

YUICHI OKUGAWA

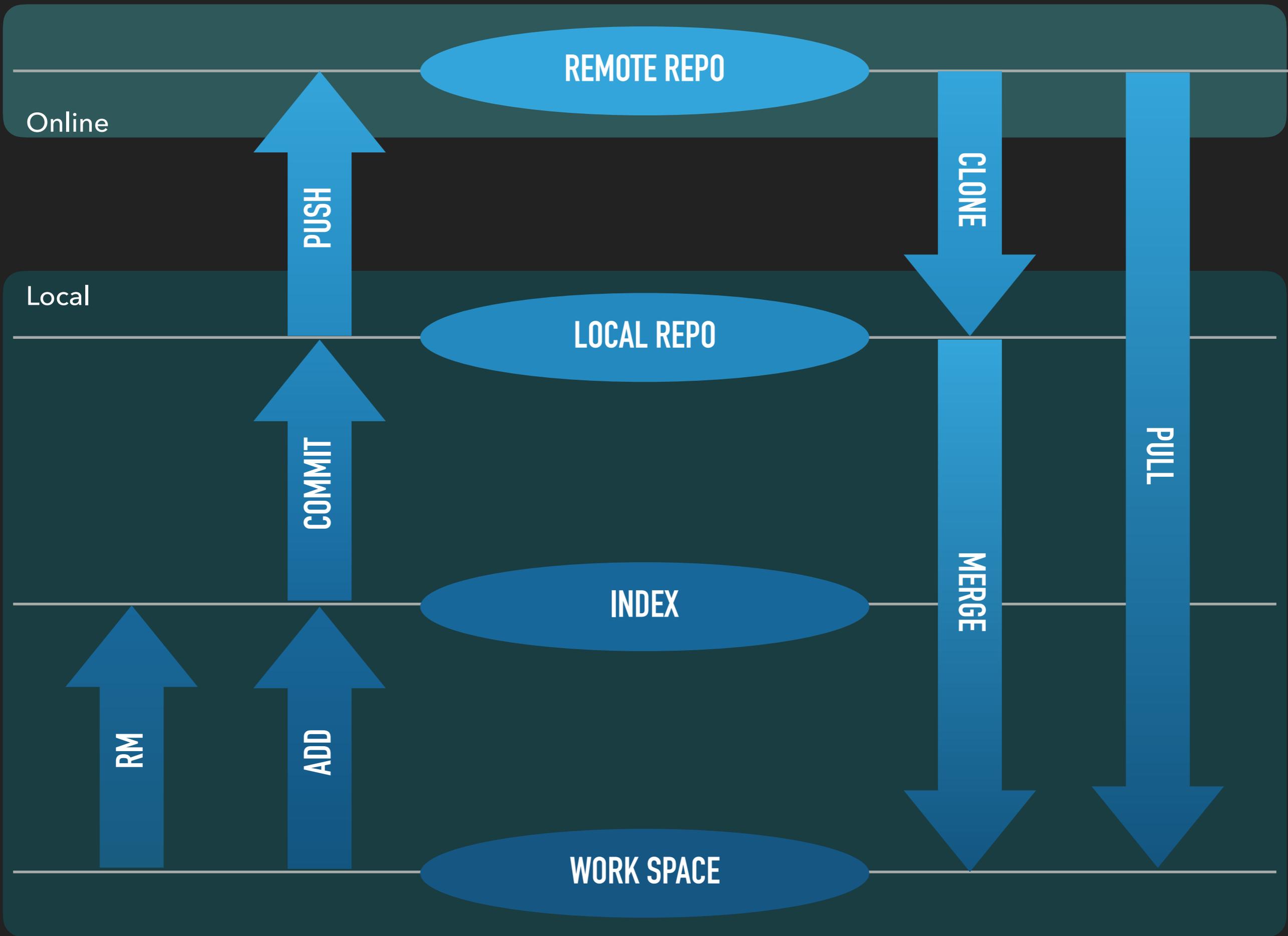
---

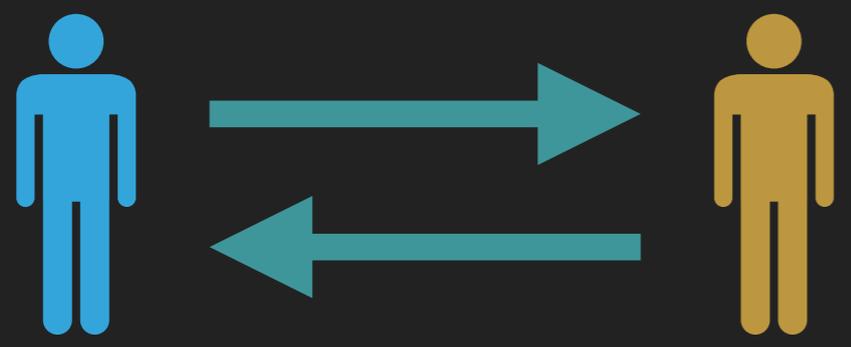
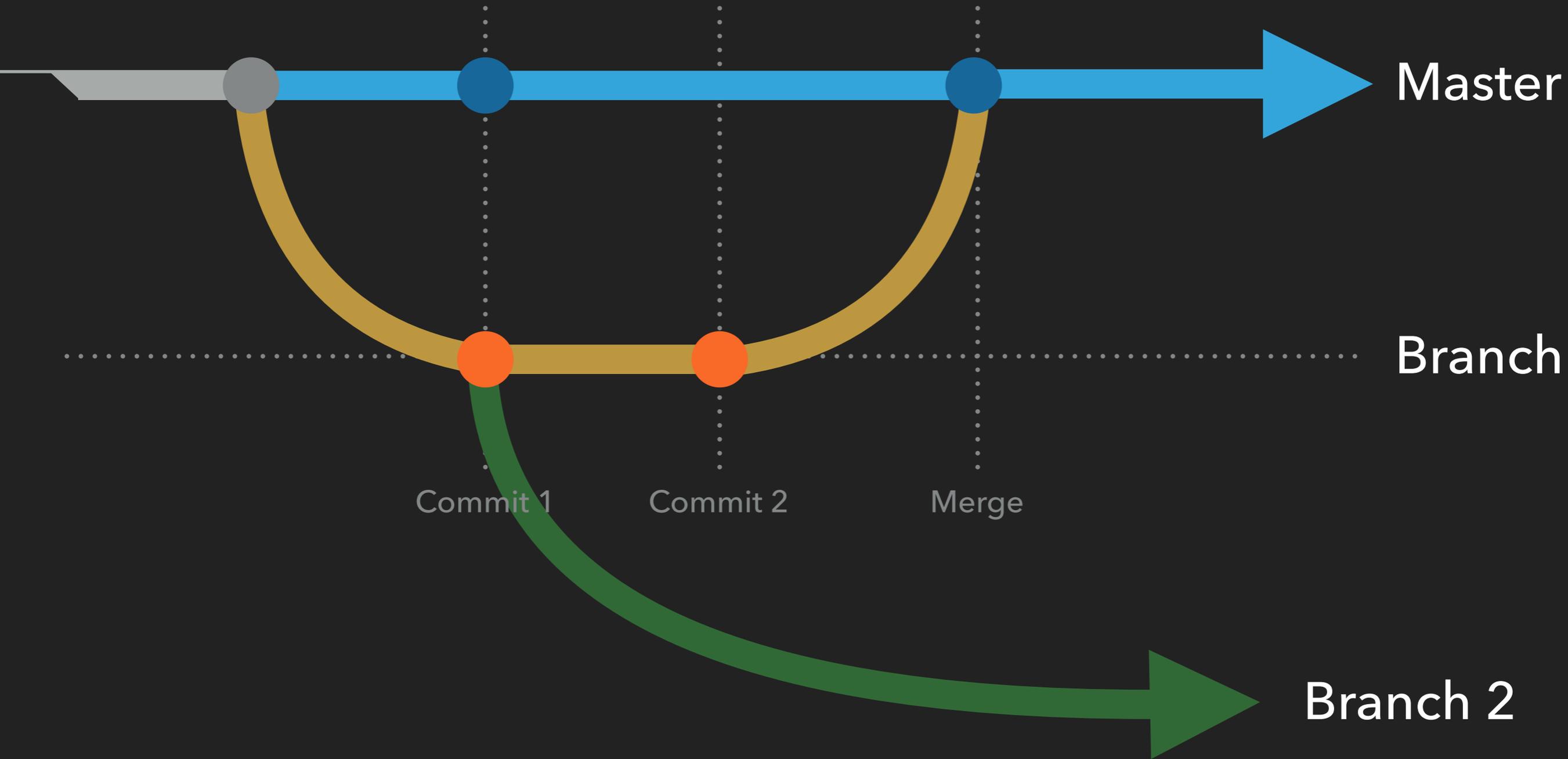
**GITHUB HANDS ON TUTORIAL**

# WHAT'S GITHUB?

- ▶ **GitHub** is a web-based hosting service for version control using Git.
- ▶ Over **31 million** users are using this tool to develop softwares.
- ▶ It can be easily accessed from command lines, making users to interact with repositories (repos) fast and easy.







# VOCABULARY

- ▶ **Repository** : A directory or storage space where your projects can live. Also known as "repo".
- ▶ **Version control** : A way to keep track of every modification to the code in a special kind of database.
- ▶ **Branch** : Division of software management so people can work on their own working space.
- ▶ **Commit** : A "snapshot" of a repository at that point in time, giving users a checkpoint to which they can reevaluate or restore their project to any previous state.

# HANDS ON SESSION

# GENERATE PUBLIC KEY

```
$ cd ~/.ssh
```

```
$ ssh-keygen -t rsa -C "your_email@example.com"
```

```
Generating public/private rsa key pair.
```

```
# Enter file in which to save the key (/Users/you/.ssh/id_rsa):  
github_id_rsa (optional)
```

```
Enter passphrase (empty for no passphrase): (optional)
```

```
# Enter same passphrase again: (optional)
```

```
$ ssh-add ~/.ssh/id_rsa
```

This will create ssh key to access the GitHub server.

For checking, enter following:

```
$ ssh -vT git@github.com
```

```
Hi yuichiok! You've successfully authenticated, but GitHub does not  
provide shell access.
```

# GENERATE PUBLIC KEY

Copy the key info from id\_rsa.pub file.

```
$ pbcopy < ~/.ssh/id_rsa.pub
```

The screenshot shows the GitHub profile page for user **Yuichi Okugawa** (username: `yuichiok`). The user is identified as an "Experimental High Energy Physics student" at "Tohoku University". The profile includes a bio edit button and a list of pinned repositories:

- TTBarAnalysis** (C++): Semi-leptonic TTBarAnalysis is presented here.
- QQbarAnalysis** (C++): Forked from QQbarAnalysis/qqbaranalysis
- VertexMacro** (C): Analysis macros regards to Vertex Charge Measurements
- SemiLeptonicAnalysis** (C): TTBarAnalysis with semi-leptonic channel
- MonoPhoton** (Makefile): Mono photon WIMPS analysis
- MC\_Data\_Analysis** (C++): data analysis for event counter

The user's profile statistics are: Overview (selected), Repositories 18, Stars 0, Followers 2, and Following 4. A user menu is open on the right, showing options like "Signed in as yuichiok", "Your profile", "Your repositories", "Your stars", "Your gists", "Help", "Settings" (circled in red), and "Sign out".

# GENERATE PUBLIC KEY

Copy the key info from id\_rsa.pub file.

```
$ pbcopy < ~/.ssh/id_rsa.pub
```

Personal settings

- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys**
- Security
- Sessions
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications

Developer settings

## SSH keys New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 SSH	<b>YuichiMac</b> 4c:eb:f8:52:9f:da:79:f3:cd:16:9a:e6:47:36:7e:f7 Added on Jun 7, 2016 Last used within the last 5 months — Read/write	Delete
 SSH	<b>KEKCC</b> 88:1b:0a:7c:27:f7:8b:4c:a2:0e:29:66:80:6d:c8:0e Added on Jul 26, 2018 Last used within the last week — Read/write	Delete
 SSH	<b>LALCC</b> e5:08:8e:7a:50:5b:64:42:6f:46:74:d6:79:03:55:5a Added on Aug 14, 2018 Last used within the last 3 months — Read/write	Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

## GPG keys New GPG key

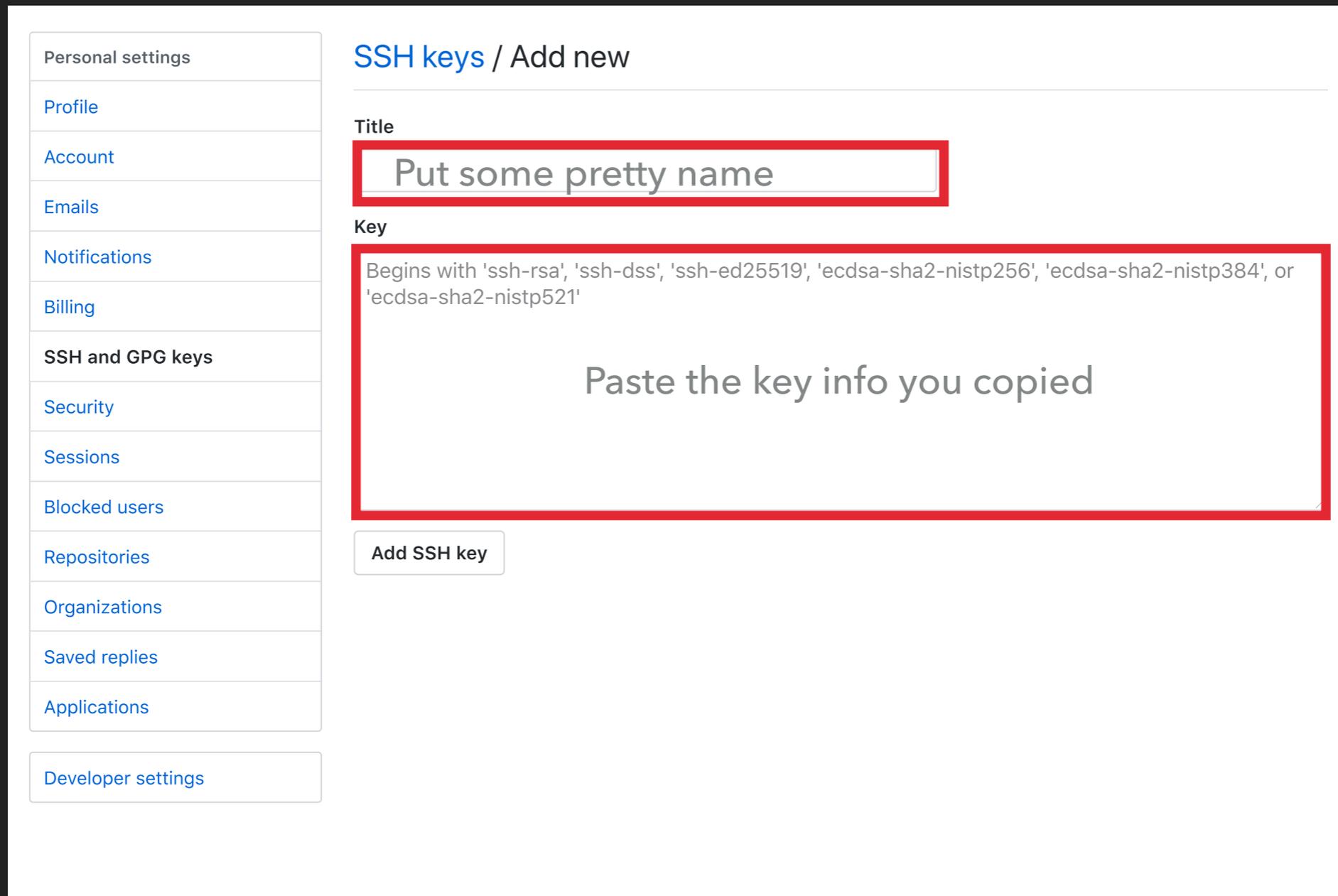
There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

# GENERATE PUBLIC KEY

Copy the key info from id\_rsa.pub file.

```
$ pbcopy < ~/.ssh/id_rsa.pub
```



The screenshot shows the GitHub 'SSH keys / Add new' page. On the left is a sidebar with navigation links: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys (highlighted), Security, Sessions, Blocked users, Repositories, Organizations, Saved replies, Applications, and Developer settings. The main content area has a title 'SSH keys / Add new' and two input fields. The 'Title' field contains the text 'Put some pretty name' and is highlighted with a red border. The 'Key' field contains a large text area with the instruction 'Paste the key info you copied' and is also highlighted with a red border. Below the key field is a button labeled 'Add SSH key'.

Personal settings

Profile

Account

Emails

Notifications

Billing

**SSH and GPG keys**

Security

Sessions

Blocked users

Repositories

Organizations

Saved replies

Applications

Developer settings

## SSH keys / Add new

Title

Put some pretty name

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Paste the key info you copied

Add SSH key

# CREATE A REPO



Search or jump to...



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



- New repository
- Import repository
- New gist
- New organization



**Yuichi Okugawa**

yuichiok

Experimental High Energy Physics student

Edit bio

Tohoku University

Organizations



**Overview** [Repositories 17](#) [Stars 0](#) [Followers 2](#) [Following 4](#)

Pinned repositories

Customize your pinned repositories

[TTBarAnalysis](#)

Semi-leptonic TTBarAnalysis is presented here.

C++

[QQbarAnalysis](#)

Forked from QQbarAnalysis/qqbaranalysis

C++

[VertexMacro](#)

Analysis macros regards to Vertex Charge Measurements

C

[SemiLeptonicAnalysis](#)

TTBarAnalysis with semi-leptonic channel

C

[MonoPhoton](#)

Mono photon WIMPS analysis

Makefile

[MC\\_Data\\_Analysis](#)

data analysis for event counter

C++

# CREATE A REPO

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

GitHub\_HATs

Choose any name you want

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

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

# CREATE A REPO

yuichiok / GitHub\_HATs

## Quick setup — if you've done this kind of thing before

 Set up in Desktop or    

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## ...or create a new repository on the command line

```
echo "# GitHub_HATs" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:yuichiok/GitHub_HATs.git
git push -u origin master
```



## ...or push an existing repository from the command line

```
git remote add origin git@github.com:yuichiok/GitHub_HATs.git
git push -u origin master
```



## ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

# CREATE A REPO



Search or jump to...



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



[yuichiok](#) / [GitHub\\_HATs](#)

[Unwatch](#) 1

[Star](#) 0

[Fork](#) 0

[Code](#)

[Issues](#) 0

[Pull requests](#) 0

[Projects](#) 0

[Wiki](#)

[Insights](#)

[Settings](#)

No description, website, or topics provided.

[Edit](#)

[Manage topics](#)

[1](#) commit

[1](#) branch

[0](#) releases

[0](#) contributors

Branch: [master](#)

[New pull request](#)

[Create new file](#)

[Upload files](#)

[Find file](#)

[Clone or download](#)

[Yuichi](#) first commit

Latest commit [0c166f7](#) an hour ago

[README.md](#)

first commit

an hour ago

[README.md](#)

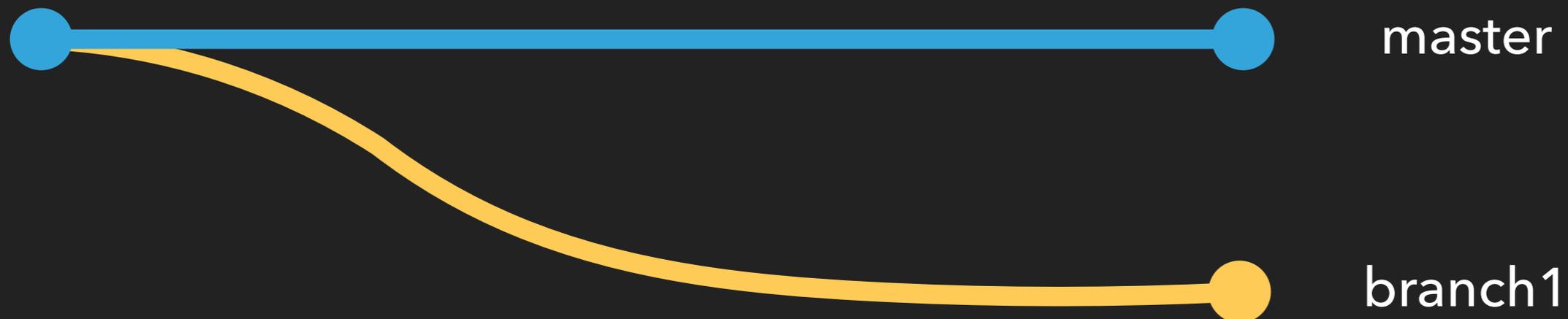


## GitHub\_HATs

# SUCCESS!

# CREATE BRANCH

```
$ git branch -v
* master 0c166f7 first commit
$ git checkout -b branch1
Switched to a new branch 'branch1'
$ git branch -v
* branch1 0c166f7 first commit
  master  0c166f7 first commit
```



# CREATE BRANCH

```
$ echo "Hello, World from branch1" >> branch1.txt  
$ git add branch1.txt  
$ git commit -m 'Second commit from branch1'  
$ git push origin branch1
```

Search or jump to... / Pull requests Issues Marketplace Explore

yuichiok / GitHub\_HATs Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Manage topics

2 commits 2 branches 0 releases 0 contributors

Your recently pushed branches:

branch1 (7 minutes ago) Compare & pull request

Branch: branch1 New pull request Create new file Upload files Find file Clone or download

This branch is 1 commit ahead of master. Pull request Compare

Yuichi Second commit from branch1 Latest commit 2f11f5d 8 minutes ago

README.md	first commit	2 hours ago
branch1.txt	Second commit from branch1	8 minutes ago

→

# MERGE BRANCH



Great! I checked **merge conflicts** with my branch and I see no issue. I will merge the branch.

master

branch1

## MERGE COMPLETE!



I have done my part of the work! I would like to **merge** to the master branch so please check my **pull request**.

# MERGE BRANCH

```
$ ls
```

```
README.md      branch1.txt
```

```
$ git checkout master
```

```
Switched to branch 'master'
```

```
Your branch is up-to-date with 'origin/master'.
```

```
$ ls
```

```
README.md
```

```
$ git merge branch1
```

```
Fast-forward
```

```
branch1.txt | 1 +
```

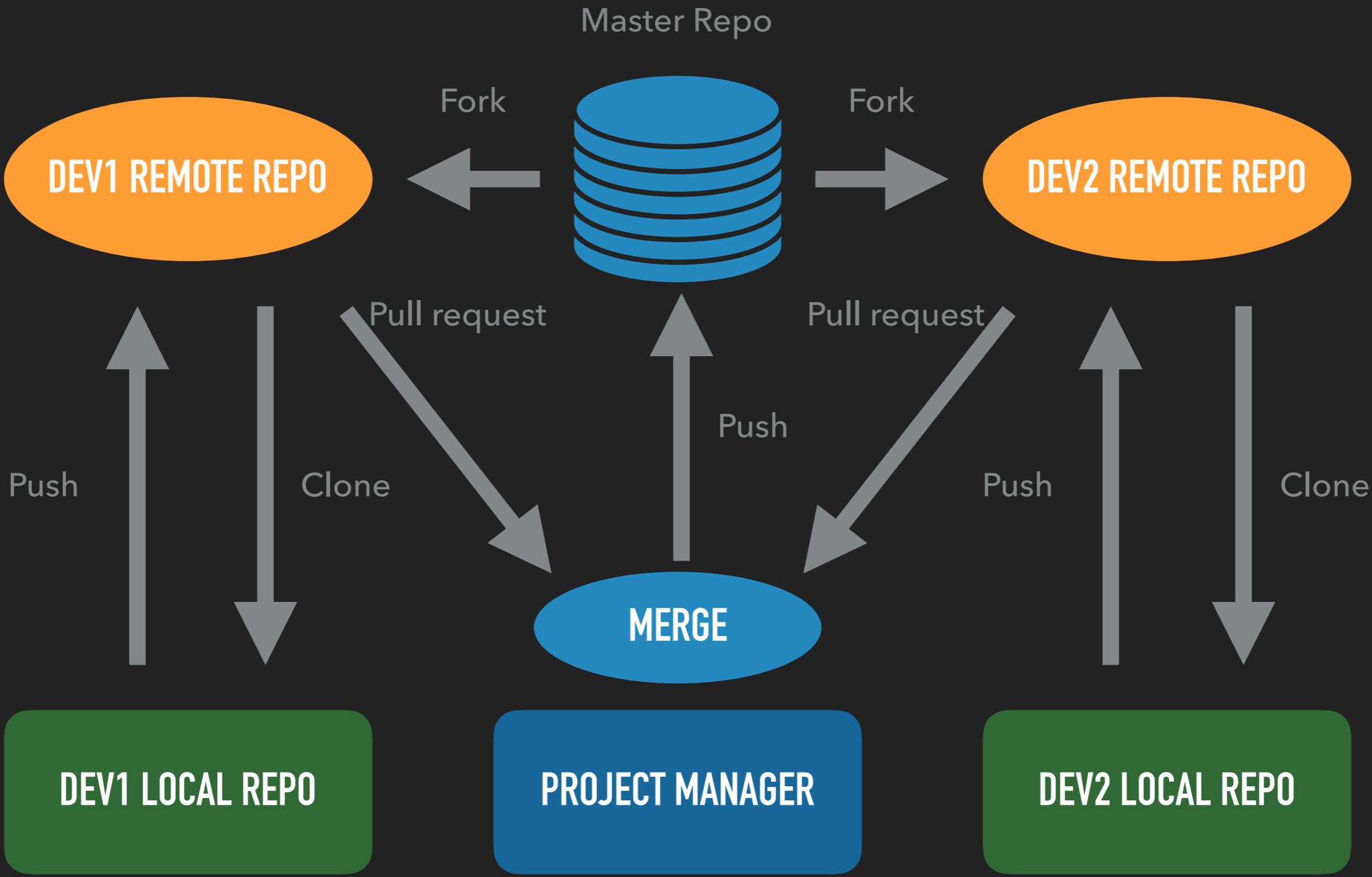
```
1 file changed, 1 insertion(+)
```

```
create mode 100644 branch1.txt
```

```
$ ls
```

```
README.md      branch1.txt
```

# FORK AND CLONE



# FORK AND CLONE

Go to [https://github.com/yuichiok/GitHub\\_HATs](https://github.com/yuichiok/GitHub_HATs)

The screenshot shows the GitHub interface for the repository 'yuichiok / GitHub\_HATs'. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the right side of the navigation bar, there are icons for notifications, a plus sign, and a grid icon. Below the navigation bar, the repository name 'yuichiok / GitHub\_HATs' is displayed. To the right of the repository name, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). The 'Fork' button is circled in red. Below the repository name, there are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main content area shows a message: 'No description, website, or topics provided.' with an 'Edit' button. Below this, there is a 'Manage topics' link. A summary bar shows '3 commits', '2 branches', '0 releases', and '0 contributors'. Below the summary bar, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table is displayed, showing the latest commit by 'Yuichi' adding 'Participant\_List.txt' 30 seconds ago. The table lists the following files and their commit messages:

File	Commit Message	Time
<a href="#">Participant_List.txt</a>	add Participant_List.txt	28 seconds ago
<a href="#">README.md</a>	first commit	14 hours ago
<a href="#">branch1.txt</a>	Second commit from branch1	13 hours ago

Below the commit history, there is a section for the 'README.md' file, which is currently empty. The repository name 'GitHub\_HATs' is displayed at the bottom of the page.

# EXERCISE

1. Clone repo from your fetched repo.
2. Edit Participant\_List.txt file in your local repo and enter your name. (next to any number from 1-10)
3. Add/commit/push the changes you made to your repo.
4. Make pull request. (press “New pull request” button)
5. Pull from the repo and check people’s name are on there.

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

"Persistence is very important. You should not give up unless you are forced to give up."  
– Elon Musk

# REVERT CHANGES

- ▶ Almost everything you do in GitHub is revertible as long as you commit your changes.
- ▶ Every commits has their own index and you can refer to them when reverting. Type `git reflog` to see them.
- ▶ In order to go back, type `git reset --hard <index>`.  
(If the commits were not pushed to the online repo yet)
- ▶ If the commits were already published, you can still go back by typing `git revert <index>`

# REVERT CHANGES

```
$ git reflog  Index
56fc757 (HEAD -> master, origin/master) HEAD@{0}: commit: reset
changes
e94dbf9 HEAD@{1}: pull origin master: Fast-forward
787f04b HEAD@{2}: commit: add Participant_List.txt
2f11f5d (origin/branch1, branch1) HEAD@{3}: merge branch1: Fast-forward
0c166f7 HEAD@{4}: checkout: moving from branch1 to master
2f11f5d (origin/branch1, branch1) HEAD@{5}: commit: Second commit from
branch1
0c166f7 HEAD@{6}: checkout: moving from master to branch1
0c166f7 HEAD@{7}: commit (initial): first commit
```

```
Yuichi:~/working/github_hats$ git log --oneline
```

```
56fc757 (HEAD -> master, origin/master) reset changes
e94dbf9 Merge pull request #1 from ahmedmustahid/master
1290541 from ahmed test
787f04b add Participant_List.txt
2f11f5d (origin/branch1, branch1) commit: Second commit from branch1
0c166f7 checkout: moving from master to branch1
0c166f7 HEAD@{6}: checkout: moving from master to branch1
0c166f7 HEAD@{7}: commit (initial): first commit
```

```
$ echo "2 number nines" >> order.txt
$ echo "a number 9 large" >> order.txt
$ git add order.txt
$ git commit -m 'an order added!'
```

```
$ echo "a number 6 with extra dip" >> order.txt
$ echo "a number 7" >> order.txt
$ git add order.txt
$ git commit -m 'more orders added!'
```

```
$ echo "two number 45s" >> order.txt
$ echo "one with cheese" >> order.txt
$ echo "a large soda" >> order.txt
$ git add order.txt
$ git commit -m 'more and more orders added!'
```



```
$ git reflog
```

```
a77d0b3 (HEAD -> master) HEAD@{0}: commit: more and more orders added!
```

```
3e6fb54 HEAD@{1}: commit: more orders added!
```

```
e5c811b HEAD@{2}: commit: an order added!
```

```
$ cat order.txt
```

```
2 number nines
```

```
a number 9 large
```

```
a number 6 with extra dip
```

```
a number 7
```

```
two number 45s
```

```
one with cheese
```

```
a large soda
```



Maybe I ordered too much. Only 2 number nines, a number 9 large, a number 6 with extra dip and a number 7 would do.

```
$ git reset --hard 3e6fb54
HEAD is now at 3e6fb54 more orders added!
$ cat order.txt
2 number nines
a number 9 large
a number 6 with extra dip
a number 7
```

