How to …
Write documentation

DOCUMENTATION, README AND CODE COMMENTS

Why documentation is important

- Making sure that you have a record for later reference, don't assume you'll remember details
- Sharing information with others on your team or at your company
- Having a reference for things that need to happen repeatedly, infrequently, or be done by lots of people etc.
- Having a historical record so that you can see why something is the way it is without having to track down the original creator
- Having a place for information that doesn't require time from others to explain things to you
- Keeping a record so there aren't conflicting points of view, such as issues that get escalated to HR

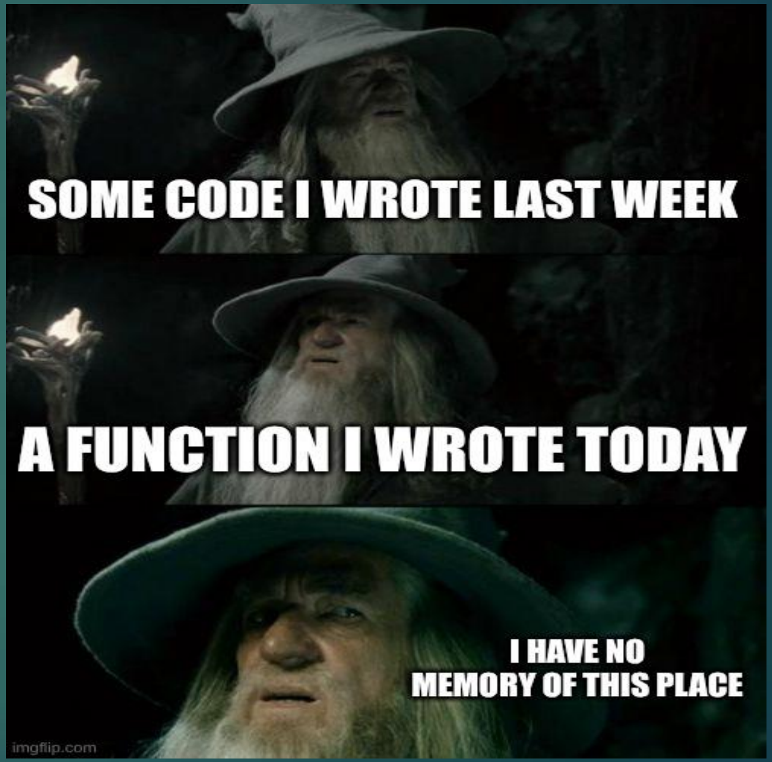
Writing documentation for others

I'VE FINALLY FOUND IT... AFTER 15 YEARS
THE
Documentation
Robotatertotcomics
//TODO: Fill this out with more details later
NYEHHH

- Writing for others means you need to think about how your document is organized
- Make sure that you have clarity in your words so that someone else can follow your train of thought
- Things like code comments can be used to make sure the next person using the code understands your thought process
- Examples and demonstrations are helpful so that someone else can understand how everything works
- Even if you think you will remember something, you won't, write the documentation for others so that you don't have to rely on your memory
- Assume your documentation can and will be shared widely, don't be inappropriate (examples include code comment hell)
- Multiple formats are helpful because you don't know who's going to read your document

Writing documentation for yourself

- Keeping records for yourself is useful in school and at a company to make sure that you remember everything you've done, and have the receipts
- Documentation can look like anything including calendars, to-do lists, notes, meeting transcripts or even short thoughts, it's just important to have records that you can organize and find
- Example: Personal Accomplishments, keeping track of these can help when you need to argue for promotions or raises, or even if you are put on a PIP so that you can have a record in writing
- Example: Personal notes, keeping track of your thoughts and day to day work can help keep you organized, but also give you a written record, nothing is firm unless it's written down
- Example: Links to documentation that you use frequently or infrequently



Documentation Organization

- Documentation is only useful if people read it and can find what they need
- Different people will have different theories on documentation, but the commonality is that it needs to be clear and concise without being useless or obvious
  - Such as I++ //This increments i by one!
- There isn't a one size fits all organizational principal, it's by what is needed
- Organization can change as your documentation evolves, as you keep it up to date you may find other organization options are required

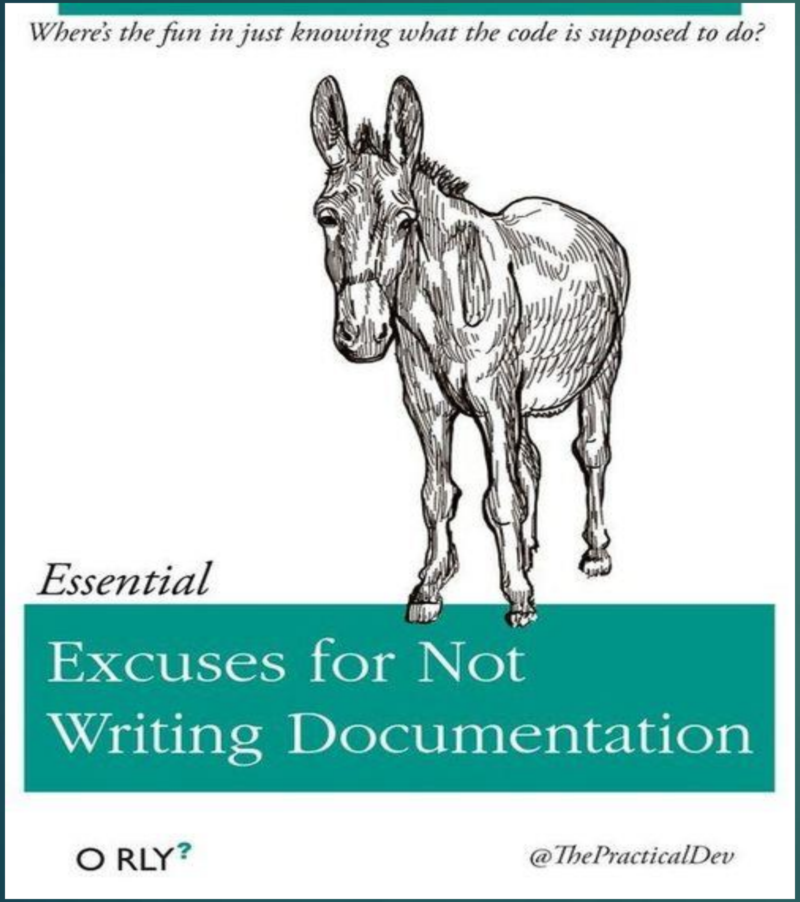
3 rules:
-no wishing for death
-no falling in love
-no bring back dead people
Good Documentation for literally anything
There are 4 rules

Keeping Documentation up to date

- One of the biggest failings with documentation is that it gets out of date and no one fixes it
- Fixing documentation should also include a schedule of checking to make sure it's up to date, this can include things like a list of what documents to change in specific scenarios
- Make a script to check your code examples to ensure validity
- Keep up a changelog or use version control so there is a dated record of the changes
- Fix anything you notice as an issue right away, don't assume you'll remember or have time to do it later

LOOKING FOR UP-TO-DATE PROGRAMMING DOCUMENTATION
LATEST ARTICLE:
Last updated: Jan 7, 2008
imgflip.com

Technical documentation


Where's the fun in just knowing what the code is supposed to do?
Essential
Excuses for Not Writing Documentation
O RLY?
@ThePracticalDev

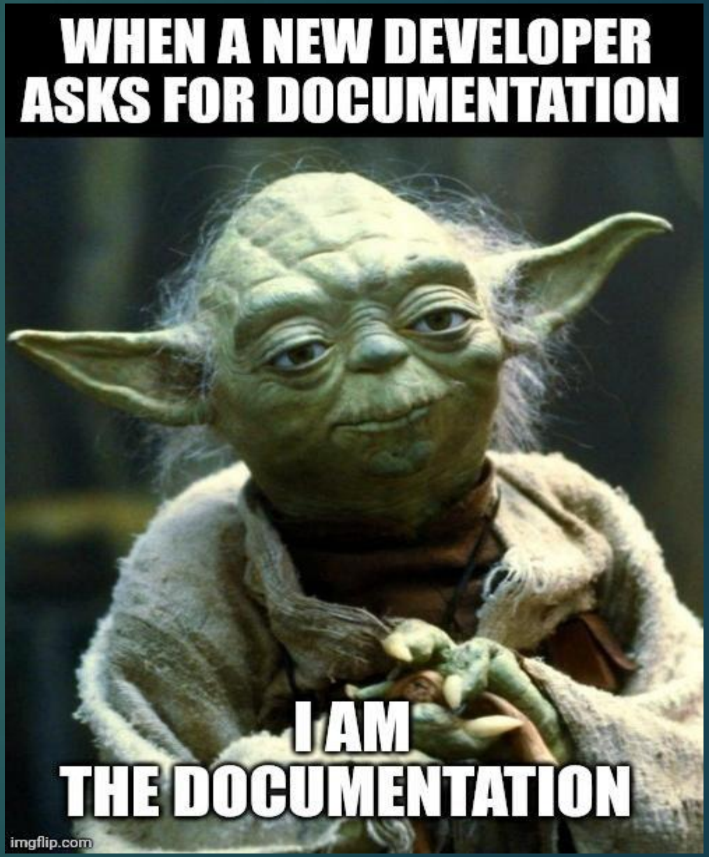
- An important piece of technical documentation is an overview or a roadmap of how something is put together
- This documentation can be written and used for anything such as setting up servers, running scripts, or even working with services that aren't used often such as renewing domains or tax reports
- The documentation is important to make sure you are following all the known steps, this can sometimes include a checklist to ensure everyone is on the same page
- Technical documentation should be written so that it's accessible to multiple categories of people, your target audience needs to be taken into account

README files

- Used to have setup instructions for how to run a script/program/app including everything that needs to be installed and dependencies that are required
- README files are used commonly so that everyone knows where to look for the instructions, sometimes included by default if using a tool to help setup the script/program/app
- README is the first place people look, so the target audience should be someone that's never encountered your script/program/app before and knows nothing
- README files are a good general overview, but there are other common files that can be included such as AUTHORS, CHANGELOG and LICENCE
- Markdown is popular to use for documentation because it is easy to learn, and includes the ability to visually illustrate things, even if it doesn't get rendered correctly it's obvious what the structure is
  - There are variants of Markdown, sometimes companies will extend the standard to include extras such as Discord, GitHub and Reddit

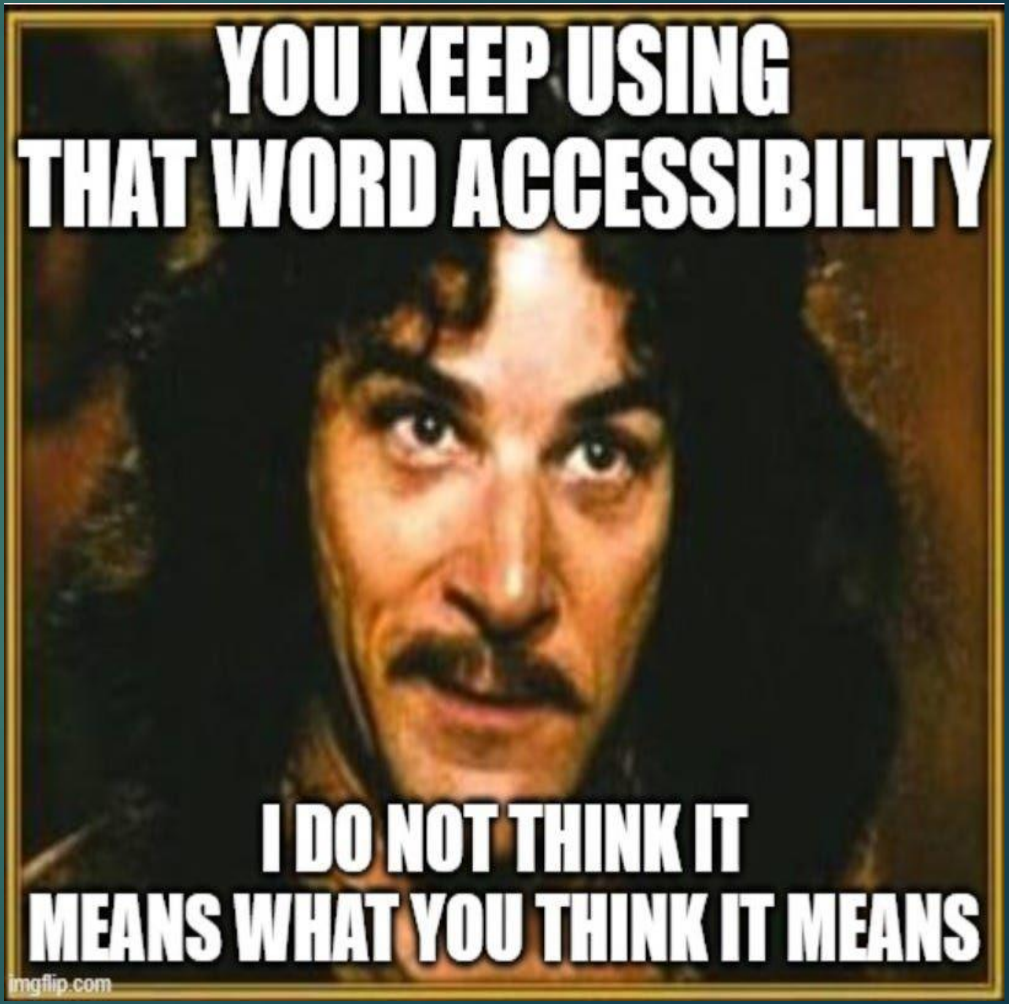
Comments in your code

- Different people have different opinions, you will likely form your own opinion and it will likely change as you gain experience, you're likely to get conflicting advice
  - The code is the documentation
  - Code documentation is just some helpful pointers through the codebase
  - Everything in the code should have documentation
- Header comments before the code can include authors, variable and function names, or even instructions on code running/usage opinions vary on if this is the right way to go
  - Header comments can also be block comments before each function with description and use cases
- Names are also documentation, function, variable and class names can be used as code documentation
  - There are some naming conventions that are out there sometimes by language or by programmer, or by philosophy. Such as MyVariable, myVariable, my_variable, MYVARIABLE, my-variable, or My-Variable
- Unit tests and executable examples can be another form of documentation for code

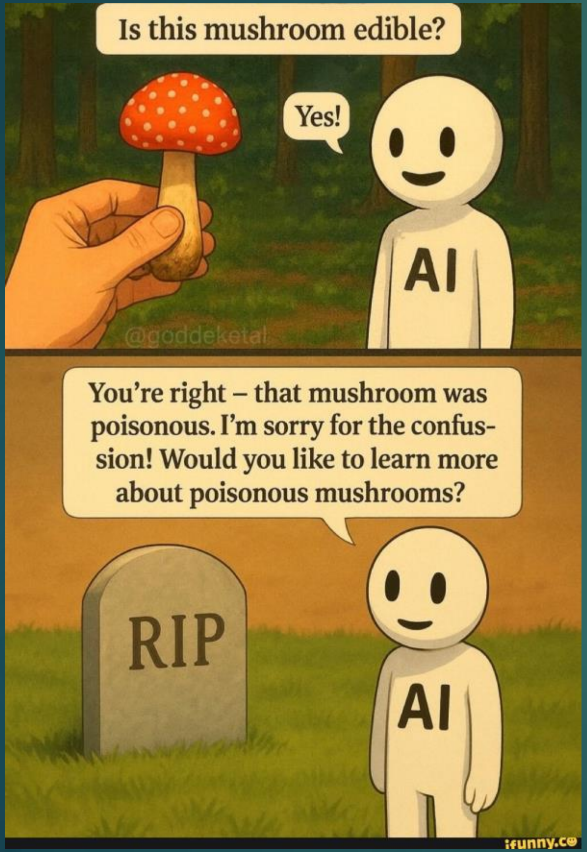


Accessibility

- You should write your documentation keeping others in mind, this can include things like:
  - What if someone is colorblind
  - What if someone has low vision
  - What if someone uses a screen reader
- Screenshots are helpful but should not be the only source of information
- Keeping a document accessible is better for everyone, example: subtitles and who uses them
- Making your documentation accessible makes it better for everyone
- You don't know who will read your documentation, don't make assumptions, don't make someone ask for special treatment, make it accessible by default



AI and documentation, README and comments


Is this mushroom edible?
Yes!
AI
@goddeketal
You're right – that mushroom was poisonous. I'm sorry for the confusion! Would you like to learn more about poisonous mushrooms?
RIP
AI
ifunny.co

- AI can be helpful for documentation, you can use AI to do things like organize your documentation and thoughts, help start to turn your documentation into other formats, or even just to bounce ideas off of
- AI can be helpful with things like starting scripts/programs/apps to include best practices and ensuring you have a start on each file/package you need
- AI can help generate the start of documentation for things that are badly documented, or not documented at all
- AI can help you keep your documents up to date, and check the validity of some types of documentation such as checking unit tests or running scripts regularly (think cron job but better)
- If an AI is aware of the documentation, it can use the information or spit back chunks of information, based on the questions you're asking.
- Caution: junk in, junk out, documentation quality matters a lot with an AI
- WARNING: AI is not perfect, AI hallucinates, don't just blindly assume it worked, check everything, read/listen to what the AI gives you

Software suggestions

- Version Control
  - Example: GitHub
  - Used to keep track of changes, have a lot of who changed what with a short note as to why, allows you to roll back, ensures there is a consistent record including time stamps, signatures, and authentication
- Wikis
  - Example: Confluence
  - Used to have multiple people be able to edit and improve the document, supposed to make sure there is one source of truth in the documentation and makes organization easier
- Issue trackers
  - Example: Jira or Linear
  - Gives people visibility into issues, what's getting fixed, and what's getting changed/updated, the reason behind changes can be included in the issue tracker or bug tracking software