

How to write good code?

Assignments tips for FoA

Assignment 1

- Release late this week or next week
 - When you get Assignment 1
-
- START NOW

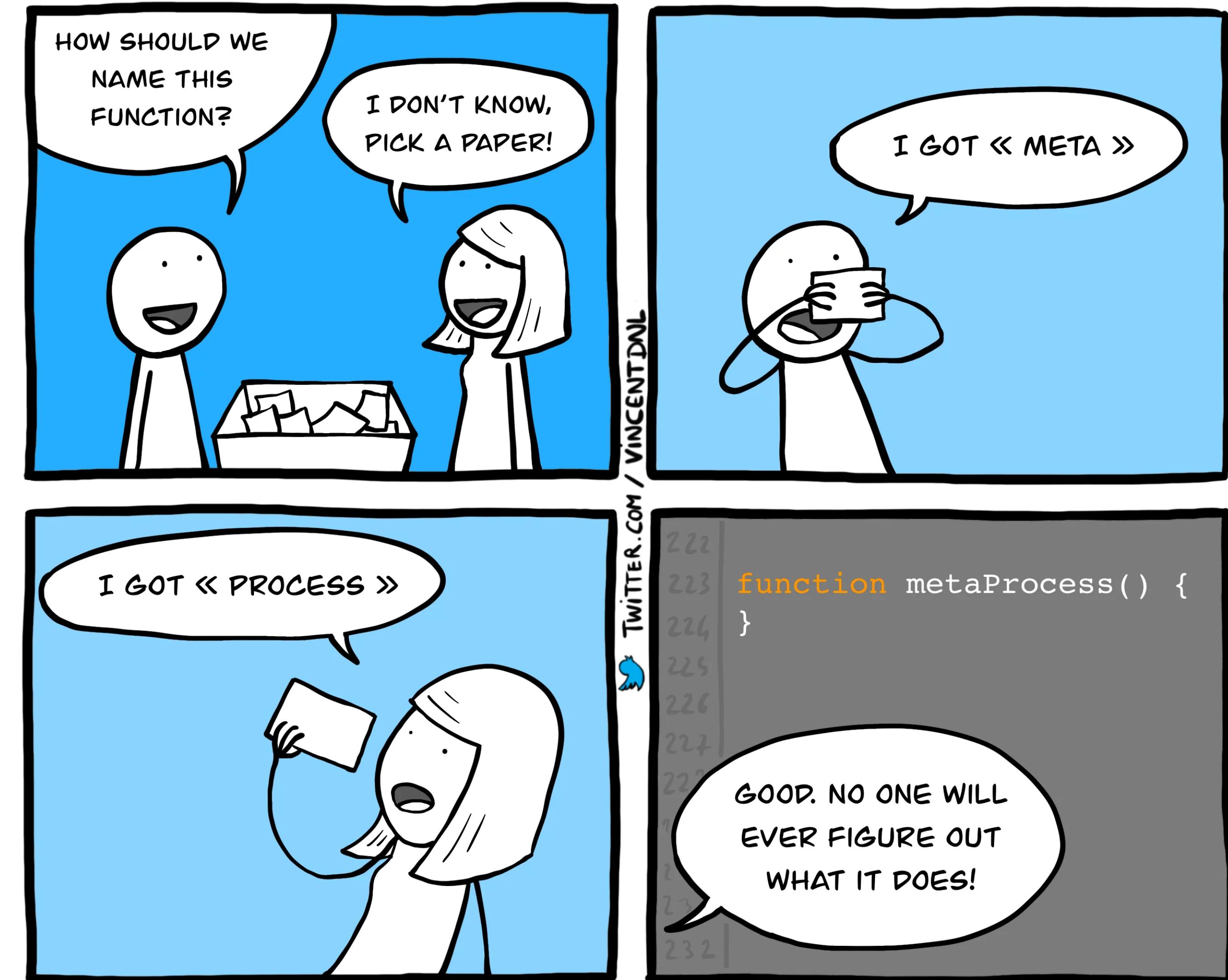
1. Read the assignment spec carefully.
2. Read the scaffolding code carefully.
3. Read the FAQ on Ed carefully, and bookmark it.
4. Read the marking rubric carefully.
5. Fill your name and student ID on the authorship statement.
6. Start working on the scaffold.

- Submit early and often. You can submit as many times as you want. Submission process is tricky the first time.
- Don't overcomplicate things. If something's given from the slides, you can use it as long as you attribute it.
- Use CCheck, it's great, if you believe it's not, you are wrong
 - Understand the output
 - **Fix your code accordingly**
 - We mark your code with it

- DRY: Don't repeat yourself. Use functions.
- Don't have all of your code in the `stage` functions.
- Watch for code repetition.

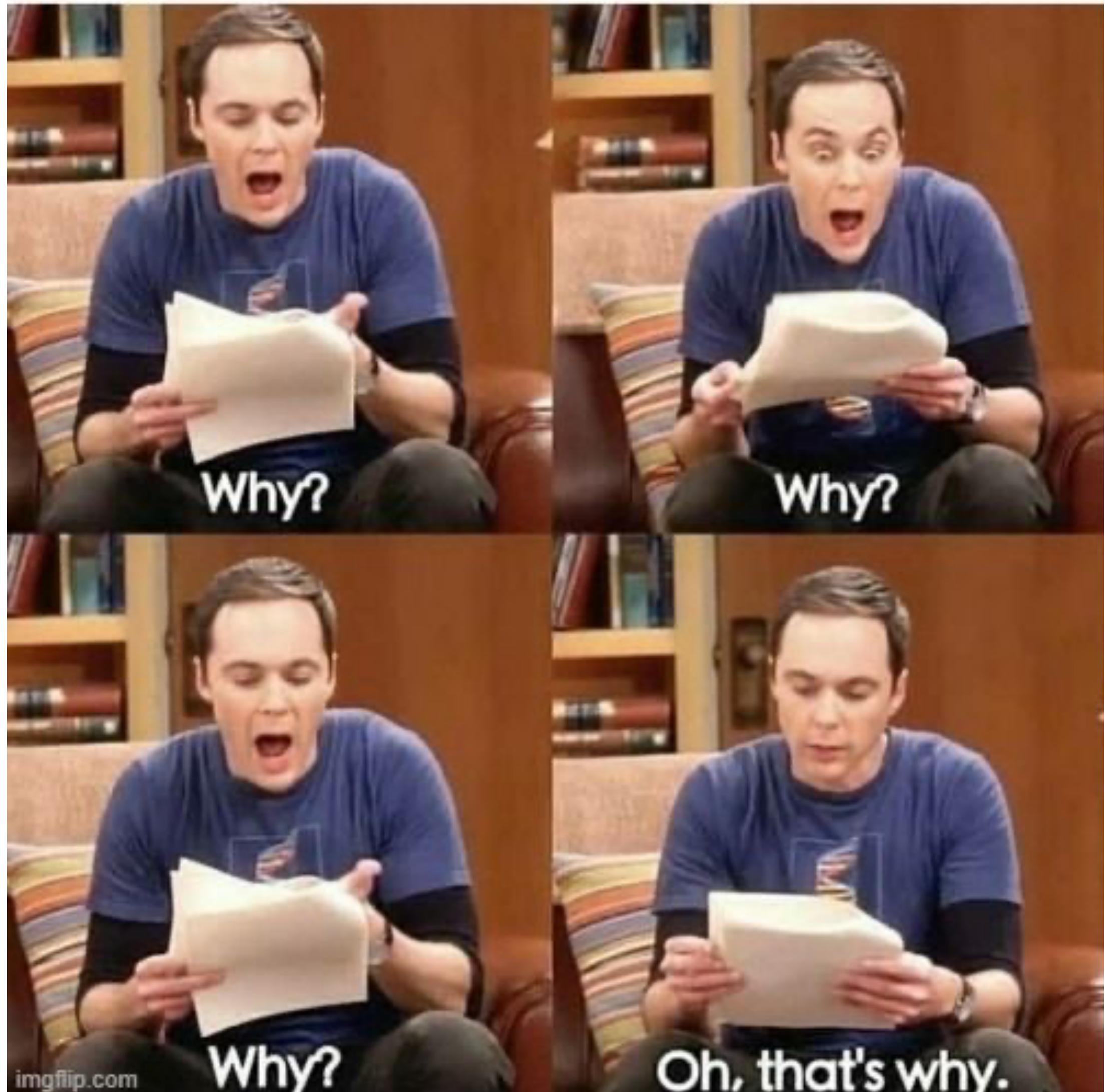
- From the spec, marks deducted for:
 - Functions too long or too complex;
 - Insufficient use of functions;
 - Unnecessary duplication/copying of data;
- A rule of thumb: all functions should be less than 50 lines long
 - unless *absolutely* necessary

- Don't forget function declarations
- Don't forget doc-strings
- Give your reasonable names
- Don't use nested functions

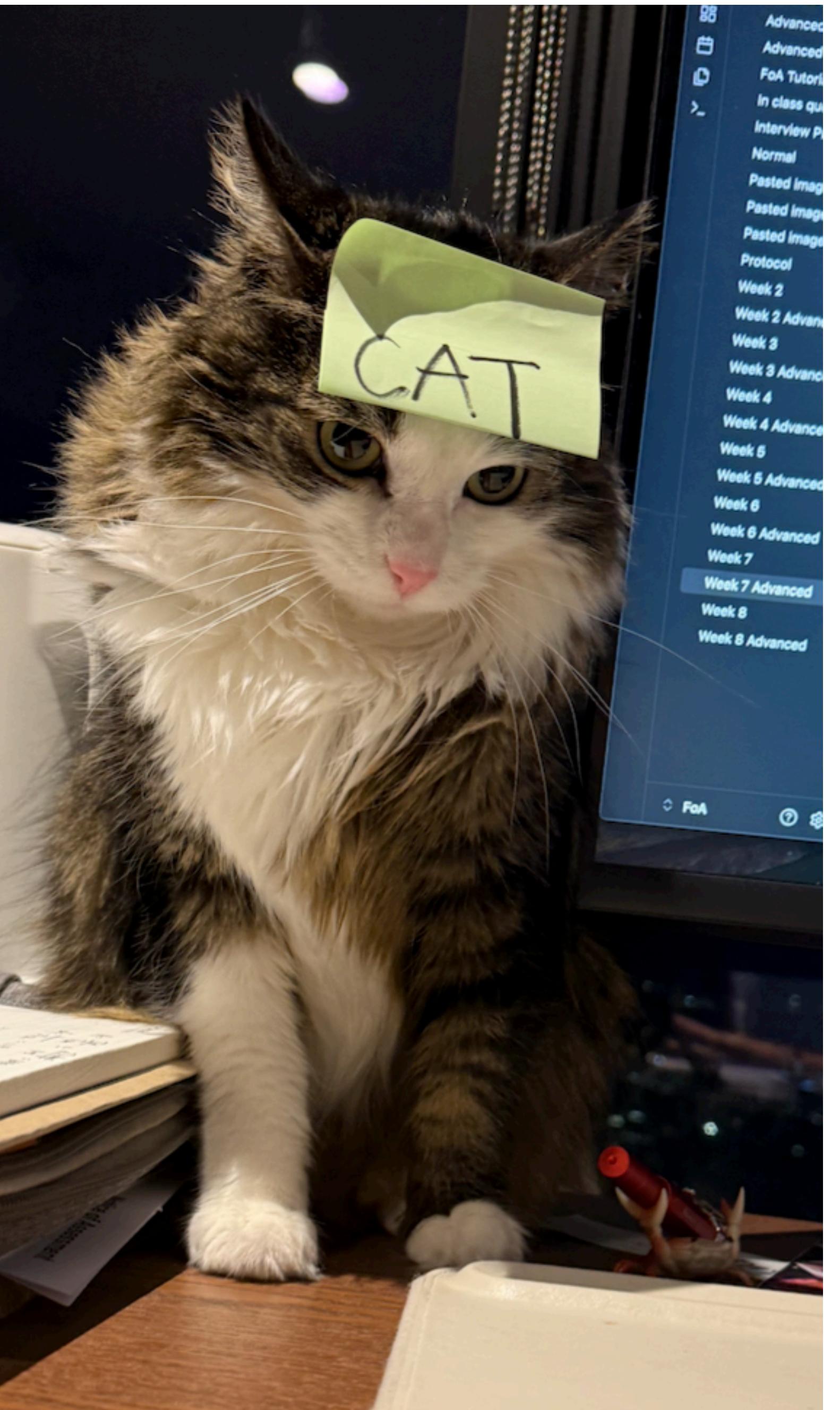


WHEN YOU LOOK AT CODE YOU WROTE LAST YEAR

- Comments when necessary
 - Doc-strings
 - Big code block
 - Tricky logic that needed explanation

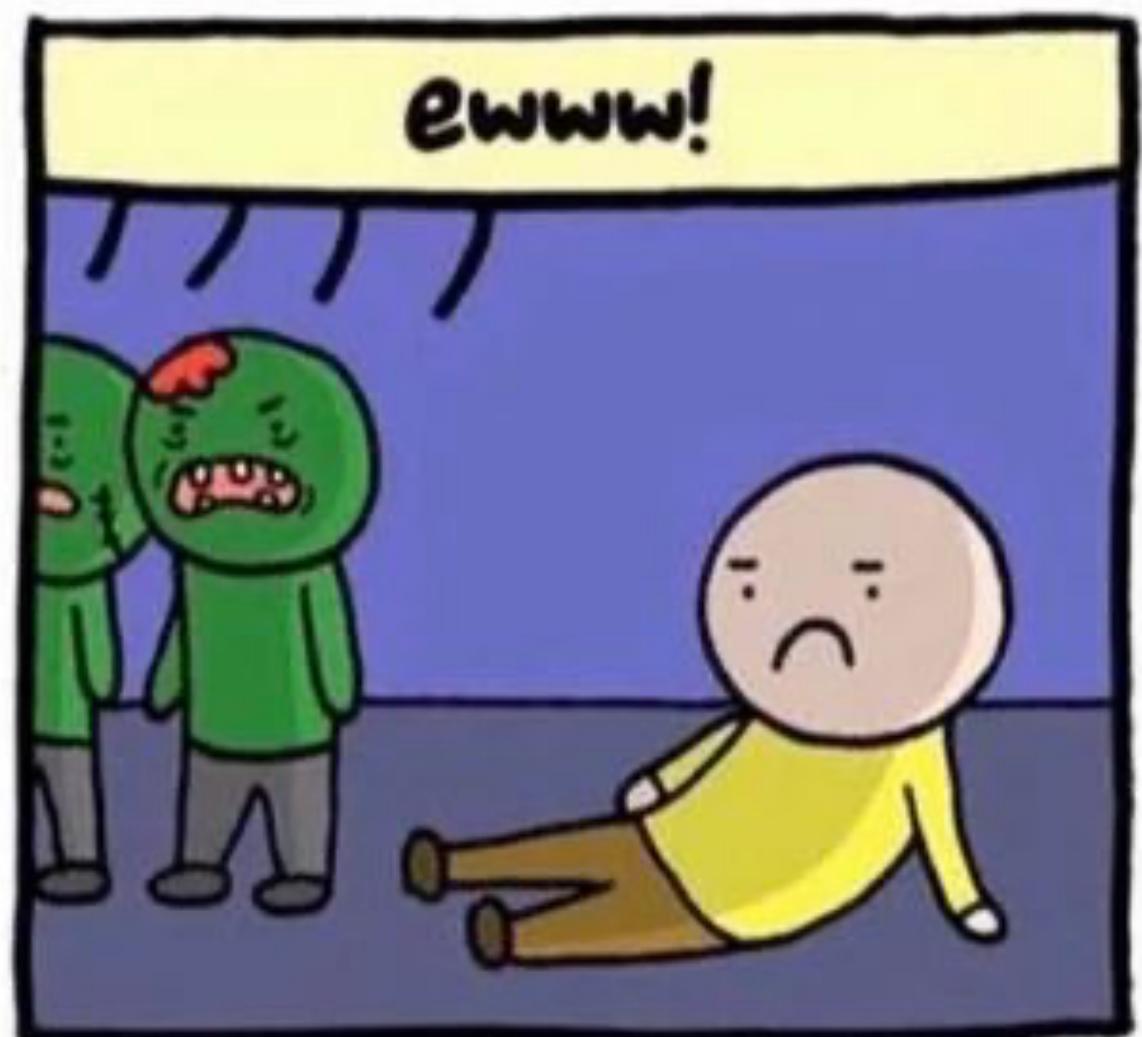
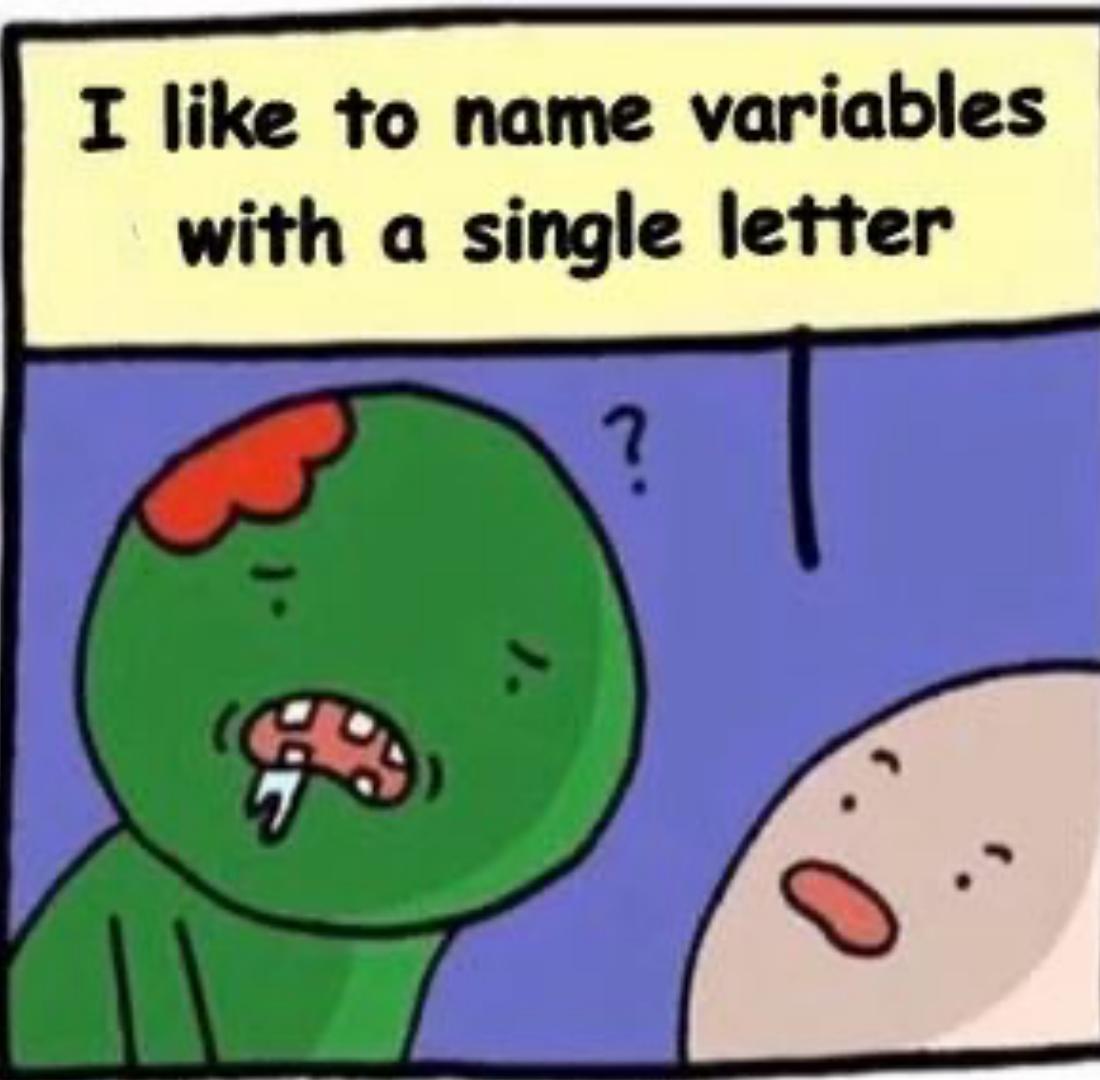
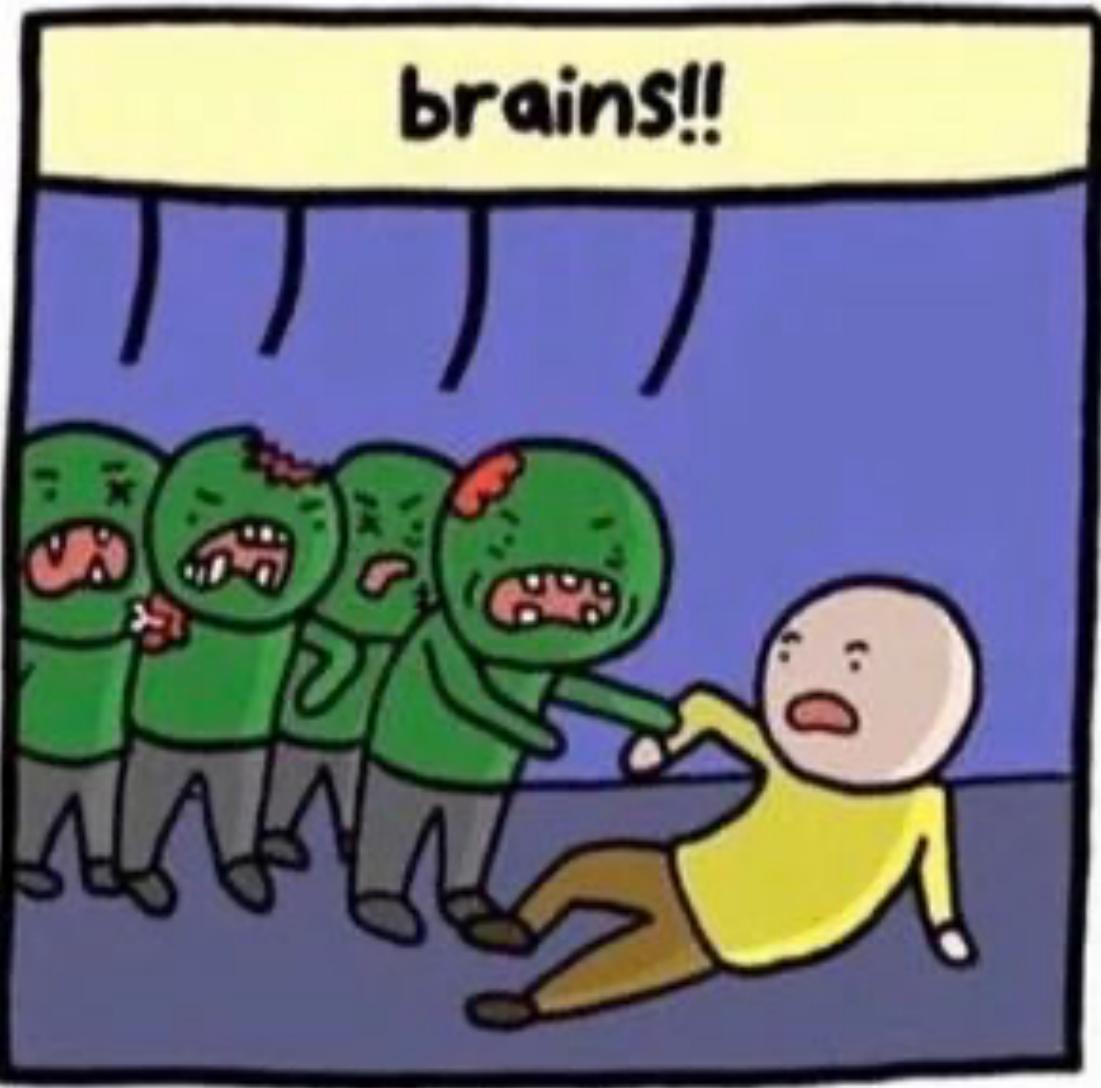
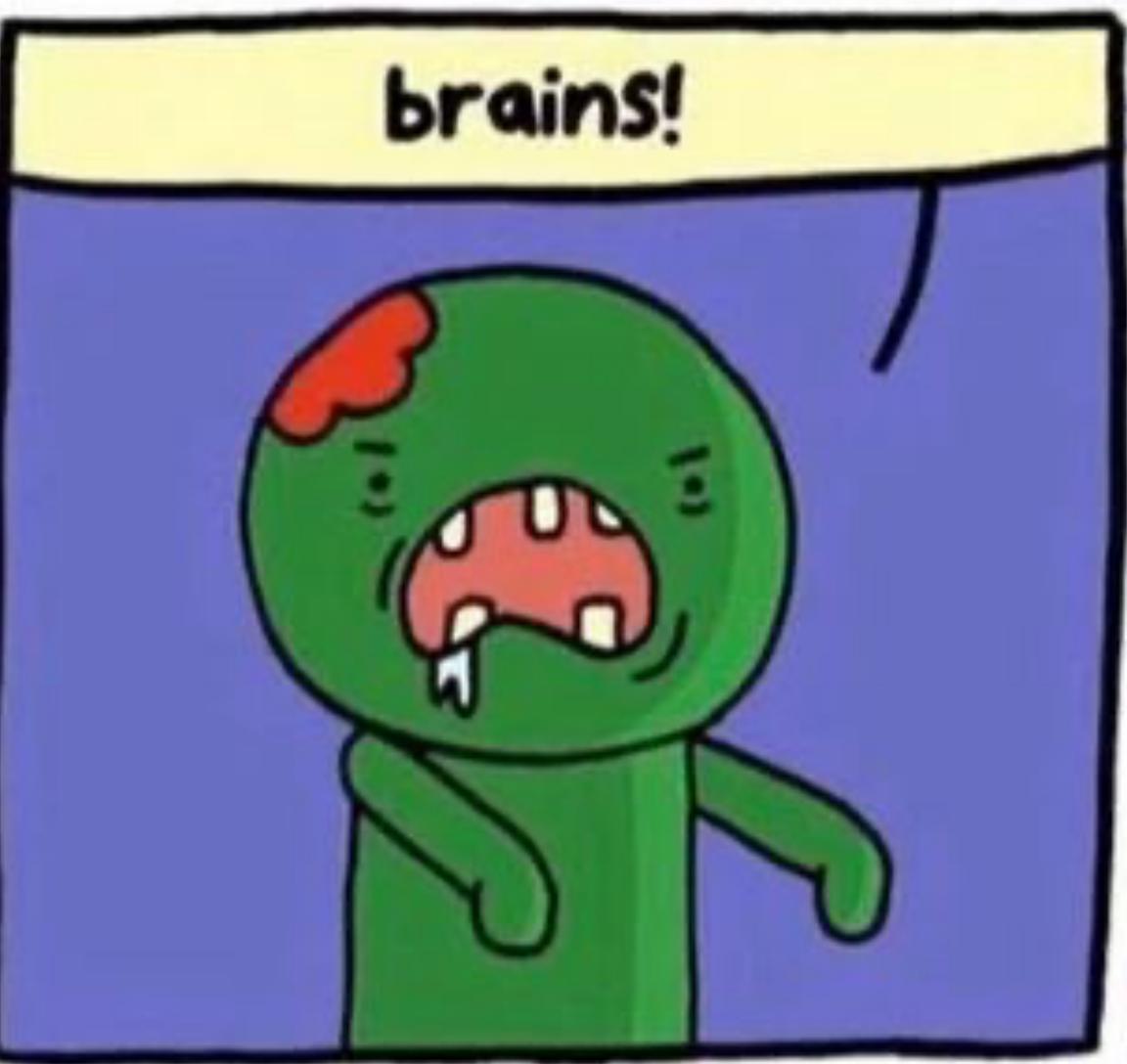


90% of code comments be like:



- Comments when ***necessary***

- Give your variable, constants reasonable names
- Please



- `#define` your 'magic numbers' and constants
- You can delete the /* add code */ lines.
- Lines need to be <= 100 chars. Use the ruler.
- Watch your indentation, braces and style. Tab and spaces are different
- Examples of good style:
 - slides (lectures, mine (most of the time))
 - workshop solutions
 - the textbook

- Rubric/spect
 - Program execution – correctness
 - Program presentation – var names, line length, etc.
 - Structure and Approach – use of functions etc.
- You get make for attempting each stage
 - There will be partial marks so please attempt them all
 - Don't give up

When there's a bug
in your code:

- Debugging can be tricky
- Print things out
- Use a debugger



- Late Penalty
- AAP/Special considerations: Email subject coordinator
- Use Ed(private post if it contains your code) / FYC / me as resources to help you with assignments
- DO NOT
 - Use LLM (we can tell)
 - Send your friends your code or ask for their code
 - Use code given by tutoring companies (we can also tell)