#21DaysOf#DrGViswanathanOpenSourceChallenge

The goal is to have some contributions merged *before* Day 21, allowing for a strong presence on your GitHub profile by the end of the challenge.

Phase 1: GitHub Fundamentals (Days 1-7)

Day 1: GitHub Account Setup and Exploration

- Create a GitHub account, if not.
- Explore the GitHub interface: dashboard, profiles, repositories, issues, pull requests.
- Understand the concept of a repository.
- Learn about Markdown for `README.md` files.
- **Here's The challenge — Create your own custom readme file for your profile. Look for some open source readme collections , and pick your best selections from their readme, to add to yours. Make it as cool as you can… because it's yours !**

Day 2: Git Basics - Local Setup

- Install Git on your local machine.
- Configure Git with your username and email.
- Learn basic Git commands: `git init`, `git add`, `git commit`.

Day 3: Git Basics - Remote Connection

- Create your first repository on GitHub.
- Link your local repository to the remote GitHub repository: `git remote add origin`, `git push`.
- Learn `git pull` to fetch changes from the remote.
- **Here's the challenge for you. NEVER PUSH ANY CODE , FROM THE GITHUB UI. always push it through the `git push` command shared above, throughout the entire challenge. Get in a habit of pushing to Git through cli.**
  Make your first push. If not your first push… then , make it a cli command based push.

Day 4: Branching and Merging

- Understand why branching is important (`git branch`, `git checkout`).
- Create new branches, make changes, and commit them.
- Learn to merge branches (`git merge`).
- **Here's the Challenge for you. Practice resolving simple merge conflicts.Create a small merge conflict yourself. You'll learn what are conflicts as you go. Try to solve it. Take screenshots along the way. Show your progress on LINKEDIN**

Day 5: Forking and Cloning

- Understand the difference between forking and cloning.
- **Fork a simple repository (e.g., a "hello world" repository or a list of beginner-friendly projects).**
- **Clone your forked repository to your local machine**.

Day 6: Understanding Issues and Pull Requests

- Explore the "Issues" tab in various repositories. Understand how issues are used for bug reports, feature requests, and discussions.
- Explore the "Pull Requests" tab. Understand how PRs are used to propose changes.

Day 7: Your First Local Contribution Simulation

- On your forked repository, create a new branch.
- Make a small, simple change (e.g., fix a typo in the `README.md`, add a new line of text).
- Commit your changes locally.
- Push your changes to your forked repository. **Do not create a pull request yet.** This is just a simulation to get you an idea on your local to remote workflow. Making sure it is solid.

Phase 2: Finding Repositories and Contributing (Days 8-20)

Day 8: Identifying Contribution Opportunities - Part 1 (Documentation and Typos)

- Focus on finding repositories with good documentation.
- Look for simple typos, grammatical errors, or areas where clarity can be improved in `README.md` files, `CONTRIBUTING.md` guides, or other documentation.
- Search for repositories with the "good first issue" or "help wanted" labels on GitHub.

Day 9: Identifying Contribution Opportunities - Part 2 (Beginner-Friendly Code Changes)

- Look for repositories that have "good first issue" labels related to minor code changes (e.g., adding a simple function, fixing a minor bug).
- Explore projects specifically designed for first-time contributors.

Day 10: Making Your First Real Pull Request (Documentation)

- Choose one of the typo/documentation fixes you identified. Could be a typo error. Could be a real issue , if you have control over your language of choice.
- Fork the original repository (if you haven't already).
- Clone your forked repository.
- Create a new branch for your specific change.
- Make the change and commit.
- Push to your forked repository.
- **Create your first Pull Request** to the original repository. Write a clear and concise PR description referencing the issue (if applicable). **Share it on Linkedin !! It's an achievement**

Days 11-14: Iterative Contributions - Small Fixes

- Identify 2-3 more simple (non documentational) improvements or minor bug fixes in different repositories.
- Repeat the fork, clone, branch, commit, push, and Pull Request process for each.
- Pay attention to the project's `CONTRIBUTING.md` guidelines for each repository.

Days 15-17: Expanding Contribution Scope - Feature Additions/Refactorings

- Look for slightly more complex "good first issues" or features that require adding a small amount of code.
- This could involve adding a new utility function, improving a script, or refactoring a small part of existing code for better readability.
- Engage with maintainers on issues or in discussions if you have questions.

Days 18-20: Advanced Contribution Practice & Profile Building

- Continue searching for contributions, aiming for projects that genuinely interest you.
- Consider contributing to projects you use regularly.
- Respond to comments and feedback on your existing Pull Requests. Learn from any requested changes.
- Refine your commit messages and PR descriptions for clarity and professionalism.
- **By the end of Day 20, you should have several contributions (Pull Requests) submitted, with some potentially merged.**

Phase 3: Reflection and Future Steps (Day 21)

Day 21: Review and Plan

- Review your GitHub profile. See all the contributions you've made.
- Reflect on what you've learned about Git, GitHub, and open source collaboration.
- Identify areas where you feel confident and areas where you still want to improve.

- Plan your next steps:
    - Continue contributing to the projects you've started.
    - Explore new projects.
    - Consider creating your own open-source project.
    - Learn more advanced Git concepts (rebase, cherry-pick, stashing).

Good luck!