

# Git Gud!

## Discussion

By Sander Beekhuis

**What did you learn today?**

# When do you use git rebase?

1. Never, changing history is dangerous
2. If you have not yet pushed the changes
3. If you are changing "your" branch
4. Always, as long as you improve the history of the repository

# How do you handle it when your feature has merge commits with the main branch.

1. I rebase my feature
2. I merge `main` into my feature

# How do you prefer to merge feature branches?

	pure git	Azure Devops	Github
1.	<code>git merge</code>	Merge (no fast-forward)	Create a merge commit
2.	<code>git merge --squash</code>	Squash commit	Squash and merge
3.	<code>git rebase main &amp;&amp; git merge --fast-forward</code>	Rebase and fast-forward	Rebase and merge
4.	<code>git rebase main &amp;&amp; git merge</code>	Semi-linear merge	N/A

# Do you look at commit history when reviewing code?

1. Yes, and I ask my colleagues to rebase when it doesn't make sense.
2. Yes, but if it is not that great I just ignore it.
3. No, not really

# The end

- Questions?
- Feedback welcome