

Version Control Using Git and GitHub: One Way to Better Manage Research Workflows

Kee Hyun Park

University of Maryland

GVPT Methods Workshop

September 22, 2020

Things that can mess you up during research

While conducting research, we may...

- have 9 different versions of a paper. Which is the latest?
- need to revert changes made to a data file 3 months ago. How?
- mistakenly delete an important CSV file completely
- have been exchanging ideas via email thread. Now it's over 100..
- share R scripts with coauthors. Which part did they change?

...and lots, lots more...

What if...

- we have a system that records changes to a set of **same files over time**
- so allows us to
 - **revert** particular file / entire project back to a previous state
 - **compare** changes over time
 - **see who modified** the contents and how
 - ***without thinking much*** about the file's location and contents?

~ exactly what version control is/does

Git: a version control system

- Git is a type of software for version control (think MS Word's "Track changes")
 - takes "screenshot"s of all files ↪ come back to previous versions anytime
 - shows detailed changes in *text-based files* (.txt, .tex, .csv, .R, and much more)

The screenshot shows a GitHub commit interface. On the left, a diff view highlights changes in a LaTeX file named `adjusted soybean_fig1.1.3/1.1.4`. The commit was made by `keehyunpark` 25 days ago. The commit message is `adjusted soybean_fig1.1.3/1.1.4`. The commit hash is `f46e79e`, and the commit message is `commit 24cddb3d7c1bd731e3f7b56b46f8589c15d9ab5`. The commit has 1 parent. The changes are shown in a unified diff format.

adjusted soybean_fig1.1.3/1.1.4

keehyunpark committed 25 days ago

1 parent f46e79e commit 24cddb3d7c1bd731e3f7b56b46f8589c15d9ab5

Showing 5 changed files with 10 additions and 3 deletions.

Unified Split

codes/soybean_fig1.1.R

keehyunpark committed 25 days ago

1 parent f46e79e commit 24cddb3d7c1bd731e3f7b56b46f8589c15d9ab5

Showing 5 changed files with 10 additions and 3 deletions.

Unified Split

... 4 codes/soybean_fig1.1.R ...

@@ -83,-5 +83,5 @@ axis(side = 2, at = seq(0,1000,200),cex.axis=2)

83 mtext("Frequency", side = 2, line = 3.2, cex=2.5)

84 mtext("Change in Crop Planting (%)",

85 side = 1, line = 3.5, cex=2.5)

86 - text(53,5,880, labels = paste(length(which(change\$chp,soypt>0)),"/3118 (23% US counties",sep=""),cex=2)

87 - text(53,5,737.5, labels = paste("increased % of soybeans",sep=""),cex=2) ⚡

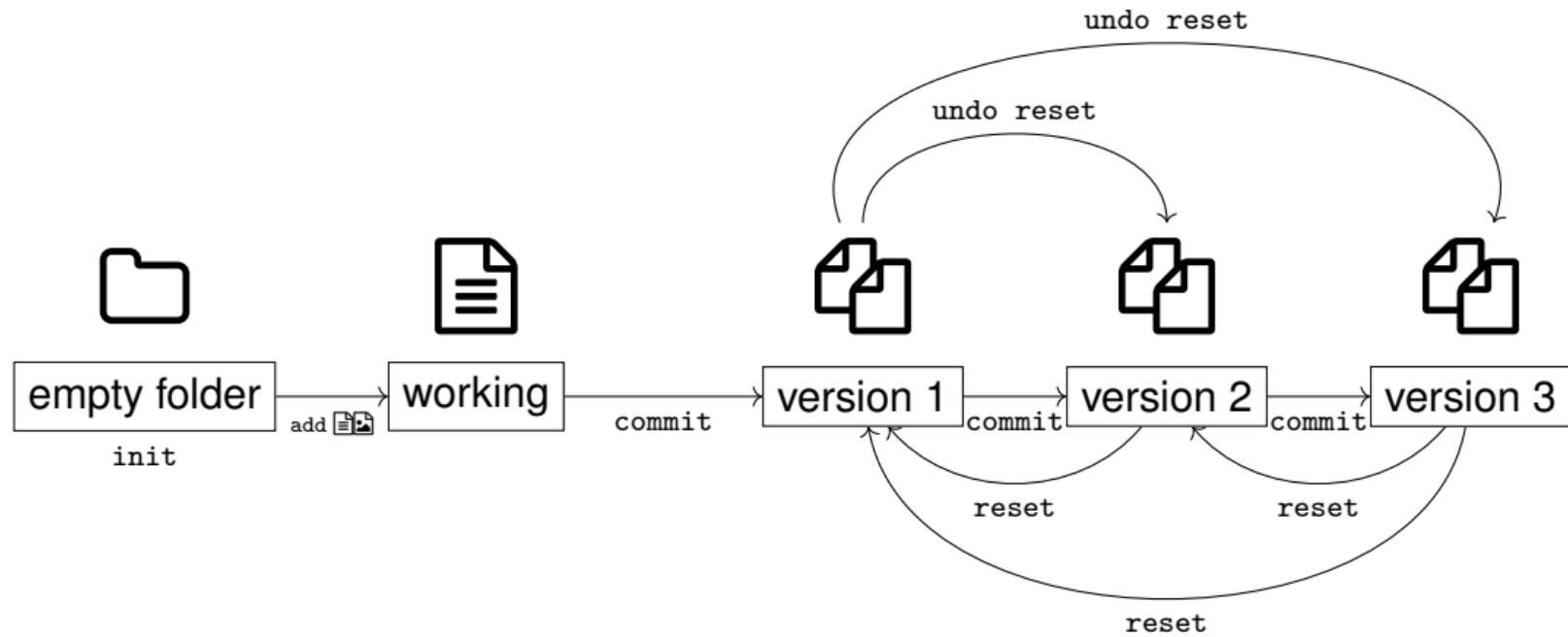
86 + text(-55,680, labels = paste(length(which(change\$chp,soypt>0)),"/3118 (23% US counties",sep=""),cex=2)

87 + text(-55,537.5, labels = paste("increased % of soybeans",sep=""),cex=2) ⚡

Tracking changes in TeXfile

Tracking changes in R script

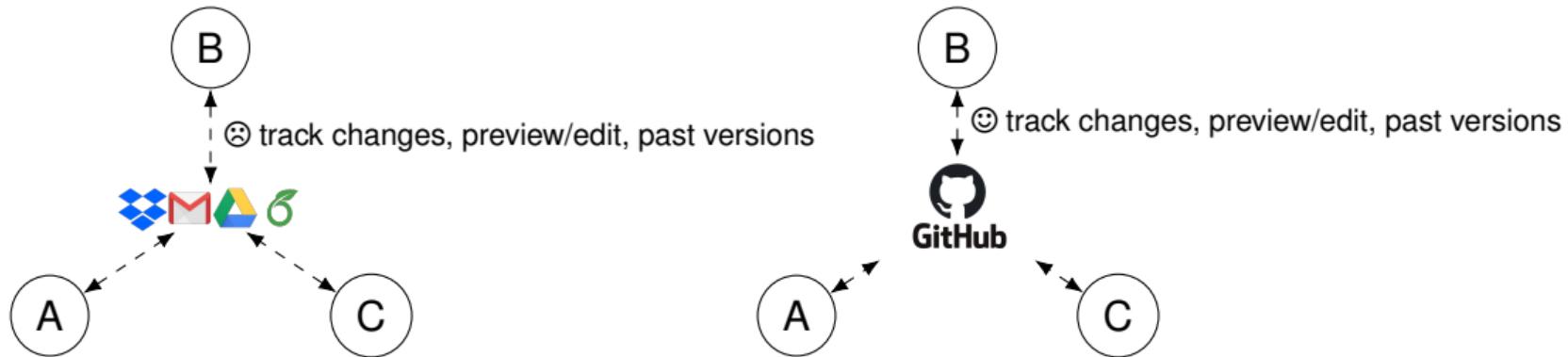
How Git works on local computer



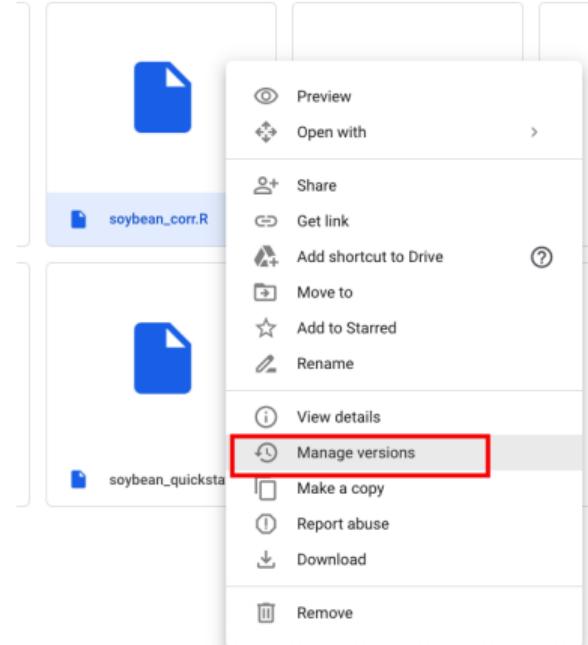
Git can take screenshots & show details of each and every step

GitHub: a platform that connects git users

- GitHub is a git-based online platform that connects individual git users
 - think Dropbox, Google Drive, Gmail, or Overleaf



GitHub vs. Others



A screenshot of the 'Manage versions' dialog box. It shows three versions of the file 'soybean_corr.R':

- Current version: soybean_corr.R, Sep 4, 6:11 PM, Kee Hyun Park
- Version 2: soybean_corr.R, Sep 4, 5:53 PM, Kee Hyun Park
- Version 1: soybean_corr.R, Sep 2, 9:37 PM, Kee Hyun Park

A red bracket on the left side of the dialog box groups the three versions. A red arrow points from the 'Manage versions' option in the Google Drive context menu to this dialog box. A 'CLOSE' button is at the bottom right of the dialog box.

Google Drive

GitHub vs. Others

Dropbox > talks > versioncontrol

Overview Hide

Click here to describe this folder and turn it into a Space [Show examples](#)

Create new file ▾

Name	Modified	Members	⋮
.Rhistory	8/27/2020, 6:48 PM	Only you	...
git.md	8/27/2020, 2:23 AM	Only you	...
notes_git.txt	9/2/2020, 1:47 PM	Only you	...
README.md ☆	8/27/2020, 2:24 AM	Only you	Share ⋮
slides_vc.pdf	8/27/2020, 6:24 PM	Only you	Share
slides_vc.Rmd	8/27/2020, 6:24 PM	Only you	Download
slides_vc.tex	8/27/2020, 6:24 PM	Only you	Send with Transfer
timon.jpg	8/21/2020, 9:59 AM	Only you	Add comment
			Star
			Version history
			Rename
			Move
			Copy
			Delete
			Pin to versioncontrol
			Pin to...

The screenshot shows a list of files in a Dropbox folder named "versioncontrol". The files include ".Rhistory", "git.md", "notes_git.txt", "README.md" (which has a star icon), "slides_vc.pdf", "slides_vc.Rmd", "slides_vc.tex", and "timon.jpg". The "README.md" file's context menu is open, with the "Version history" option highlighted by a red box. A red circle also highlights the "Share" button in the menu.

Dropbox

GitHub vs. Others

The screenshot shows the Overleaf interface for a project titled "working paper". On the left, there's a file browser with "main.tex" selected. The main workspace shows the LaTeX code: `131 \end{document}`. At the top, there are buttons for "Label this version", "Compare to another version", and "Download project at this version" (which is highlighted with a red oval). To the right of these are "All history" and "Labels" buttons. A sidebar on the right shows a log entry: "Mon, 17th Aug 2020" with "Edited main.tex" at "3:00 pm • You". Below the log, it says "You're currently seeing the last 24 hours of changes in this project." It also lists "Upgrade to get full Project History, plus:" followed by a bulleted list of features: "✓ Unlimited projects", "✓ Multiple collaborators per project", "✓ Full document history", "✓ Sync to Dropbox", "✓ Sync to GitHub", and "✓ Compile larger projects". A "Start Free Trial!" button is at the bottom.

Overleaf

GitHub vs. Others

	GitHub	Google Drive	Dropbox	Overleaf
collaborator	∞	∞	∞	1
max file number	∞	∞	∞	2,000
max indiv. upload	100MB	50MB (text)	50GB	50MB (zip)
basic storage	1GB	15GB	2GB	∞
track changes	Yes	Google Docs	Dropbox Paper	Yes
past versions	∞	30 days	30 days	24hrs
web preview/edit	text-based files	limited	limited	TeX

Web-based version control systems (basic plan)

How do we use Git and GitHub?

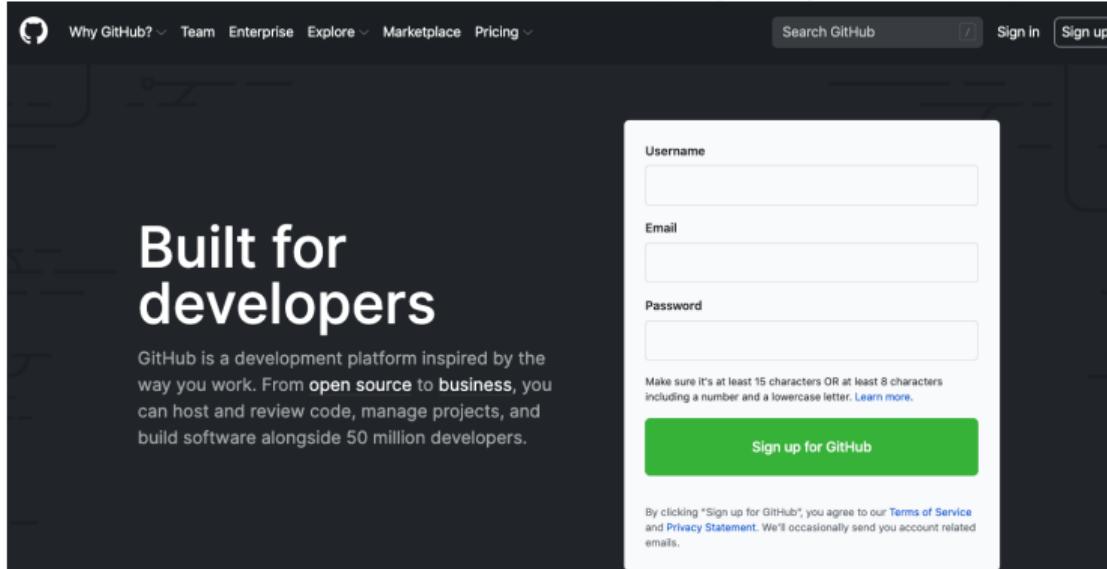
1. Install Git on your local computer (<https://git-scm.com/download>)

The screenshot shows the official Git website at <https://git-scm.com/>. The main navigation bar includes links for "About", "Documentation", "Downloads", and "Community". The "Downloads" section is currently selected and expanded, showing options for "GUI Clients" and "Logos". Below this, a sidebar highlights the availability of the "Pro Git book" by Scott Chacon and Ben Straub. The main content area is titled "Download for macOS" and provides several installation methods:

- Homebrew**: Instructions to install Homebrew if not already present, followed by the command `$ brew install git`.
- Xcode**: Notes that Apple includes a binary package of Git with Xcode.
- Binary installer**: Mentions Tim Harper's installer, noting the latest version is 2.27.0, released 2 months ago on 2020-07-22.
- Building from Source**: Notes that tarballs can be found on kernel.org, with the latest version being 2.28.0.

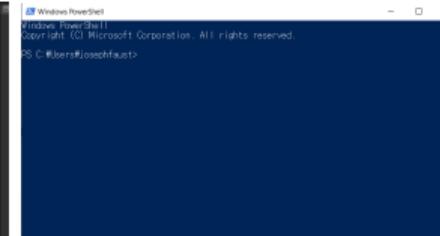
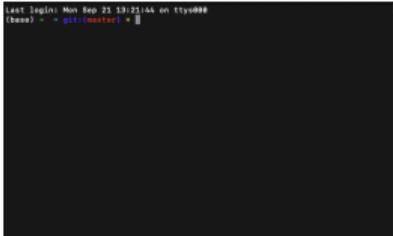
How do we use Git and GitHub?

2. Create a GitHub account (<https://github.com>)

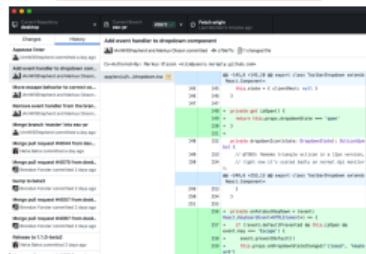


How do we use Git and GitHub?

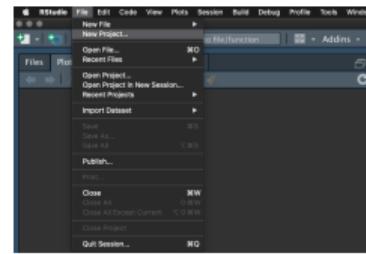
3. Choose the most comfortable mode of version control for local Git



Mac Terminal



Windows PowerShell



GitHub Desktop

RStudio

How do we use Git and GitHub?

4. Initiate Git in your interested project folder
5. Write papers, codes, produce figures, but this time version control with Git

Let's give it a try

Setup Git

- Open Terminal (Mac) or PowerShell (Windows)
 - Terminal: spotlight search → type “terminal.app” + return
 - PowerShell: press “ + R” → type “powershell” + enter
- In the Terminal/Shell,

```
git config --global user.name "yourusername"  
#recommend using your local computer username.
```

```
git config --global user.email "youremailaddress@someproducer"  
#recommend using email that you used during GitHub sign-up.
```

Assign your project folder and initiate Git

- In the Terminal/Powershell,

```
cd Downloads
```

#meaning: change directory (move) to "Downloads" folder.

```
mkdir practice
```

#meaning: make directory (folder) "practice" under Downloads.

```
cd practice
```

#meaning: move to "Downloads/practice".

```
echo "This is some text" > test.txt
```

#meaning: make a "test.txt" with a line "This is some text" written.

```
git init
```

#meaning: initiate git at the current "Downloads/practice" folder.

- Now you should have a hidden ".git" folder created under "Downloads/practice"

Record your progress

- In the Terminal/Shell,

```
git add --all
```

#meaning: temporarily record all changes that I made so far.

```
git status
```

#meaning: check changes made so far, before confirming new version.

The screenshot shows a terminal window with the following session:

```
Last login: Tue Sep 22 04:16:05 on ttys000
(base) ~ git:(master) ✘ cd Downloads
(base) ~ Downloads git:(master) ✘ mkdir practice
(base) ~ Downloads git:(master) ✘ cd practice
(base) ~ practice git:(master) ✘ echo "This is some text" > test.txt
(base) ~ practice git:(master) ✘ git init
Initialized empty Git repository in /Users/khpark/Downloads/practice/.git/
(base) ~ practice git:(master) ✘ git add --all
(base) ~ practice git:(master) ✘ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test.txt
```

A red rectangular box highlights the last three lines of the output, specifically the commit message and the staged file.

Confirm changes and make new version of work

Type:

```
git commit -m "first commit"
```

#meaning: confirm new version and add a tag "first commit".

- What you write for this tag is *very* important. The “future you” will look at these individual tags and figure out what each version actually contains.

Make second/third version

Type:

```
echo "another line" >> test.txt  
git add --all  
git commit -m "second commit"
```

```
echo "Yet another" >> test.txt  
git add --all  
git commit -m "third commit"
```

Check detailed changes

```
git diff HEAD^^ HEAD test.txt
```

#meaning: show diff between previous (HEAD^^) and current (HEAD) versions.

```
practice — git diff HEAD^^ HEAD^ test.txt — git — less + git diff HEAD^^ HEAD^ test.txt — 93x26
diff --git a/test.txt b/test.txt
index a46c98a..7dc9775 100644
--- a/test.txt
+++ b/test.txt
@@ -1 +1,3 @@
 This is some text
+another line
+Yet another
(END)
```

We're comparing the first and the third version here

Revert changes

```
git reset HEAD~2
```

#meaning: Reset to the first version.

```
git reset 'HEAD@{1}'
```

#meaning: Cancel reset.

```
git log
```

#Try 'git log' to see if reset / undo reset really worked.

Local project folder ⇒ GitHub

- We have "practice" folder with "third committed" test.txt file.
- Let's sync this with GitHub
- Go to <https://github.com/yourusername> and sign-in

Local project folder ⇒ GitHub

The screenshot shows a GitHub user profile for 'Kee Hyun Park' (keehyunpark). The profile includes a green circular icon with a white cross and four squares, a bio about deep learning for time series forecasting, and links to two repositories: 'deep-learning-time-series' and 'AIT722'. At the top, there are navigation links for Overview, Repositories (41), Projects, Packages, and a search bar. A red circle highlights the 'Repositories' link, and another red circle highlights the 'New' button in the top right corner of the repository section.

Kee Hyun Park
keehyunpark

Edit profile

3 followers · 63 following · 104

University of Maryland
College Park
khpark@umd.edu

deep-learning-time-series

Forked from Alrot0/deep-learning-time-series

List of papers, code and experiments using deep learning for time series forecasting

Jupyter Notebook 110 Apache License 2.0 Updated 24 days ago

AIT722

Forked from gmu-cit/AIT722

Course materials for AIT722: Theories and Models in Geo-Social Data Analytics

4 Updated 26 days ago

New

For example, <https://github.com/keehyunpark>

Local project folder ⇒ GitHub

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner * keehyunpark +

Repository name * / practice ✓

Great repository names are short and memorable. Need inspiration? How about [reimagined-couscous](#)?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file This is where you can write a long description for your project. [Learn more](#).

Add .gitignore Choose which files not to track from a list of templates. [Learn more](#).

Choose a license A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

Local project folder ⇒ GitHub

- Now, back to the Terminal/Shell,

```
git remote add origin https://github.com/yourusername/practice  
#meaning: connect "Downloads/practice" to GitHub repo (origin).
```

```
git push origin master  
#meaning: push "Downloads/practice" (master) to GitHub (origin).
```

- Refresh browser or visit <https://github.com/yourusername/practice> again

Local project folder ⇒ GitHub

The screenshot shows a GitHub repository page for the user 'keehyunpark' with the repository name 'practice'. The repository is private. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the navigation bar, it shows 'Code' is selected, with 'master' branch, 1 branch, and 0 tags. A red circle highlights the '3 commits' link in the commit list. The commit list shows a single commit by 'keehyunpark' titled 'third commit' made 9 minutes ago, which includes a file named 'test.txt'. A red box highlights the 'third commit' link. A red box also highlights the 'Click!' button next to the commit count. Below the commit list, there is a placeholder for a README and a 'Add a README' button. To the right of the commit list, sections for About, Releases, and Packages are visible, each with a 'Click!' button.

We can see that contents in local "practice" folder are now up on GitHub

Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

Code

master

1 branch

0 tags

keehyunpark third commit

bd2e47e 9 minutes ago

3 commits

9 minutes ago

test.txt

third commit

Add a README

About

No description, website, or topics provided.

Click!

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

© 2020 GitHub, Inc.

Terms

Privacy

Security

Status

Help

Contact GitHub

Pricing

API

Training

Blog

About

Local project folder ⇒ GitHub

GitHub shows all commit history

Click!

Code Issues Pull requests Actions Projects Security Insights Settings

master

Commits on Sep 22, 2020

- third commit keeHyunPark committed 11 minutes ago
- second commit keeHyunPark committed 11 minutes ago
- first commit keeHyunPark committed 11 minutes ago

bd2e47e
659b4c8
02957a8

Newer Older

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Local project folder ⇒ GitHub

The screenshot shows a GitHub commit page for a repository named 'keehyunpark/practice'. The commit is titled 'third commit' and was made by 'keehyunpark' 11 minutes ago. It has 1 parent commit, with the ID '659b4c0' and the commit message 'commit bd2e47e442ab50366f3aa6b775cd22215b6b816a'. The commit message also includes the URL 'https://github.com/keehyunpark/practice/commit/bd2e47e442ab50366f3aa6b775cd22215b6b816a'. Below the commit message, it says 'Showing 1 changed file with 1 addition and 0 deletions.' A diff view shows the contents of 'test.txt':

```
diff --git a/test.txt b/test.txt
index 1234567..8901234 100644
--- a/test.txt
+++ b/test.txt
@@ -1,2 +1,3 @@
 1   This is some text
 2   another line
+ 3   Yet another
```

A red box highlights the third line of the diff, which contains the text '+ Yet another'. To the right of the commit message, the text 'this is the ID of this commit' is overlaid in red. At the bottom of the commit page, there is a comment section with a 'Comment on this commit' button.

You can see the added line
(highlighted green) in third commit

Local project folder ⇒ GitHub

The screenshot shows a GitHub repository page for the user 'keehyunpark' with the repository name 'practice'. The repository is private. The 'Code' tab is selected. A commit titled 'keehyunpark third commit' is shown, featuring a file named 'test.txt'. A large red circle highlights the 'test.txt' file entry. To the left of the file list, the text 'Click!' is overlaid. On the right side of the commit card, there is an 'About' section with a note about no description, website, or topics provided, and a 'Releases' section indicating no releases have been published. At the bottom of the commit card, there is a prompt to 'Add a README' with a corresponding button. The footer of the page includes links for copyright information, terms, privacy, security, status, help, contact, pricing, API, training, blog, and about.

keehyunpark / practice Private

Code Issues Pull requests Actions Projects Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

keehyunpark third commit bd2e47e 13 minutes ago 3 commits

test.txt third commit 13 minutes ago

Add a README

About

No description, website, or topics provided.

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Local project folder ⇒ GitHub

A screenshot of a GitHub repository page for 'keehyunpark / practice'. The 'Code' tab is selected. A file named 'test.txt' is shown under the 'master' branch. The commit history shows a single commit from 'keehyunpark' titled 'third commit' made 13 minutes ago. The commit message indicates 1 contributor and 3 lines (3 sloc) with 43 Bytes. The commit content is:

```
1 This is some text
2 another line
3 Yet another
```

Below the commit content, there are buttons for 'Raw', 'Blame', and an edit icon (pencil). A red circle highlights the edit icon, and a red box with the text 'Click!' is overlaid on the commit message area. Another red box with the same text 'You can "edit" the file directly on GitHub' is overlaid on the bottom right of the commit message area.

GitHub navigation bar: Unwatch, Code, Issues, Pull requests, Actions, Projects, Security, Insights, Settings.

Repository details: keehyunpark / practice (Private), master, practice / test.txt, Go to file, History.

Commit details: keehyunpark third commit, Latest commit bd2e47e 13 minutes ago, History.

File content: 3 lines (3 sloc) | 43 Bytes

Commit content:

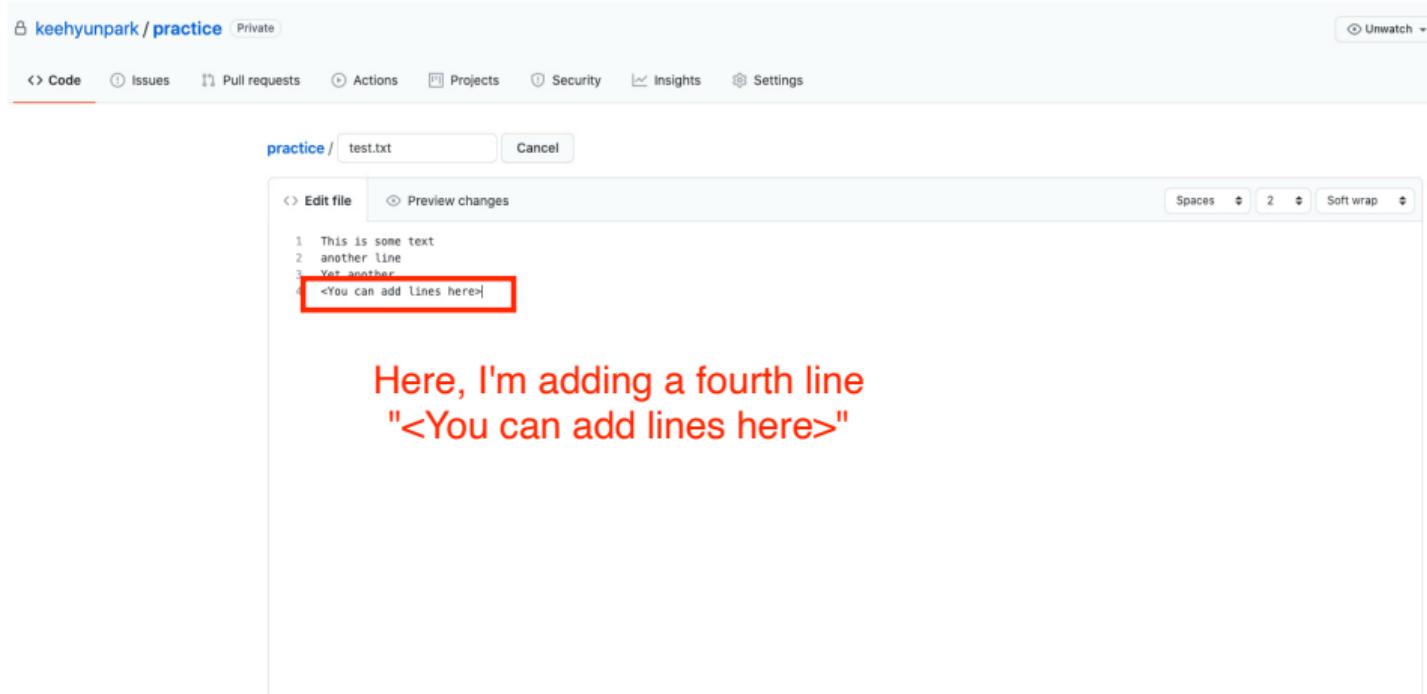
```
1 This is some text
2 another line
3 Yet another
```

Buttons: Raw, Blame, edit icon.

Annotations: Click!, You can "edit" the file directly on GitHub.

Page footer: © 2020 GitHub, Inc., Terms, Privacy, Security, Status, Help, Contact GitHub, Pricing, API, Training, Blog, About.

Local project folder ⇒ GitHub



The screenshot shows a GitHub repository interface for the user 'keehyunpark/practice'. The 'Code' tab is selected. A file named 'test.txt' is open, showing the following content:

```
1 This is some text
2 another line
3 Yet another
4 <You can add lines here>
```

A red box highlights the fourth line, "<You can add lines here>". Below the editor, a red text overlay reads:

Here, I'm adding a fourth line
"<You can add lines here>"

Local project folder ⇒ GitHub

Let's make another commit
and leave a tag "fourth commit"

Commit changes

fourth commit

Add an optional extended description...

keehyunpark22@gmail.com

Choose which email address to associate with this commit

⌘ Commit directly to the `master` branch.

⌘ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

Local project folder ⇒ GitHub

The screenshot shows a GitHub repository page for 'keehyunpark / practice'. The 'Code' tab is selected. A commit titled 'keehyunpark fourth commit' is highlighted with a red underline. The commit message contains four lines of text: '1 This is some text', '2 another line', '3 Yet another', and '4 <You can add lines here>'. Below the commit, a large red text overlay reads 'You can see the changes made'.

keehyunpark / practice Private

Code Issues Pull requests Actions Projects Security Insights Settings

master practice / test.txt Go to file ...

keehyunpark fourth commit Latest commit 4b3e13e now History

1 contributor

4 lines (4 sloc) | 68 Bytes

1 This is some text
2 another line
3 Yet another
4 <You can add lines here>

You can see the changes made

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Local project folder ⇒ GitHub

- We just made a "fourth commit" on GitHub repo. We can now "pull" this change to our local "Downloads/practice" folder.

```
git pull origin master
```

#meaning: Pull origin (GitHub repo) to master (local "practice")

Local project folder ← GitHub

- What if we want to copy someone's GitHub repository to a local folder?
- Let's try cloning <https://github.com/eandrewjones/gvpt-methods>

Local project folder ← GitHub

The screenshot shows a GitHub repository page for the user 'EandrewJones' named 'gvpt-methods'. The repository has 1 branch and 0 tags. The README.md file contains the following content:

```
keehyunpark Update README.md
8c1bd5e on Aug 17 4 commits

README.md
Update README.md
last month

README.md

UMD Government and Politics Methodology Workshop Series

2020-2021 Maintainers & Contributors:
• Evan A. Jones
• Xiaonan Wang
• Ted Ellsworth
• Kee Hyun Park
• Tiago Ventura
```

The repository has 2 unwatched, 1 starred, and 0 forks. It includes sections for About, Releases, Packages, and Contributors.

Goal: copy this folder to local Downloads folder

Local project folder ← GitHub

- In the Terminal/Shell,

```
cd ..
```

#meaning: move one ladder up, to "Downloads" folder.

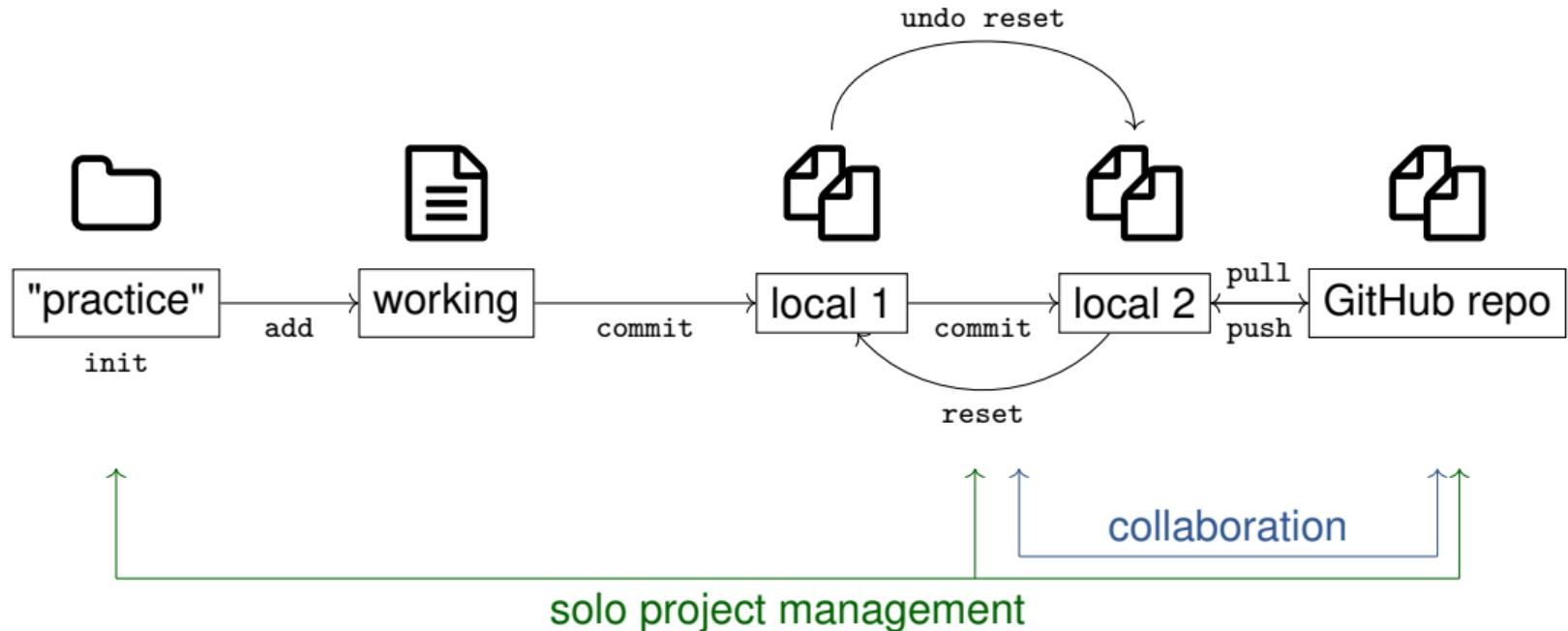
#Be careful! Should place a space between "cd" and ".."

```
git clone https://github.com/eandrewjones/gvpt-methods
```

#meaning: copy GitHub repo to local "gvpt-methods" folder.

- You'll notice that "Downloads/gvpt-methods" has been created

Summary



Concluding Remarks

In this talk:

- introduce Git and GitHub as a mode of workflow management
- show that both are good candidates for solo/collaborative work
- particularly useful for tracking detailed changes, managing past versions

But:

- Git and GitHub *may* work best for empiricists who use programming languages (R, Python, julia..) to deal with massive/various data types (XML,JSON..)
- Jargons, jargons... barrier to entry
- Other popular modes (e.g. MS Word + emails / Google Drive + @) are as much as effective

Recommend learning! You can always come back to the materials.

- Slides and other materials available at:

<https://github.com/EandrewJones/gvpt-methods>

- Follow-up questions:

khpark@umd.edu