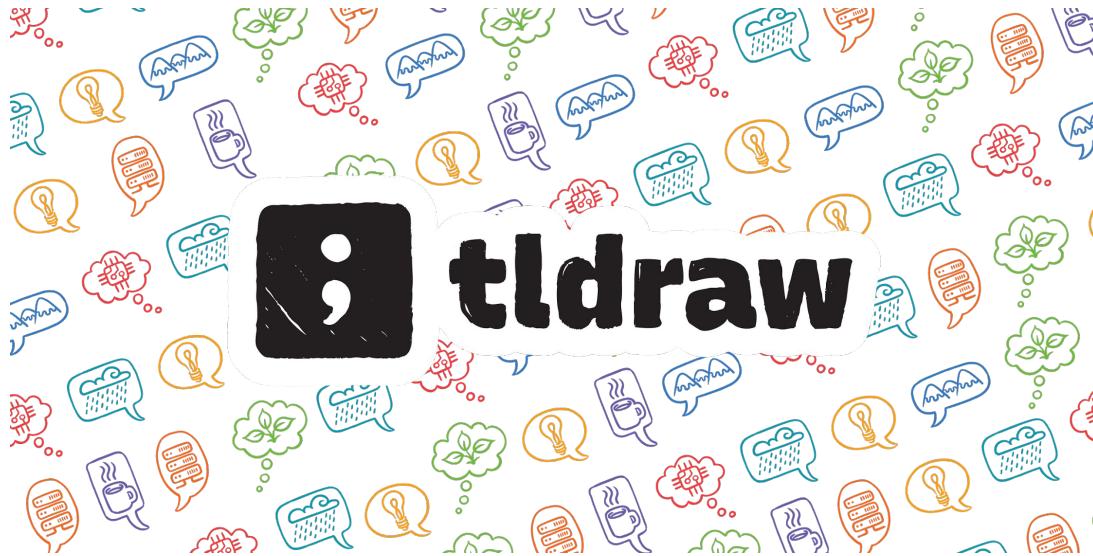
- Goal: develop and test a working model or set of working hypotheses about how (some part of) a system works
- Working model: an understanding of the pieces of the system (components), and the way they interact (connections)
- Focus: Observation, probes, and hypothesis testing
  - Helpful tools and techniques!



essentially,  
all models are wrong,  
but some are useful

George E. P. Box

# Live Demonstration: tldraw



<https://github.com/tldraw/tldraw>

# Steps to Understand a New Codebase

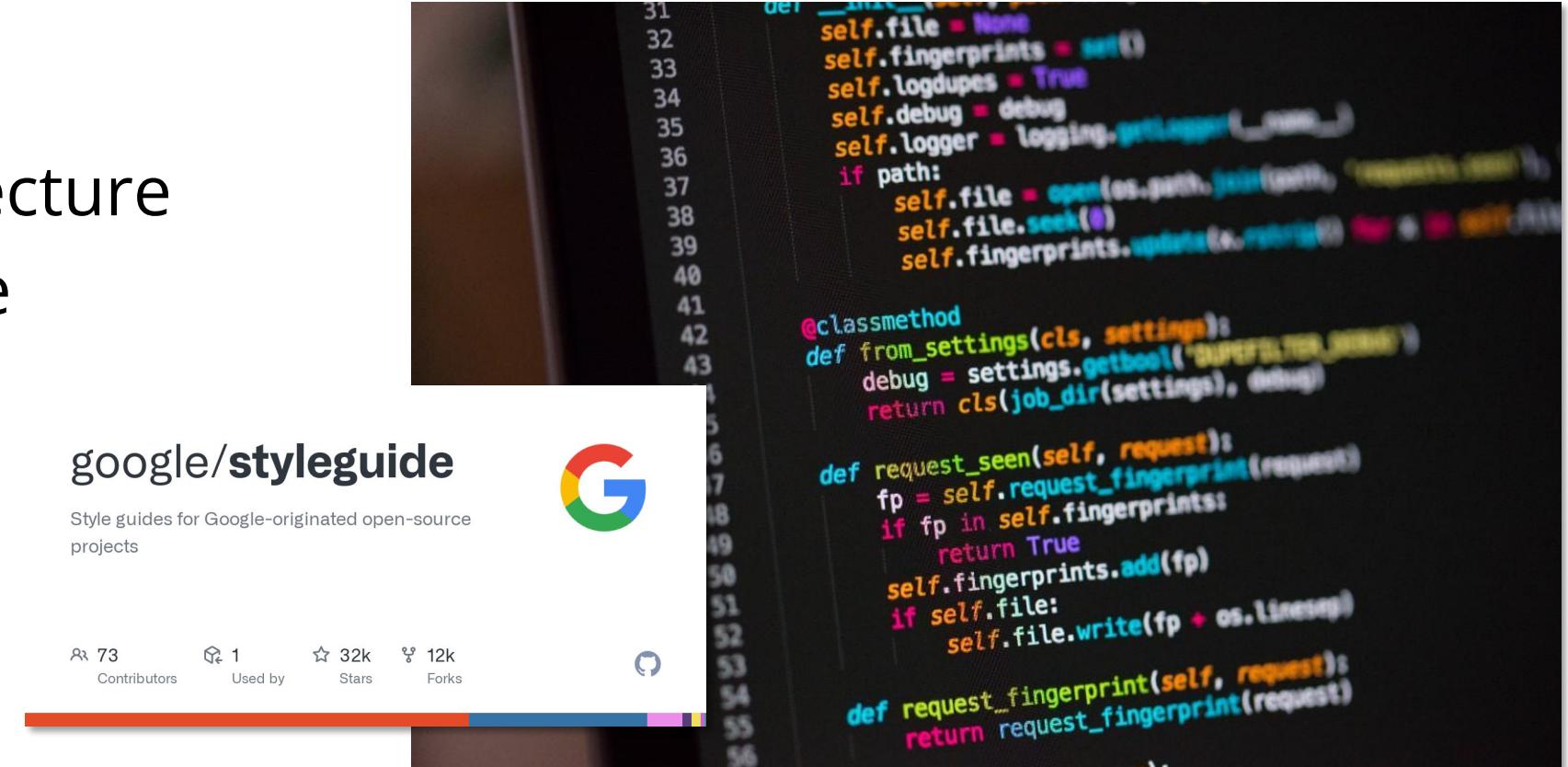
- Look at README.md
- Clone the repo.
- Build the codebase.
- Figure out how to make it run.
- What do you want to mess with?
  - Clone and own
- Traceability - Attach a debugger
  - View Source
  - Find the logs.
  - Search for constants (strings, colors, weird integers (#DEADBEEF))

# Participation Activity

- Take out a piece of paper.
- Write down one pro and one con about trying to understand a new codebase by compiling and building it vs. just reading the code.
- Pair with your neighbor and discuss your answers. Do you agree?
- Share with the class!
- Write your own andrewID on the paper, leave it at the end of class.

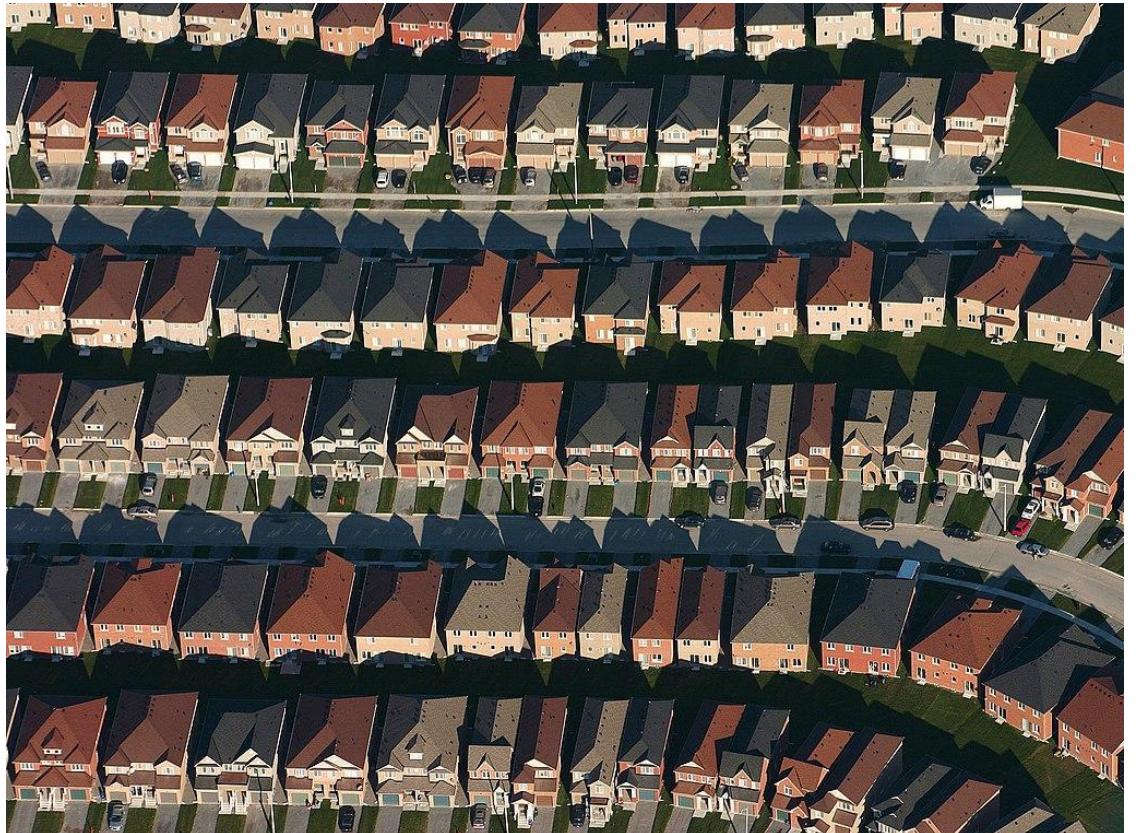
# Observation: Software is full of patterns

- File structure
- System architecture
- Code structure
- Names
- ...

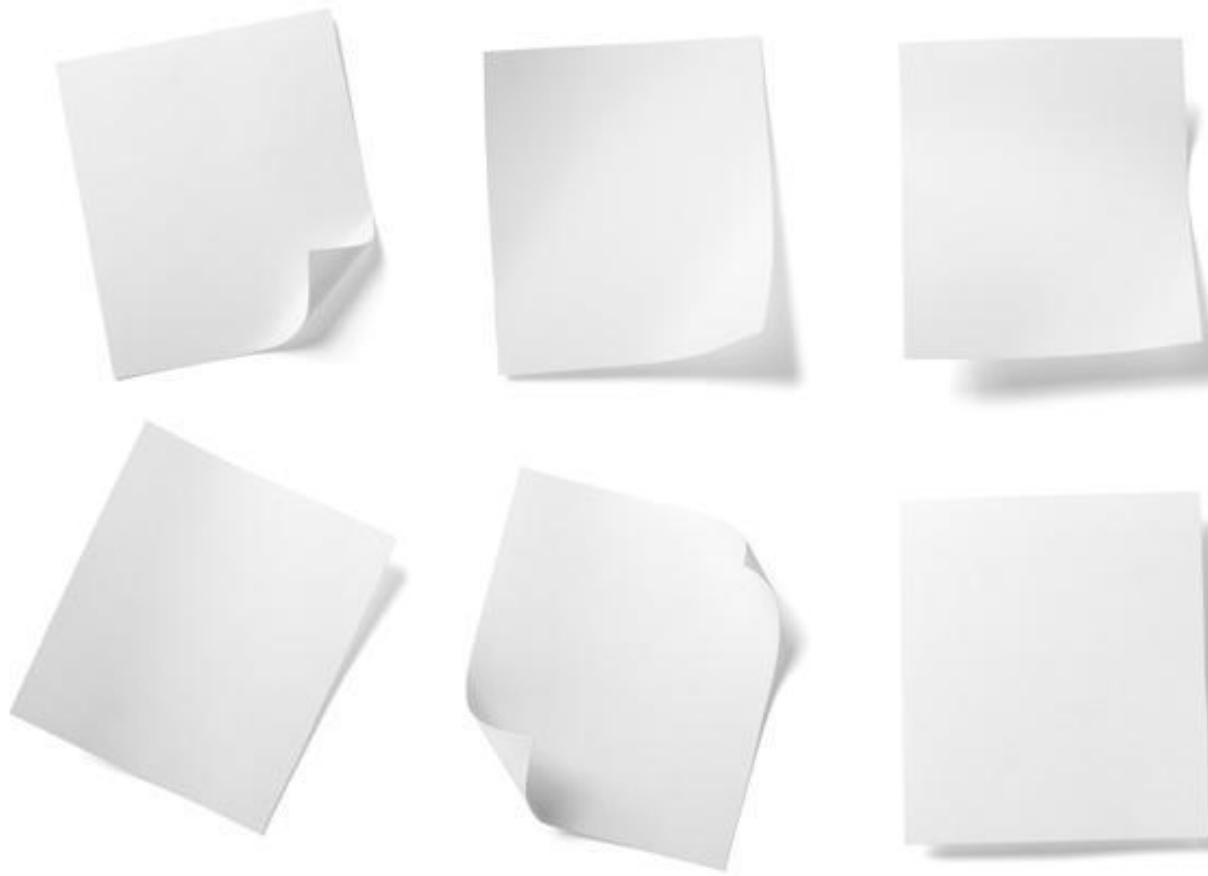


# Observation: Software is massively redundant

- There's always something to copy/use as a starting point!



# Observation: Code must run to do stuff!



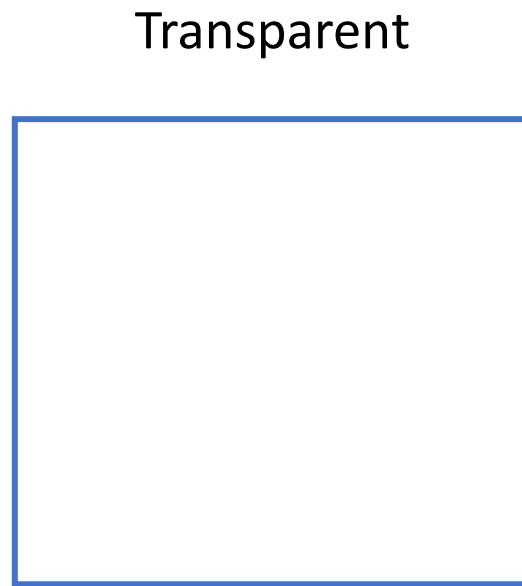
Observation: If code runs, it must have a beginning...



# The Beginning: Entry Points

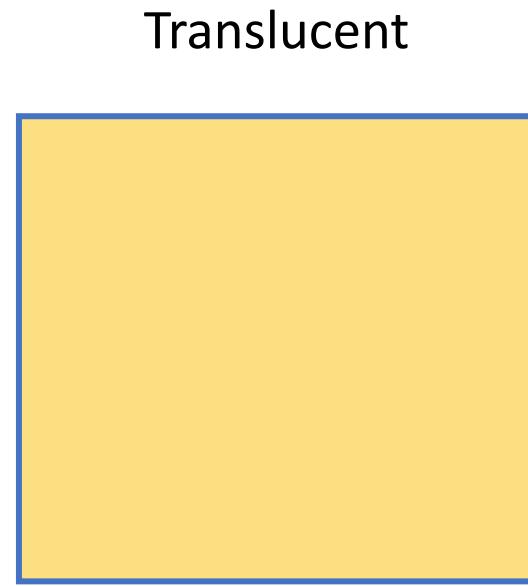
- Locally installed programs: run cmd, OS launch, I/O events, etc.
- Web apps server-side: Browser sends HTTP request (GET/POST)
- Web apps client-side: Browser runs JavaScript, event handlers

# Can running code be Probed/Understood/Edited?



Source code built locally

(P+U+E)



Binaries running locally

Open source

(P+U)

Closed source

(P)



Server-side apps running remotely

Open source

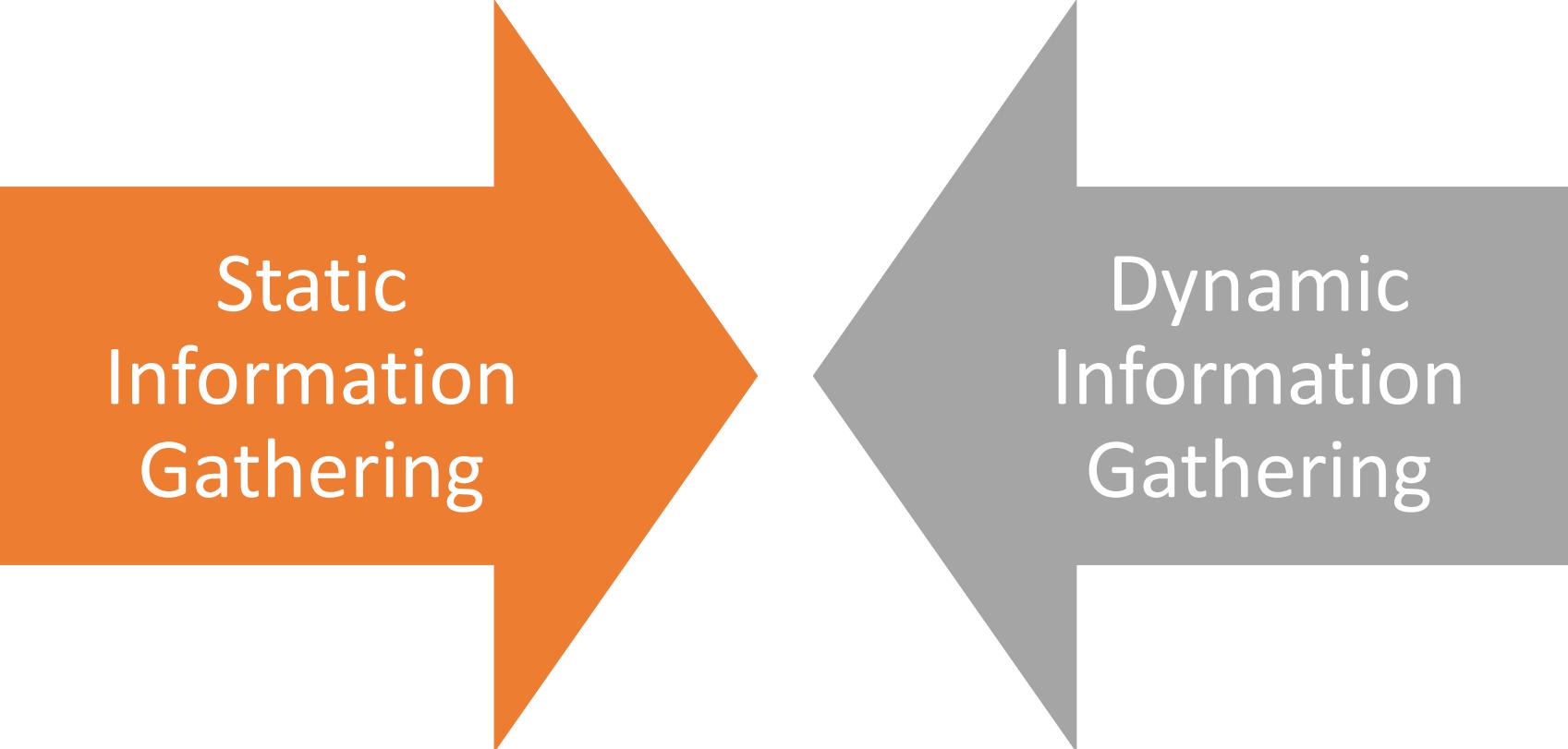
(U)

Closed source

(Talk to NSA)

# Creating a model of unfamiliar code

Source code built  
locally



Static  
Information  
Gathering

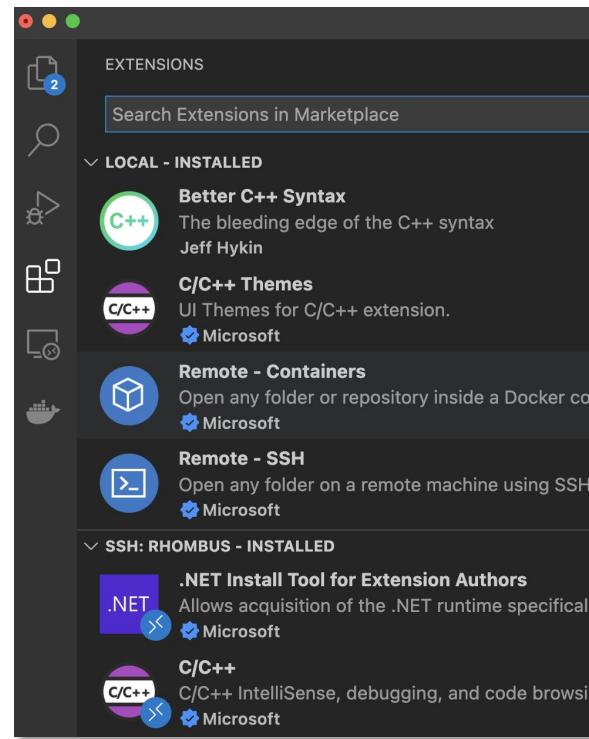
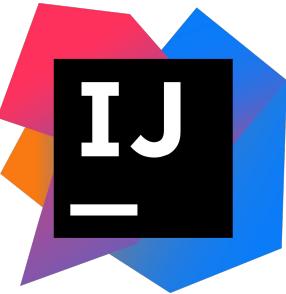
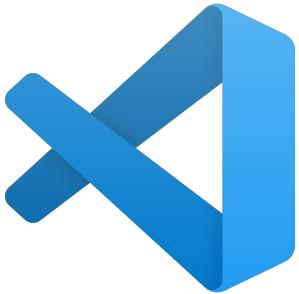
Dynamic  
Information  
Gathering

# Static Information Gathering

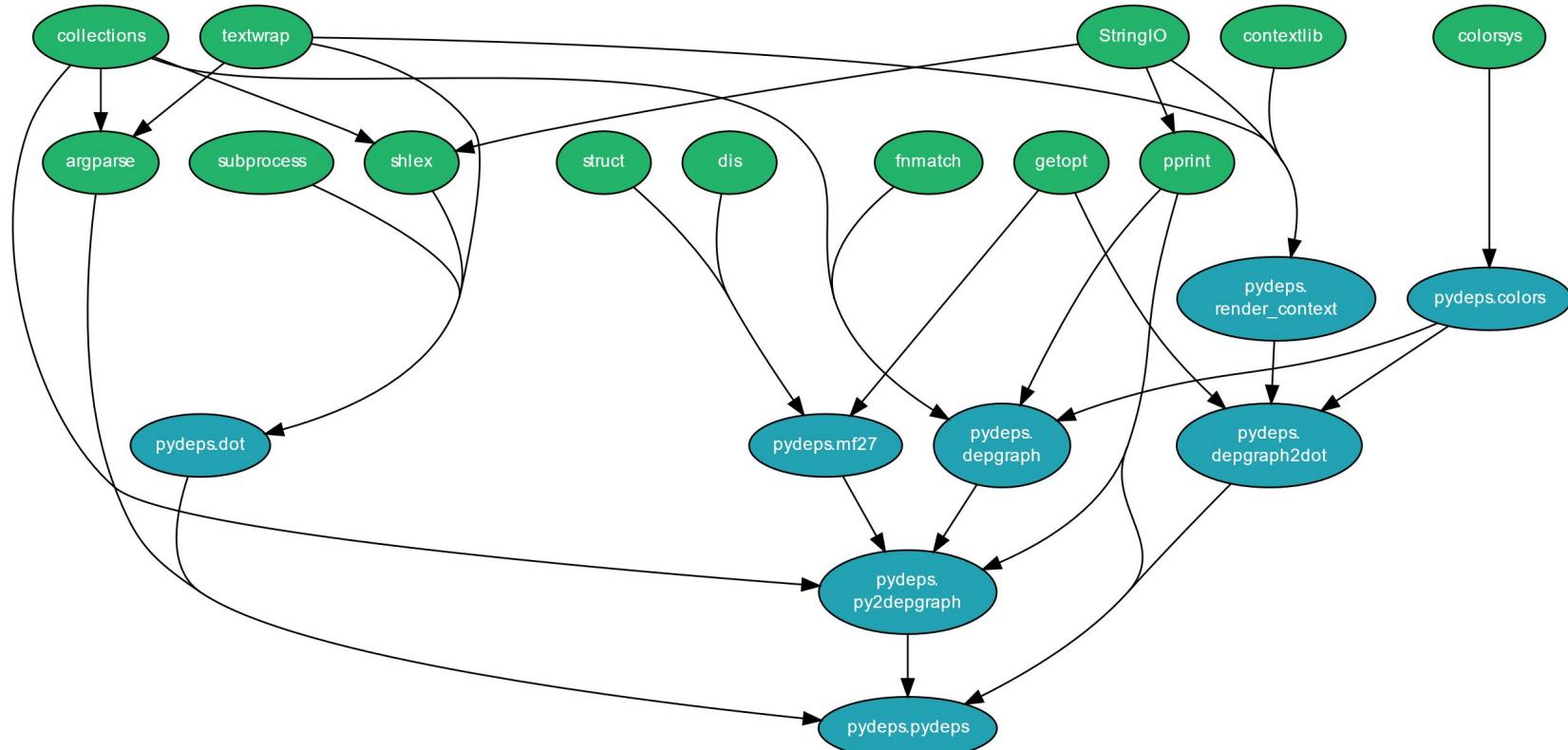
- Basic needs:
  - Code/file search and navigation
  - Code editing (probes)
  - Execution of code, tests
  - Observation of output (observation)
- Many choices here on tools! Depends on circumstance.
  - grep/find/etc. Knowing Unix tools is invaluable
  - A decent IDE
  - Debugger
  - Test frameworks + coverage reports
  - Google (or your favorite web search engine)
  - ChatGPT or LaMA

# Static Information Gathering: Use an IDE!

## Real software is too complex to keep in your head

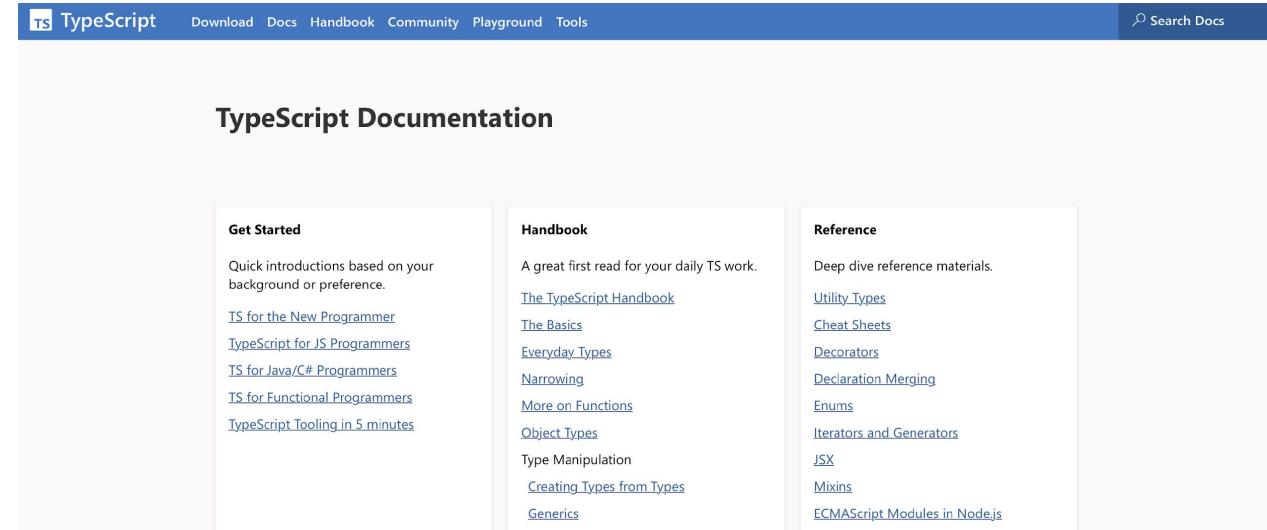
A screenshot of the Visual Studio Code interface. The top bar shows the file 'blog-post.js — gatsby-graphql-app'. The left sidebar shows the Explorer view with files like 'utils.js', 'index.js', and 'blog-post.js'. The main editor area shows code for 'blog-post.js' with syntax highlighting and code completion suggestions. The bottom status bar shows the terminal output: 'info i [wdm]: Compiled successfully.', 'info i [wdm]: Compiling...', 'info i [wdm]: DONE Compiled successfully in 63ms', and 'info i [wdm]: info i [wdm]: Compiled successfully.'.

# Dependency maps



# Consider documentation and tutorials judiciously

- Great for discovering entry points!
- Can teach you about general structure, architecture (more on this later in the semester)
- Often out of date.
- As you gain experience, you will recognize more of these, and you will immediately know something about how the program works
- Also: discussion boards; issue trackers



The screenshot shows the official TypeScript Documentation website. At the top, there is a navigation bar with links for "Download", "Docs", "Handbook", "Community", "Playground", and "Tools". On the far right of the navigation bar is a search bar labeled "Search Docs". Below the navigation bar, the title "TypeScript Documentation" is centered. The page is divided into three main sections: "Get Started", "Handbook", and "Reference". The "Get Started" section contains links to "TS for the New Programmer", "TypeScript for JS Programmers", "TS for Java/C# Programmers", "TS for Functional Programmers", and "TypeScript Tooling in 5 minutes". The "Handbook" section contains links to "The TypeScript Handbook", "The Basics", "Everyday Types", "Narrowing", "More on Functions", "Object Types", "Type Manipulation", "Creating Types from Types", and "Generics". The "Reference" section contains links to "Utility Types", "Cheat Sheets", "Decorators", "Declaration Merging", "Enums", "Iterators and Generators", "JSX", "Mixins", and "ECMAScript Modules in Node.js".

# Discussion Boards and Issue Trackers

The screenshot shows the Stack Overflow search results for the query "java on mac". The results page has a header with navigation links like "About", "Products", "For Teams", a search bar, and "Log in / Sign up". Below the header, there's a sidebar with links for "Home", "PUBLIC", "Questions", "Tags", "Users", "Companies", "COLLECTIVES", "Explore Collectives", and "TEAMS". A callout for "Stack Overflow for Teams" is present. The main content area displays 500 results, with the first few questions listed:

- How to set or change the default Java (JDK) version on mac OS?** (1311 votes, 36 answers, 1.4m views) - How can you change the default version of Java on a mac? ...
- How to install Java 8 on Mac** (1271 votes, 34 answers, 1.3m views) - I want to do some programming with the latest JavaFX, which requires Java 8. I'm using IntelliJ 13 CE and Mac OS X 9 Mavericks. ... But /usr/libexec/java\_home doesn't...
- Where is Java Installed on Mac OS X?** (861 votes, 20 answers, 1.0m views) - I downloaded Java 7u17 on Mac OS 10.7.5 from here and successfully installed it. To do some JNI programming, I need to know where Java is installed on my Mac. ....
- How do I install Java on Mac OSX allowing version switching?** (479 votes, 11 answers) - I want to install OpenJDK Java on Mac OSX and have it work alongside other JDK's since it is a newer release. ... It looks like only the current version too: brew cask inf...

At the bottom of the page, there's a link to the original URL: "Open "https://stackoverflow.com/questions/21964709/how-to-set-or-change-the-default-java-jdk-version-on-macos?r=SearchResults"" in a new tab.

The screenshot shows the Sismics / reader issue tracker interface. The top navigation bar includes "Code", "Issues 30", "Pull requests", "Actions", "Projects", "Wiki", "Security", and "Insights". The "Issues" tab is selected. The search bar contains "is:issue is:open". The main list of issues includes:

- 30 Open, 131 Closed
- Rss feed
- Docker and database docker name
- Error on OPML import
- feature request: naive baynes ham / spam classifier
- file is broken msg on mac
- default credentials don't work
- Detect duplicate article in different feed
- Android : Dark mode

Each issue entry shows the title, a brief description, and the number of comments (e.g., "4", "1").

# Dynamic Information Gathering

Change helps to inform and refine mental models

- Build it.
- Run it.
- Change it.
- Run it again.
- How did the behavior change?



# How to start?

- Confirm that you can build and run the code.
  - Ideally both using the tests provided, and by hand.
- **Confirm that the code you are running is the code you built!**
- Confirm that you can make an externally visible change
- How? Where? Starting points:
  - Run an existing test, change it
  - Write a new test
  - Change the code, write or rerun a test that should notice the change
- Ask someone for help

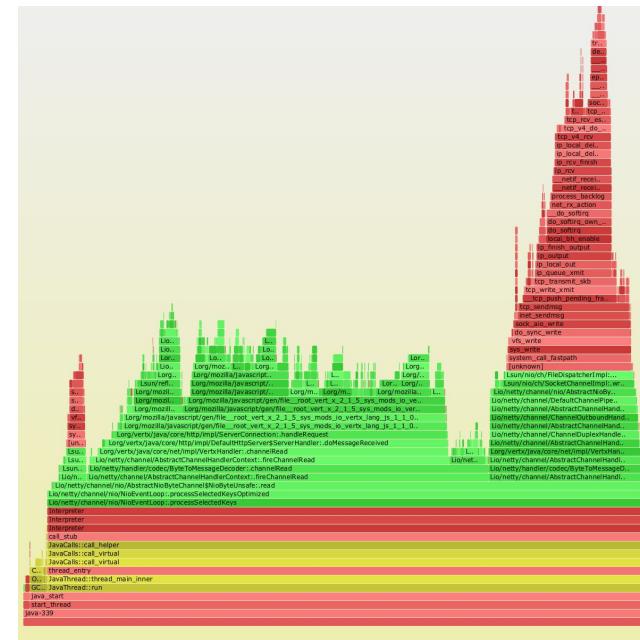
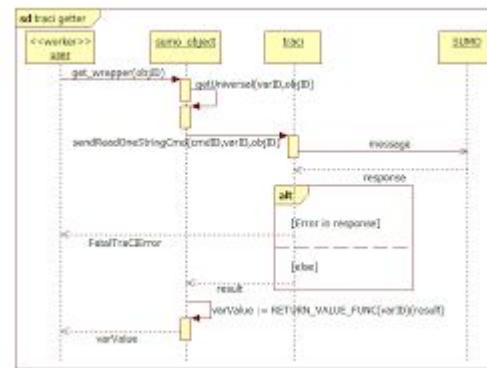
# Probes: Observe, control or “lightly” manipulate execution

- `print("this code is running!")`
- Structured logging
- Debuggers
  - Breakpoint, eval, step through / step over
  - (Some tools even support remote debugging)
- Delete debugging
- Chrome Developer Tools

The screenshot shows the Visual Studio Code interface during a debug session. The code editor displays `Owner.java` from the `spring-petclinic` project. A breakpoint is set on the line `this.telephone = telephone;`. The debugger sidebar shows the `VARIABLES` tab with a context menu open over the `telephone` variable, with the option `Break When Value Changes` highlighted by a yellow arrow. The `CALL STACK` shows multiple threads, with one thread labeled `PAUSED ON DATA BREAKPOINT`. The `BREAKPOINTS` section at the bottom shows a checked checkbox for `Owner.telephone : String`, also highlighted by a yellow arrow.

# Runtime code analysis tools

- Collect runtime traces and visualize them
  - Flame graphs
  - Sequence diagrams
- Use judiciously



# Tip: Find a particular thing and trace the action backward

NodeBB

Home / Categories

CATEGORIES

- Announcements**  
Announcements regarding our community
- General Discussion**  
A place to talk about whatever you want
- Comments & Feedback**  
Got a question? Ask away!
- Blogs**  
Blog posts from individual members

0 TOPICS    0 POSTS    No new posts.

1 TOPICS    2 POSTS    A about 16 hours ago @admin Hello

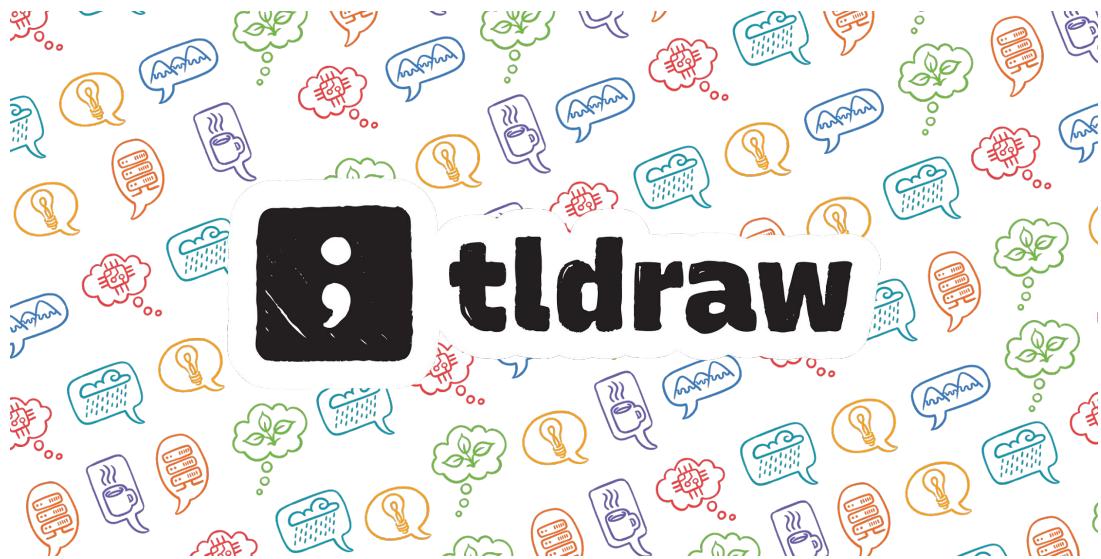
0 TOPICS    0 POSTS    No new posts.

0 TOPICS    0 POSTS    No new posts.

Powered by NodeBB | Contributors

E.g.,  
Where do categories come from?  
How are they stored?  
How are they rendered?

# Let's try some of these techniques again...



<https://github.com/tldraw/tldraw>

# Remember...

- Reading and understanding code is one of the most important skills you should learn
- It's common to get stuck or feel overwhelmed. **Don't give up!**
- Consider yourself lucky! Things are much easier today



# Learning Goals

- Understand and scope the task of taking on and understanding a new and complex piece of existing software
- Appreciate the importance of configuring an effective IDE
- Contrast different types of code execution environments including local, remote, application, and libraries
- Enumerate both static and dynamic strategies for understanding and modifying a new codebase