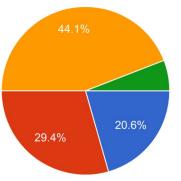
SMP git and github workshop

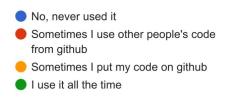


https://github.com/benroberts999/git-workshop

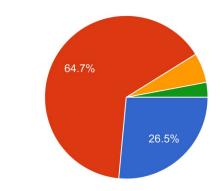
Ben Roberts https://broberts.io/teaching

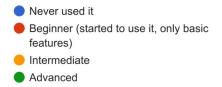
Have you used github.com (or gitlab or similar)? 34 responses



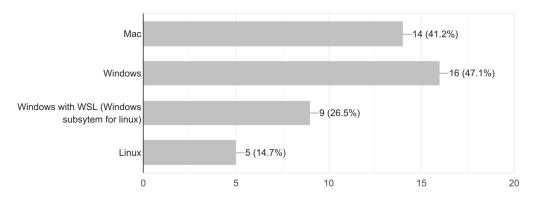


What's your current experience with git? ³⁴ responses

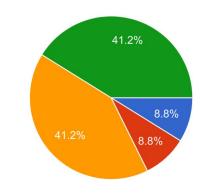


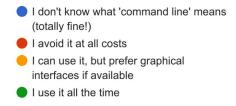


What operating system do you use for coding? ^{34 responses}



How comfortable are you with using the command line? ³⁴ responses

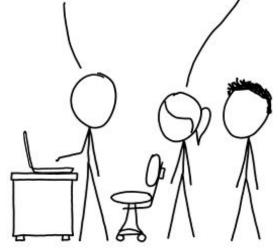




THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOU DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



git

- Version control
 - Snapshots (entire project), called "commits"
 - Stores diff/delta only: great for plain text
 - Easily re-wind, backtrace errors

• Branching

- Simultaneous development
- Edit code without impacting working code
- Merge into main branch when working

Remote synchronisation

- Share and store code remotely
- Allows collaboration
- Github: website to host projects "repositories"

• Somewhat steep learning curve

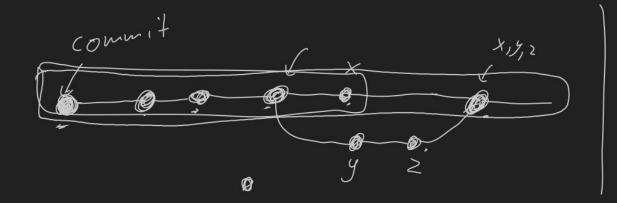
- Ugly interface: can wrote learn, but
- Basic understanding helps immensely
- Interactive workshop

xkcd.com/1597/

Git: basics

- Snapshots of entire project: commits
- Commit: what, who, why
- Graph: commits are nodes
- HEAD is current node
- Branch (give name to nodes)
- Remote

- Commit contains:
 - Data (form of a diff)
 - Author
 - Message
 - Parent(s) (preceding commits)
 - Id: hash of the data





Workshop Tasks

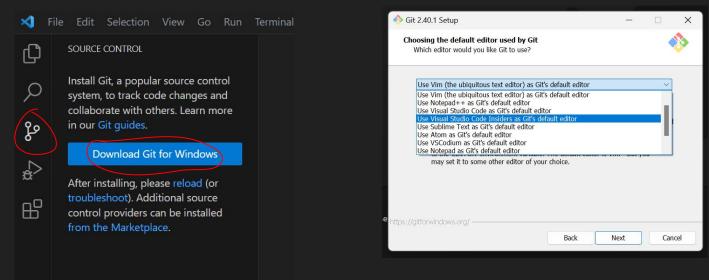
Head over to: <u>https://github.com/benroberts999/git-workshop</u>

Pre-work

- Make yourself a github user account <u>https://github.com</u>
- Install Visual Studio Code: <u>https://code.visualstudio.com/</u>
 - Install git (next slide)
- (optional) If you want to use the command line, set up ssh keys:

https://docs.github.com/en/authentication/connecting-to-github-with-ssh

Install git on windows (use vscode)



Install on mac: https://www.atlassian.com/git/tutorials/install-git

Comes with Xcode, or install using brew, MacPorts, or from <u>git-scm.com/</u>

\$ git config --global user.name "Emma Paris"
\$ git config --global user.email "eparis@atlassian.com"

Extensions/tools

Valuable VSCode extensions:

• <u>Git Graph</u>

• Show visible git graph =>

👢 Git Graph 🛛 🗙

• <u>GitLens</u>

• Many other nice/useful tools

∢	File	Edit	Selection	View	Go	Run	
Ŋ		EXTENSI	ONS: MARKET	PLACE	7	ඩ ≣	× ···
		git					
م			Git Graph View a Git mhutchie		of yoı		り 78ms sitor 殺み
₽ 2		P	Git Histo View git lo Don Jayan	og, file h	istory		
		N	GitLens – Superchai				le — ⊡ …

	Branches: Show All Show Remote Branches		Q 🖸 🗘	မ္ပ
Graph	Description	Date	Author	Commit
•	Porigin/VQE) merge			
	P origin/vqe2 Writes 1 and 2 particle integrals to file	4 May 2023 15:49	Ben Roberts	d49fd075
	New branch	4 May 2023 15:34	Ben Roberts	f7ab4c2f
9	P origin/gh-pages Make documentation [GH Actions]	2 May 2023 16:43	github-actions[bot]	4f8d2e11
6	• 🔽 dev origin 😰 origin/HEAD 😰 origin/main Moves 'subtract 1' feature into jL operator; previ	2 May 2023 16:00	Ben Roberts	6cb94560
	Updates install.sh for M1 mac	31 Mar 2023 11:12	Ben Roberts	
•	🦻 origin/tst misc	30 Mar 2023 17:30	Ben Roberts	ecd3cdb9
- •	Update	30 Mar 2023 17:04	Ben Roberts	
	Update to command-line options	24 Mar 2023 14:53	Ben Roberts	7e7a5922
	Minor change: Makes Adams compatible with gcc7	23 Mar 2023 10:38	Ben Roberts	b64e70f9
	Adds VQE module (and minor Coulomb function update)	22 Mar 2023 15:10	Ben Roberts	dc9d7bfd
	Minor: Update VQE outputs	22 Mar 2023 14:18	Ben Roberts	169be5c5
	Minor: update to k_minmax_W	22 Mar 2023 09:47	Ben Roberts	49fc571d
	Adds VQE test	21 Mar 2023 19:23	Ben Roberts	ffe0a351
	Lists CSFs	21 Mar 2023 16:01	Ben Roberts	69ef9115
	VQE: initial working	21 Mar 2023 13:24	Ben Roberts	26dcd91d
	Adds fill_ab function for QkTable: wills Q_abab and Q_abba integrals	21 Mar 2023 12:25	Ben Roberts	891e30b7
	calculates single-particle MEs	21 Mar 2023 10:46	Ben Roberts	c69e3aa1
	Begins VQE module	21 Mar 2023 10:20	Ben Roberts	a8497c54
	Adds fmt2:: - automatically only styles output text if is a tty	20 Mar 2023 18:08	Ben Roberts	98857c48

Summary of tasks

- 1. Fork + clone repo (or create new one)
- 2. Add some changes: commit
- 3. Push to github
- 4. Make changes on github, and pull them down
- 5. Create a new branch: make commits and swap between branches
- 6. Merge our branches



Fork and/or clone (1-2)

t-worksh	nop (Public)				St Pin 💿)Unwatch 🗻 👻	양 Fork 3 → ☆ St	
Pull reques	sts 🕞 Actions 🖽 Projects 🖽	Wiki 🛈 Security 🖂 Insights 🕸 S	ettings					
	알 main → 알 1 branch । ⊙ 0 tag		Go to file Add file -	<> Code •	About No description, website, o		\$	
	benroberts999 Prepare repo for :	SMP git workshop 2023		🕑 10 commits	provided.	n topics		
	🗅 .gitignore	Cleanup readme		last year	🖽 Readme			
	LinuxCommandLine.md	Prepare repo for SMP git workshop 202			☆ 0 stars ⊙ 1 watching			
	C README.md	Prepare repo for SMP git workshop 202			ণ্ণ 3 forks			
	🗋 WorkshopTasks.md	Prepare repo for SMP git workshop 202:						
	E README.md			P	Releases			
	E README.md					Q	SOURCE CONTROL	
•		repo to your gi				م م	In order to use git features, you of folder containing a git repository from a URL. Open Folder Clone Repository	
•	Cione: dowi	nload git repo	to your po	7		ß	To learn more about how to use source control in VS Code read o	
•	git clone git@git	hub.com:benroberts	999/git-works	shop.git				
•	Or, GUI =>					6		
•	Or, initialise a ne	ew repo with 'git init'						

• DON'T put a git repo inside dropbox/onedrive etc.

Interlude: ssh keys

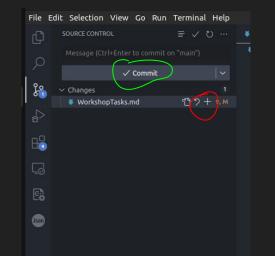
• To set up ssh keys for github:

https://docs.github.com/en/authentication/connecting-to-github-with-ssh

- If you use VSCode, you won't need to do this
- To use command line, you will
- Generate a public/private key pair (if not already)
- Upload public key to github
- Be careful, follow above instructions

Make and commit some changes (3-4)

- Use git diff to see your changes
- Stage and commit your changes
 - Git status
 - git add <...>
 - git commit -m <your commit message>
 - Or, commit using the source control tab in VSCode
 - See: <u>https://www.gitkraken.com/learn/git/best-practices/git-commit-message</u>



Remote

- Concept of a "remote"
 - Synchronise changes
- Named: by convention/default: "origin"
- If pulled from github: already set up

- git remote -v
 - List remotes with URL
- git remote add <name> <url>
 - Adds a new remote

- We can push/pull changes to/from remote
 - Keeps them syncd
- https://www.atlassian.com/git/tutorials/syncing

URCE CONTROL	≣ ✓ Ŭ ···	③ README.md >	×
lessage (Ctrl+Enter to commit or		Views] # Git and GitHub Workshop
		View & Sort	J GitHub Workshop
		Pull Push Clone Checkout to Fetch Commit Changes Pull, Push	<pre>version control soft s you to take snapshc asily track the char by storing the _diff repository typically d, it works best for mpressed files. allows _branching - version of the proje f. When you're happy</pre>
		Branch Remote	<pre></pre>
		Stash	Remove Remote
		Tags	repositories.
		Show Git Output	orials available:

ŶР

Json

Push changes up to your github (5)

- Push changes up to your github 'remote', and look at changes on github.com
 - \circ git push origin main
- Check on github: see that the changes are there

Now, make a change on github (6-7)

- git status
 - O it should say "up to date", because is doesn't know about the upstream changes
- git fetch
 - This will grab the info from 'upstream', but not download the changes
- git status
 - Now, it should say there are changes to be pulled
- git pull
 - We should now have those changes locally

Branch (8)

Make a new branch, check it out

- * `git branch <branchname>`
- * then: `git checkout <branchname>`
- * Or: `git checkout -b <branchname>` (does both at same time)

Use `git branch` to list available branches

* Then make changes and commit them as before

* `git log` (or `git log --graph --oneline`)

Branch (8)

Or, use GUI:

File	Edit Selection View Go Run Terminal H	elp	
Ð	SOURCE CONTROL 臣 🗸 🖔 👢	···· ① README.mc	t x
_	Message (Ctrl+Enter to commit on "main")	Views	▶] # Git and GitHub Workshop > ## Getting
	√ Commit	View & Sort	J GitHub Workshop
fo		Pull	
		Push	version control software.
\$		Clone	s you to take snapshots (called ∢ easily track the changes made to
~~		Checkout to	by storing the _difference_ betw
₿		Fetch	repository typically doesn't gro , it works best for plain text t
		Commit	 > mpressed files.
ΓĢ		Changes	allows _branching - allowing ye
6		Pull, Push	version of the project while lea
ЦŖ		Branch	 Merge Branch
Json		Remote	Rebase Branch Ironise (
-		Stash	 Create Branch github
		Tags	 Create Branch From
		Show Git Outp	Rename Branch
		14 * gi	Delete Branch
		gett	ing- Publish Branch
			Also others, e.g.:
			<https: git="" www.atlassian.com=""> <https: git="" td="" tutor:<="" www.atlassian.com=""></https:></https:>
			chttns+//ait-scm com/hook/en/v/> :

Branch: push new branch

- * push this branch up to github.com: `git push origin <branchname>`,
 or `git push --set-upstream origin <branchname>`
- * Swap back/forth between branches:
 - `git checkout main`
 - `git checkout branchname`

File Edi	t Selection	View Go	Run Tern	ninal Help		
Q	SOURCE CONTRO)L	e 🗸	0 ₩ …		
	Message (Ctrl+Enter to commit on "main")					
~		🕂 Sync Cha	anges 1↑			
fo						

File Edi	it Selection View Go Run Terminal Help		
C	SOURCE CONTROL 🛛 🗄 🗸 🔤	() README.md \times	
		Views 🕨] # Git and GitHub Workshop
	O Sync Changes 1↑	View & Sort 🔹 🕨	
ze		Pull	nits
		Push	
à		Clone	version control software.
		Checkout to	s you to take snapshots (called
₿		Fetch	easily track the changes made by storing the difference be
			repository typically doesn't (
		enenges	<pre>i, it works best for plain tex</pre>
C.)		Pull, Push ►	wing
		Branch ► Remote ►	Pull ile the r
Json		Stash	Pull (Rebase)
		Tags >	Pull from onisi
			Push gith
		Show Git Output	Push to
		14 Some cu	Fetch
		16 * git tu getting∙	
		17 * Also	
			tps://www.atlassian.com/git>

Merge (9)

Three ways this can go:

- "Fast forward"
 - No changes on destination branch

- Merge
 - Non-conflicting changes on both branch
 - Creates a new "merge commit"

ile Ed	it Selection View Go	Run Terminal Hel	P				
Ð	SOURCE CONTROL	■ ✓ ひ 幕 …	• (i) README.m	ıd ×			
			Views		# Git and GitHub Workshop		
	↔ Sync Ch	anges 1↑	View & Sort				
ze			Pull		its		
			Push				
a⊳ æ			Clone		version control softwa	re.	
			Checkout to		you to take snapshots	(called	
₿			Fetch		asily track the change by storing the differ		
6			Commit		repository typically d	pesn't (
			Changes		, it works best for pl mpressed files.	ain tex	
C)			Pull, Push Branch			'lowing	
			Remote		Rebase Branch	/hile 1 the 1	
Json			Stash		Create Branch		
			Tags		Create Branch From	versi	
			Show Git Out	out	Rename Branch	gith	
					Delete Branch		
					Publish Branch		
			<pre>16 * git tutorial from GitHub: <<u>https://do</u> getting-started-with-git></pre>				
				Also	others, e.g.:		
				<http: -http: -http:</http: 	s://www.atlassian.com/	<u>git></u>	

• Conflict

- Changes could not be auto-merged
- Manually fix/decide, create new commit
- Use a good text editor like VSCode

Conflict (10)

If you are feeling brave: create a conflict

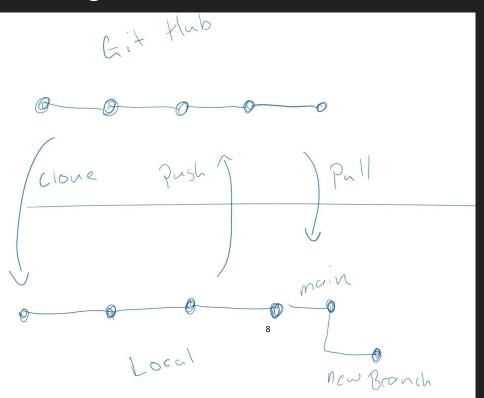
- change (+commit) same part of a document on both branches before merging

 BR:git-workshop\$ git merge newbranch Auto-merging README.md CONFLICT (content): Merge conflict in README.md Automatic merge failed; fix conflicts and then commit the result.
 BR:git-workshop\$

git

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes <<<<< HEAD (Current Change) git is an awesom version control software. ======= git is a great version control software. >>>>> newbranch (Incoming Change) It allows you to take snapshots (called commits) of your project, so you can easily track the changes made to a project

Whiteboard Drawings



Whiteboard Drawings

