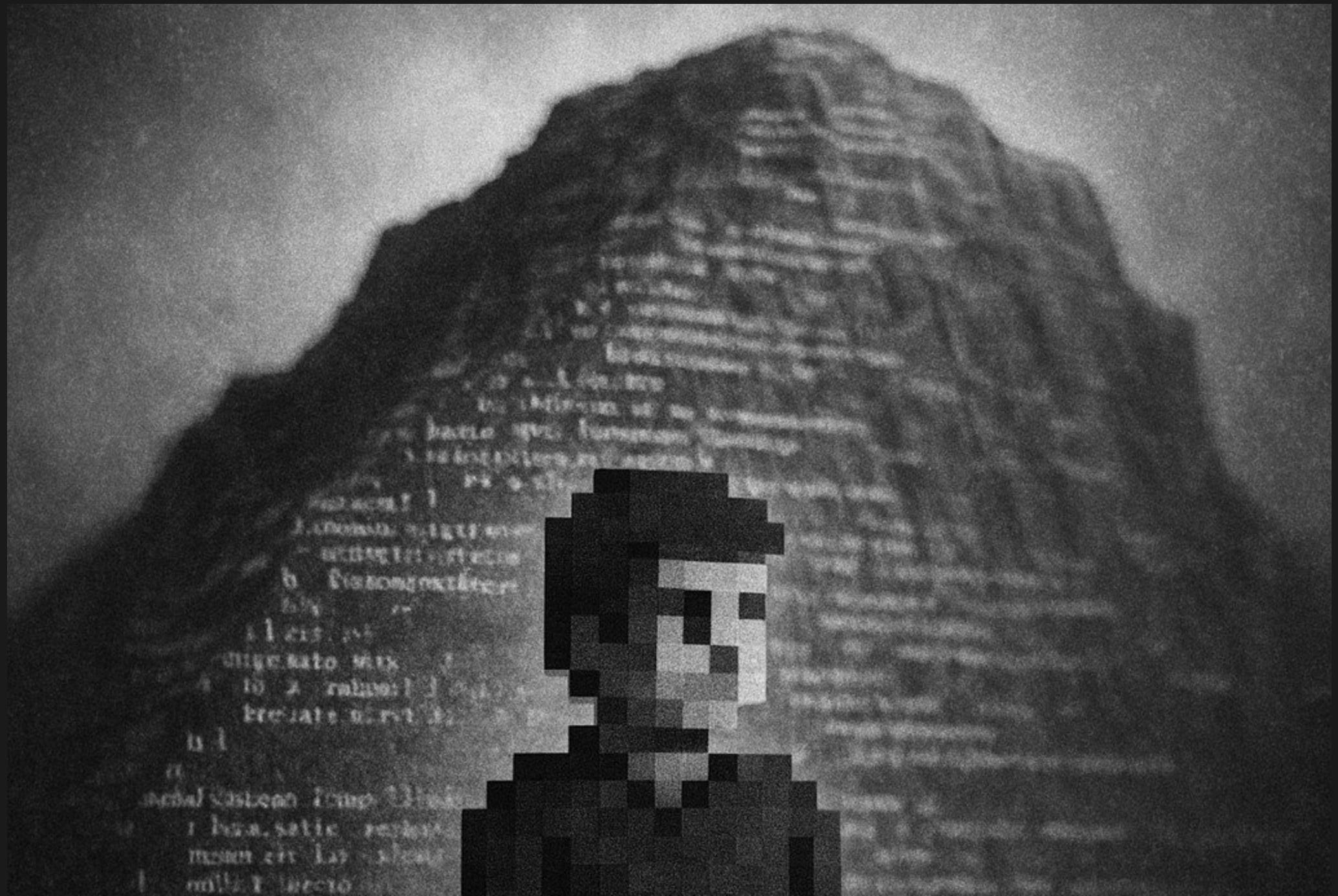
Code Reading Workshop

2025-06-03



Reading Code is Scary

- Provide concrete methods that are better than
  - (Aimlessly) browsing in the IDE
  - Debugging
  - Reading (outdated) Documentation
  - Waiting on an Epiphany

Introduction

- We spend more time **reading** code than **writing** code
- We don't practice reading code much
- What if we practice reading code explicitly?

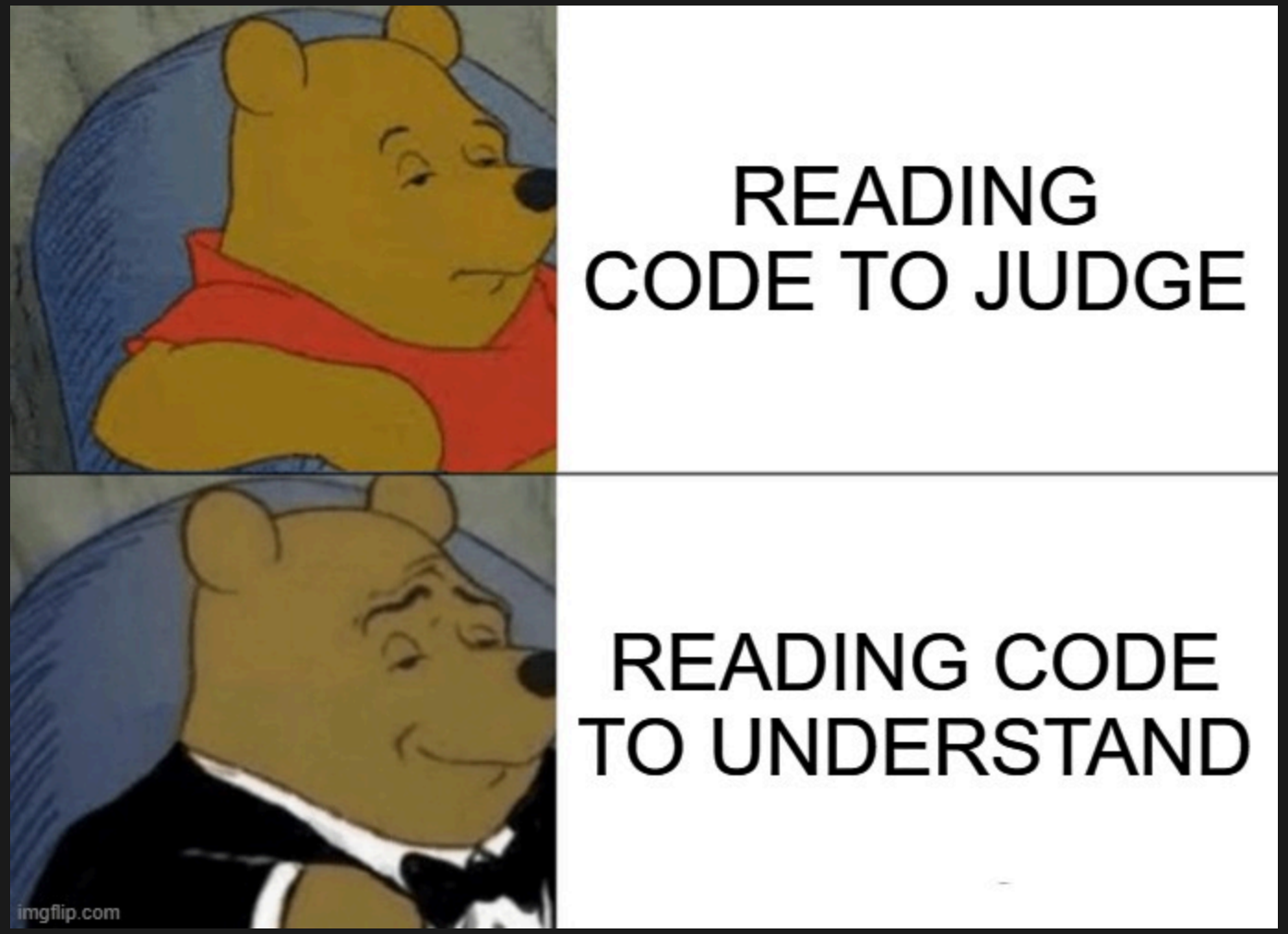
Further References

- Code Reading Club
- Code Reading Club – Felienne Hermans
- GitHub - CodeReadingClubs/Resources
- Benefits of joining a Code Reading Club - Marit van Dijk

Expectations

> A Code Reading Club is half Book Club, half Escape Room

Expectations II



Expectations III

- Unfamiliar Code
- Consecutive, concrete and fast exercises
  - From "First Impression"
  - To "Getting a good understanding"

Expectations IV

- No correct answers
- Learn how you and others reason about code
- There are different instincts and strategies, all are valid

Hands-on Workshop

Exercise 1: First Glance

- Practice to get a quick first impression of the code
- It doesn't matter how trivial your findings are
- This is about the immediate, intuitive reaction

Glance at the Code (1 min)

- Note down the **first**, **second** and **third** thing that catches your eye
- Use the reminder of the minute to think about **why** you picked those

Discussion

- What did you pick and why?
- Do those observations help with deciding what to look for next?
- Are there lines or statements that were chosen by everyone vs by only a few?

Technical Discussion

- What is the domain, framework and programming language?
- What knowledge do you think is required to better understand the code?

Sidetrack: IDE Support

> Code is not read linearly
>
> Good IDEs support via

- Syntax highlighting
- Search
- Find Usages
- Peek (go to definition)
- Collapse/Expand
- ...

Dependency Graph

> Bring order to the chaos

- Manually create a dependency graph
- Visualize the structure

Examine the Code (5 min)

- Square all the class definitions
- Circle all the function definitions
- Underline all the variables

Draw a link between the definitions and their usage(s)

Discussion

- Where there any problems?
- Are there any patterns visible?
- How does the data flow through the code?
- What parts of the code seem to warrant more attention?

Discussion

- What is the structure of this code?
- What about the nesting level?
- Are whitespaces used to structure the code?
- What other *anchors* can you use?

Understanding the Domain

> Glance at the bigger picture

> The limits of my language are the limits of my world

- Discover the main concepts
- Learn the vocabulary of the domain

Find Concepts (5 min)

- Note down the three most important concepts
- Could be names, classes, functions, variables, algorithms, data structures, assumptions or techniques

Discussion

- What concepts did you pick?
- Topics covered by many vs by few
- What strategies were used to decide?

Most Important Lines (3 min)

- Find the three most important lines

Discussion

- What lines are covered by many people?
- Discuss why particular lines were chosen
- Can we agree as a team?

Summary (3 min)

- Try to write down the concise essence of the code in no more than two short sentences.

Discuss

- Can the team agree on a combined summary?
- What would you ask the original developer?

Retrospective

- Take a few moments to reflect on the session
- What methods felt good, what felt awkward?
- Were there any impediments?
- What did you learn?
- How can you apply what you have learned in your daily work?