📼 teaching computing with Git & GitHub

🔗 bit.ly/teach-r-online-mats

dr. mine çetinkaya-rundel
dr. colin rundel

Activity

while we wait to get started...

- Go to bit.ly/gh-username
- Enter your GitHub username

you...

- are familiar with R
- are familiar with Git and GitHub
- are interested in teaching version control
- are interested in using GitHub as your learning management system
- might be interested in automation tools offered by GitHub for auto feedback

why

git & github

Goals for version control with Git & GitHub

- Centralize the distribution and collection of all student assignments
- Enable students to work collaboratively
- Force students to use (learn) Git & GitHub
  - Version control is a best practice for reproducible research
  - Widely used in industry
  - Publish / share work

github

as a student

Aside - Git credentials

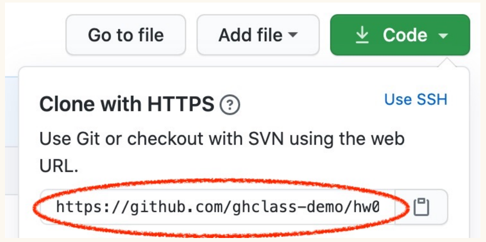
Using *https* for authentication is highly recommended

- Students will have to enter their username and password each time they *clone* or *push*
- Credentials can be cached, see Happy Git and GitHub for the useR
- Chapter 10
- Alternatively, see the credentials package and its vignette.

Your turn!

We recommend one person in each group share their screen and everyone work together to work through the document.

- Check your email and accept the invitation
- Obtain the *https* Git url from the GitHub repository
- Open RStudio Cloud and start a new project with this url
- Work through Task 0 in the README



Use Git or checkout with SVN using the web URL.

```
https://github.com/ghclass-demo/hw0
```

If you did not receive an invite you can make your own copy of the repo using the *Use this template* button here: https://github.com/rundel/hw1

15:00

github

as an instructor

Basic Structure

On Github,

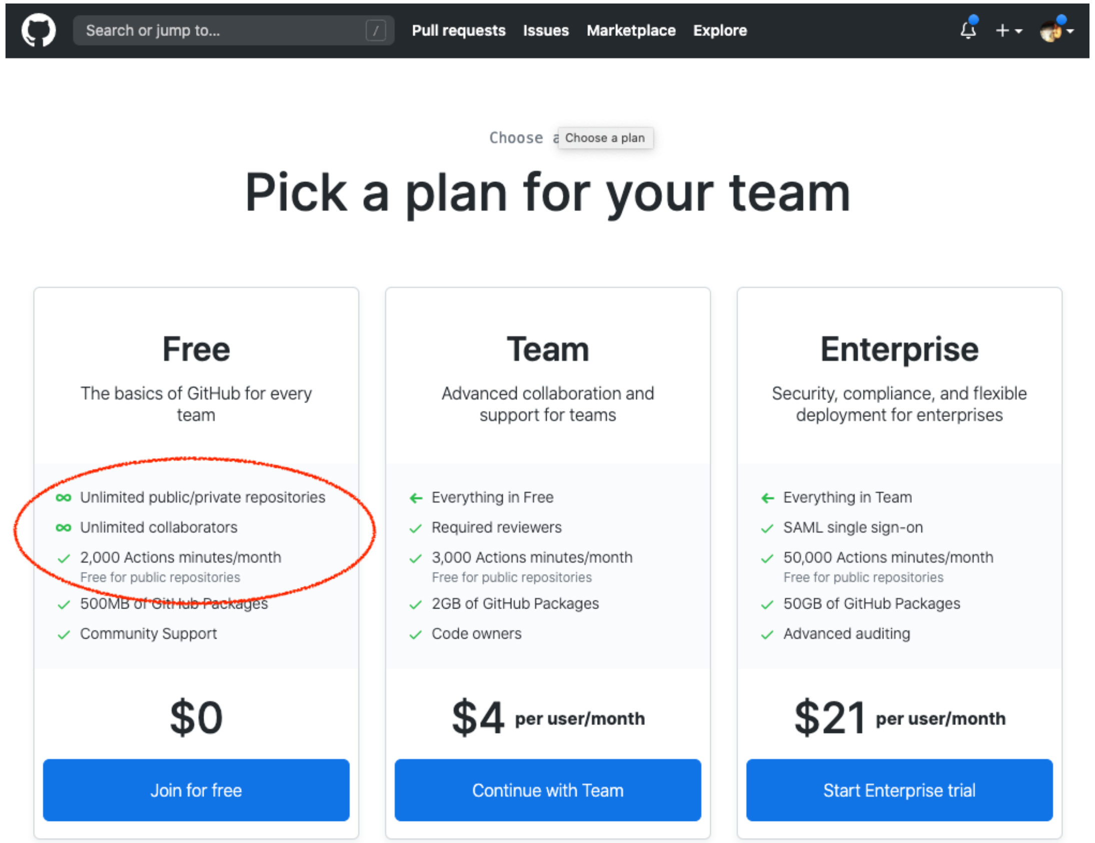
- 1 Organization / class
- 1 repo / (student or team) / assignment
- Student and team repos are all private by default
- Students are added as members
- Tutors / TAs are added as owners (admins)

Basic Workflows

1. Create organization
2. Invite students
3. Create assignment(s)
4. Collect and grade assignments(s)

Create course organization

https://github.com/organizations/new



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

Choose a Choose a plan

Pick a plan for your team

Free

The basics of GitHub for every team

- Unlimited public/private repositories
- Unlimited collaborators
- 2,000 Actions minutes/month
  Free for public repositories
- 500MB of GitHub Packages
- Community Support

$0

Join for free

Team

Advanced collaboration and support for teams

- Everything in Free
- Required reviewers
- 3,000 Actions minutes/month
  Free for public repositories
- 2GB of GitHub Packages
- Code owners

$4 per user/month

Continue with Team

Enterprise

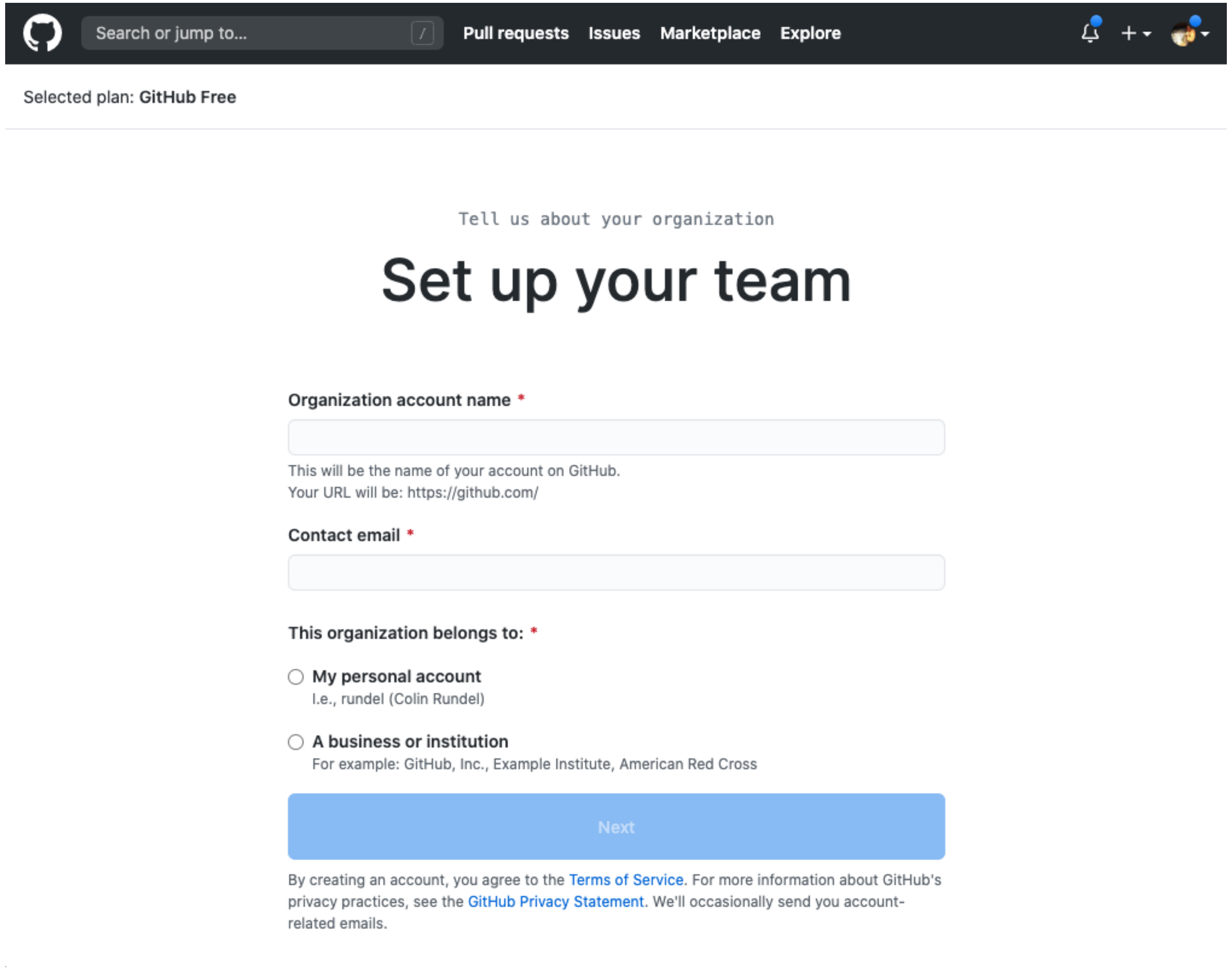
Security, compliance, and flexible deployment for enterprises

- Everything in Team
- SAML single sign-on
- 50,000 Actions minutes/month
  Free for public repositories
- 50GB of GitHub Packages
- Advanced auditing

$21 per user/month

Start Enterprise trial

Create course organization



Selected plan: GitHub Free

Tell us about your organization

Set up your team

Organization account name *

This will be the name of your account on GitHub.
Your URL will be: https://github.com/

Contact email *

This organization belongs to: *

- My personal account
  I.e., rundel (Colin Rundel)
- A business or institution
  For example: GitHub, Inc., Example Institute, American Red Cross

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.

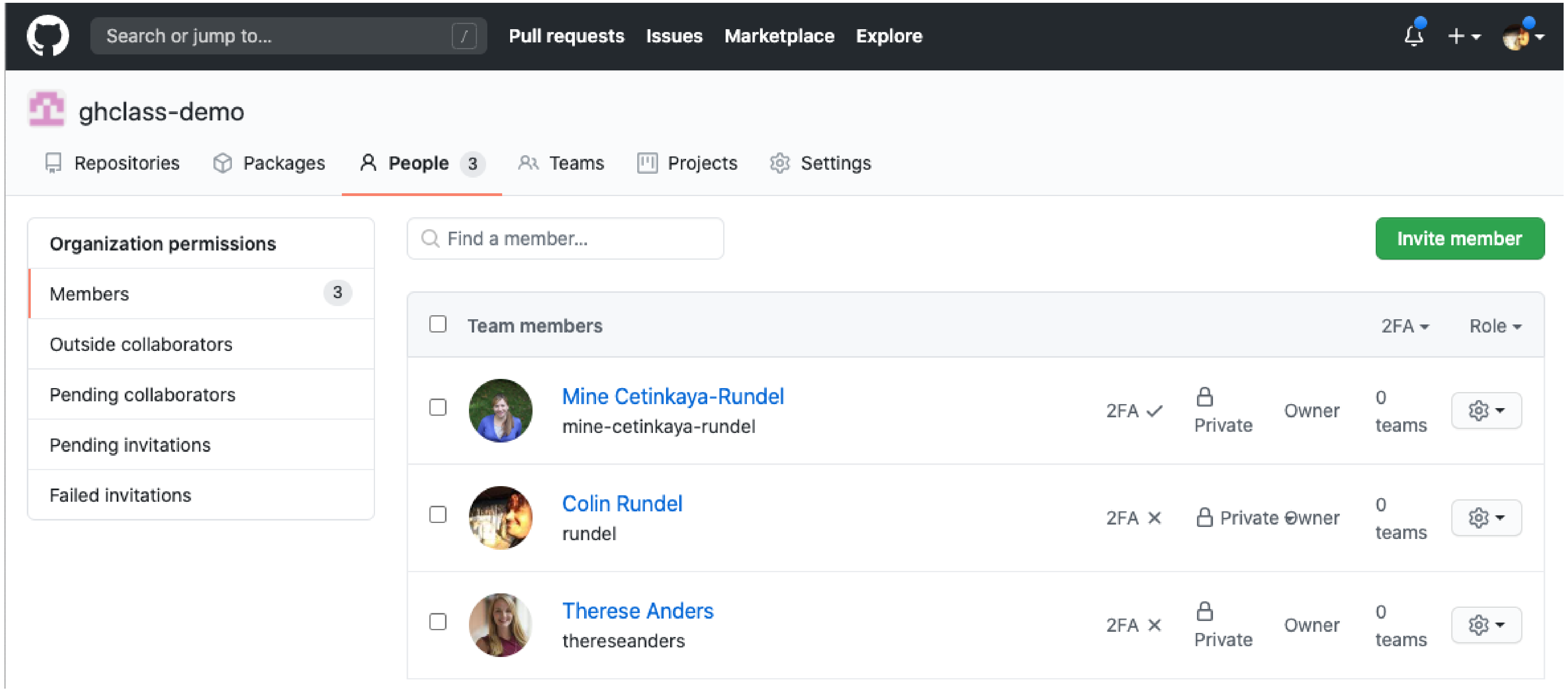
Education discount

While no longer required, GitHub offers a number of education benefits which you can register for here: https://education.github.com/benefits.

Of particular note are:

- Free GitHub swag here
    - https://education.github.com/toolbox/offers#github_swag
- Free Team plans for academic organizations and Pro plan for educators
    - https://education.github.com/toolbox/offers#github

Org Setup



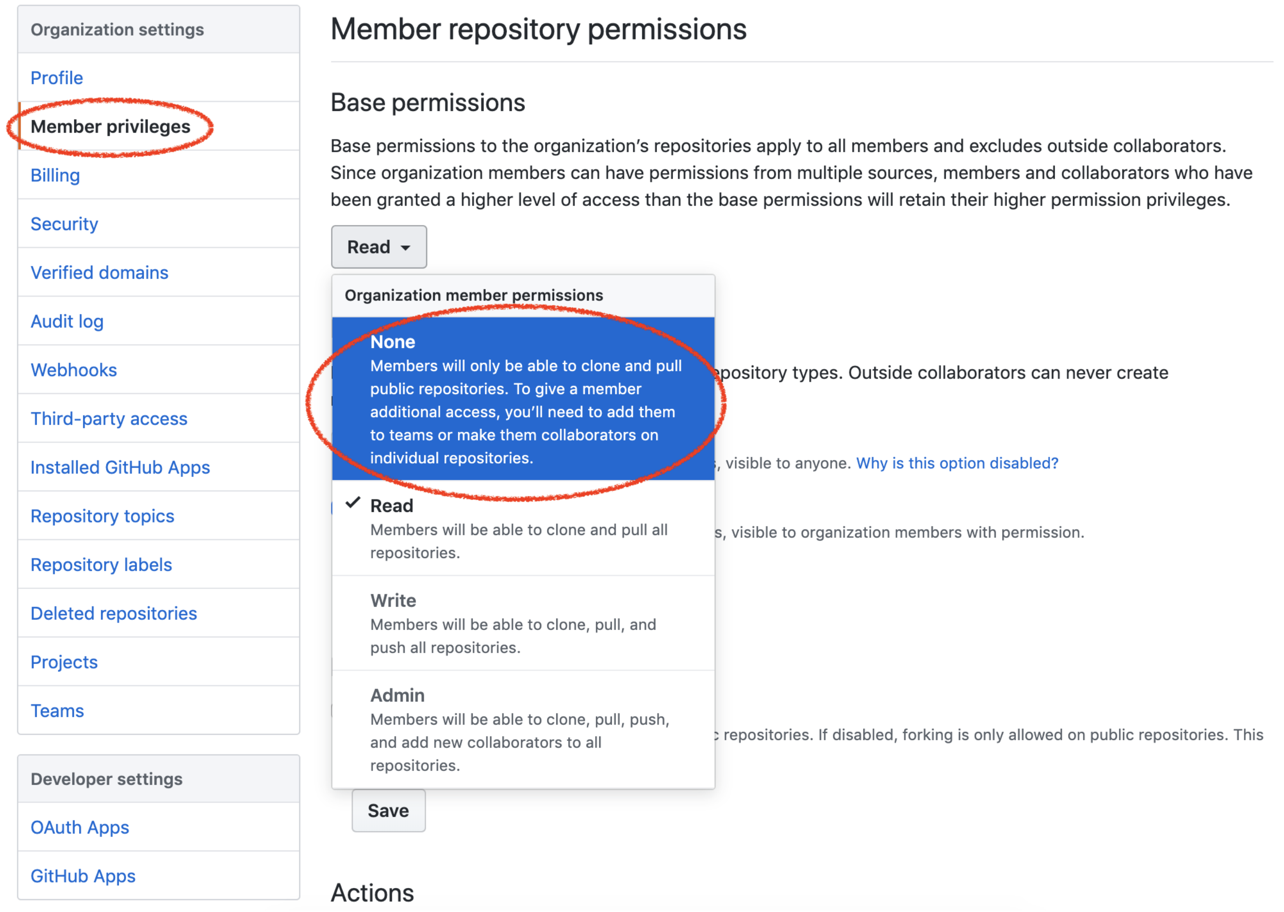
ghclass-demo

Repositories · Packages · People 3 · Teams · Projects · Settings

Organization permissions	
Members	3
Outside collaborators	
Pending collaborators	
Pending invitations	
Failed invitations	

Team members		2FA		Role	
Mine Cetinkaya-Rundel	mine-cetinkaya-rundel	2FA ✓	Private	Owner	0 teams
Colin Rundel	rundel	2FA ✕	Private	Owner	0 teams
Therese Anders	thereseanders	2FA ✕	Private	Owner	0 teams

Member Privileges



Organization settings

- Profile
- Member privileges
- Billing
- Security
- Verified domains
- Audit log
- Webhooks
- Third-party access
- Installed GitHub Apps
- Repository topics
- Repository labels
- Deleted repositories
- Projects
- Teams

Developer settings

- OAuth Apps
- GitHub Apps

Member repository permissions

Base permissions

Base permissions to the organization's repositories apply to all members and excludes outside collaborators. Since organization members can have permissions from multiple sources, members and collaborators who have been granted a higher level of access than the base permissions will retain their higher permission privileges.

Read

Organization member permissions

None
Members will only be able to clone and pull public repositories. To give a member additional access, you'll need to add them to teams or make them collaborators on individual repositories.

✓ **Read**
Members will be able to clone and pull all repositories.

Write
Members will be able to clone, pull, and push all repositories.

Admin
Members will be able to clone, pull, push, and add new collaborators to all repositories.

epository types. Outside collaborators can never create

, visible to anyone. Why is this option disabled?

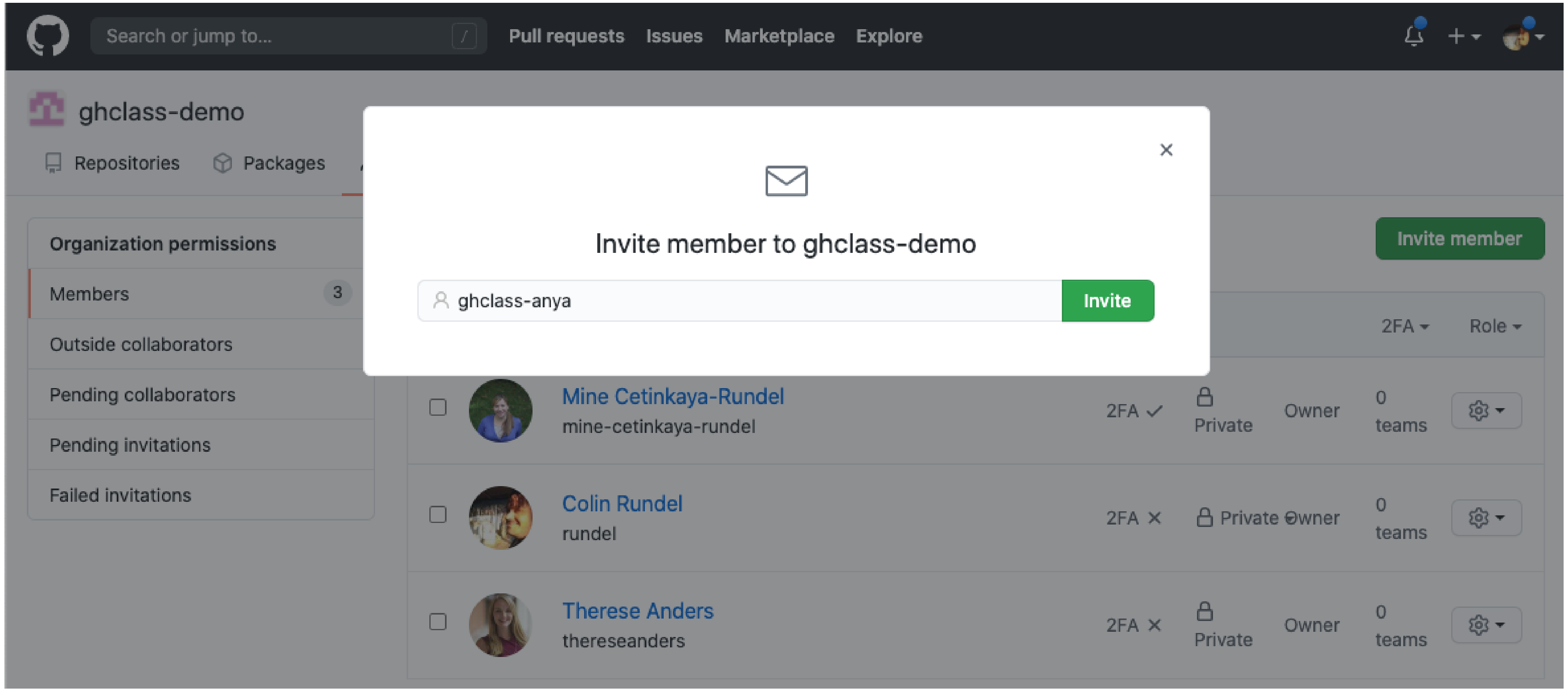
s, visible to organization members with permission.

repositories. If disabled, forking is only allowed on public repositories. This

Save

Actions

Invite students



Invite member to ghclass-demo

ghclass-anya | Invite

Mine Cetinkaya-Rundel mine-cetinkaya-rundel	2FA ✓	Private	Owner	0 teams
Colin Rundel rundel	2FA ✕	Private	Owner	0 teams
Therese Anders thereseanders	2FA ✕	Private	Owner	0 teams

x 150 students ...

at this scale, doing anything with the GitHub UI starts to get quite tedious…

📦 ghclass

📦 ghclass

- Design to automate interactions with GitHub (via its API) for class management
- The package is ~4 years old and still under active development
- Detailed introduction and documentation available on the package website: http://rundel.github.io/ghclass
- The package is not on CRAN (but will be imminently), for now it can be installed from GitHub using:

```r
devtools::install_github("rundel/ghclass")
library(ghclass)
```

📦 design

Some design principals behind the package:

1. All functions are prefixed to indicate what they operate on (e.g. `org`, `repo`, `team`, `local_repo`, etc.)

2. Most functions are vectorized over their parameters, allowing related operations to be grouped

3. Most actions are non-destructive or backed by Git, the handful of dangerous operations will warn you

Aside - GitHub tokens

`ghclass` uses the GitHub API to interact with your organization and repos - the API verifies your identity using a personal access token which must be created and saved in such a way that `ghclass` can find and use it.

- Create a token at github.com/settings/tokens
- Once created, assign it to the `GITHUB_TOKEN` as an environmental variable in R by,
    - Run `usethis::edit_r_environ()`
    - Add `GITHUB_PAT="alphanumeric string of your GitHub token"` to the opened `.Renviron` file.
    - Save, close, restart R for changes to take effect

Checking tokens

If the token is found and works correctly the following code should run without error

```
github_test_token()
```

```
## ✓ Your GitHub PAT authenticated correctly.
```

If instead the token is invalid or not found, you will see something like the following

```
github_test_token("BAD_TOKEN")
```

```
## x Your GitHub PAT failed to authenticate.
## └─GitHub API error (401): 401 Unauthorized
##    ├─ API message: Bad credentials
##    └─ API docs: https://developer.github.com/v3
```

Invite students

- Collect student account names (and an email or other identifier)

```r
students = c("ghclass-anya", "ghclass-bruno", "ghclass-celine", "ghclass-diego")
org_invite(org = "ghclass-demo", user = students)
```

```
## ✓ Invited user 'ghclass-anya' to org 'ghclass-demo'.
## ✓ Invited user 'ghclass-bruno' to org 'ghclass-demo'.
## ✓ Invited user 'ghclass-celine' to org 'ghclass-demo'.
## ✓ Invited user 'ghclass-diego' to org 'ghclass-demo'.
```

Rate limits

From GitHub's API docs,

> To prevent abuse, an authenticated user is limited to 50 organization invitations per 24 hour period. If the organization is more than one month old or on a paid plan, the limit is 500 invitations per 24 hour period.

Applying the education discount to an org => paid plan

Check member status

Who is already in?

```r
org_members(org = "ghclass-demo")
```

```
## [1] "mine-cetinkaya-rundel" "rundel" "thereseanders"
```

Who has not accepted their invitation?

```r
org_pending(org = "ghclass-demo")
```

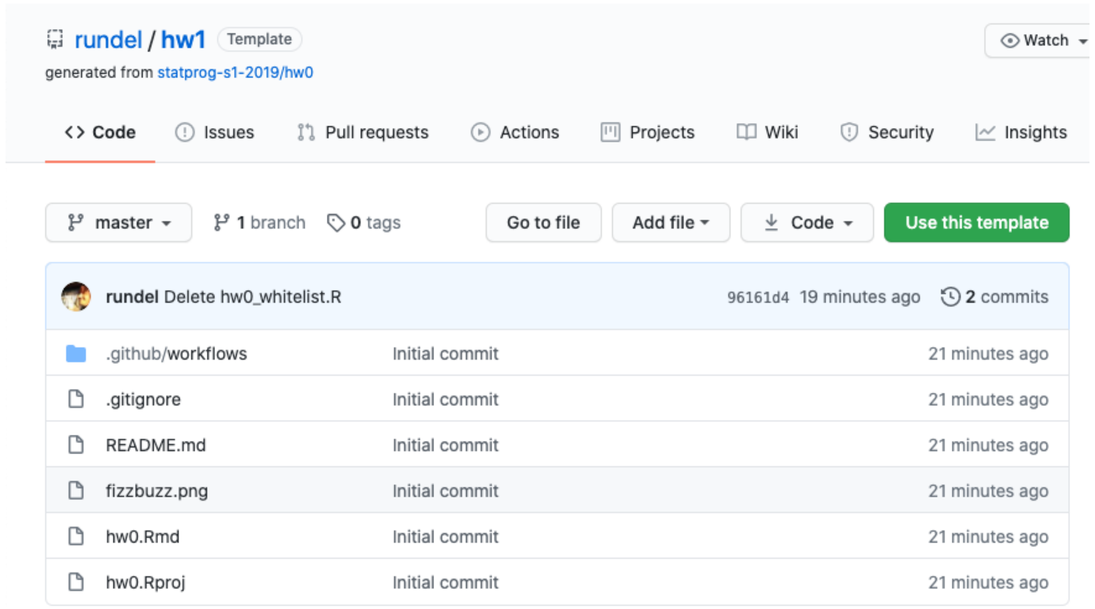
```
## [1] "ghclass-anya"   "ghclass-bruno"  "ghclass-diego"  "ghclass-celine"
```

Creating assignments

Starter repo

All assignments are just repositories on GitHub

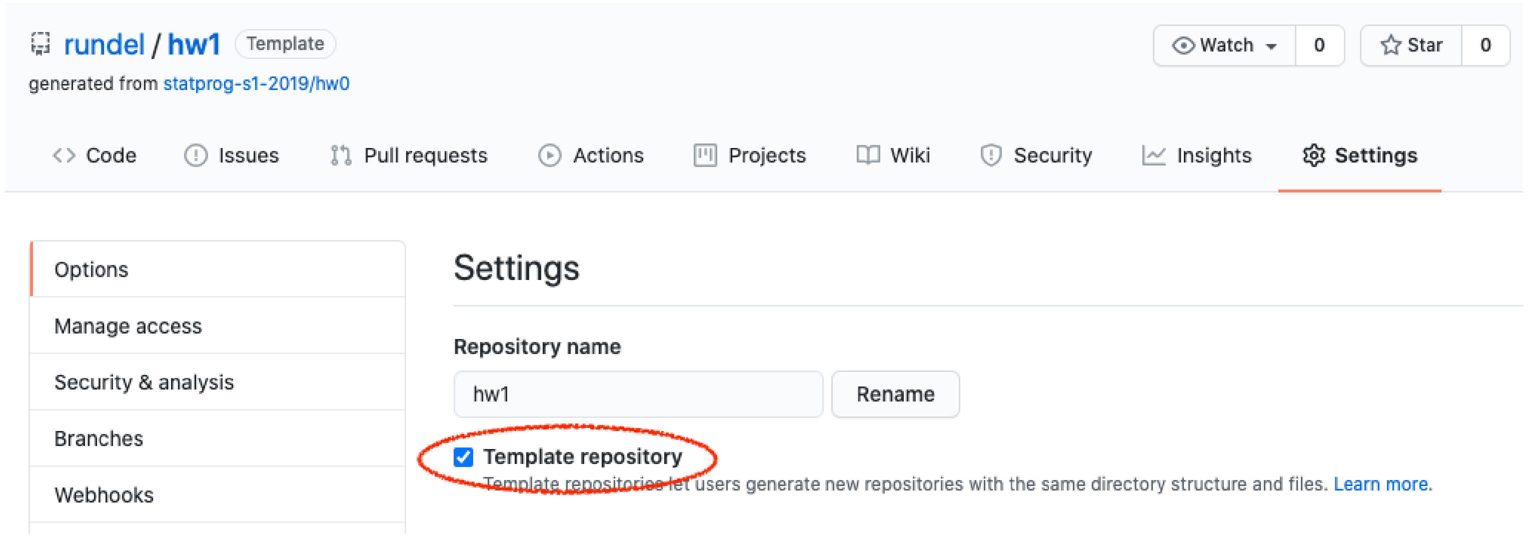
- each is made up of a collection of files necessary for that assignment (e.g. README, templated Rmd, Rproj file, etc.)
- repos can be public or private and belong to any org



rundel Delete hw0_whitelist.R 96161d4 19 minutes ago 2 commits

.github/workflows	Initial commit	21 minutes ago
.gitignore	Initial commit	21 minutes ago
README.md	Initial commit	21 minutes ago
fizzbuzz.png	Initial commit	21 minutes ago
hw0.Rmd	Initial commit	21 minutes ago
hw0.Rproj	Initial commit	21 minutes ago

Template Repos


rundel / hw1 Template

generated from statprog-s1-2019/hw0

Settings

Repository name: hw1

[✓] Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more.

```r
repo_set_template("rundel/hw1")
```

```
## ✓ Changed the template status of repo 'rundel/hw1' to TRUE.
```

```r
repo_is_template("rundel/hw1")
```

```
TRUE
```

Create assignments

```
org_create_assignment(
  org = "ghclass-demo",
  repo = paste0("hw01-", students),
  user = students,
  source_repo = "rundel/hw1"
)
```

```
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-anya'.
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-bruno'.
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-celine'.
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-diego'.
## ✓ Added user 'ghclass-anya' to repo 'ghclass-demo/hw01-ghclass-anya'.
## ✓ Added user 'ghclass-bruno' to repo 'ghclass-demo/hw01-ghclass-bruno'.
## ✓ Added user 'ghclass-celine' to repo 'ghclass-demo/hw01-ghclass-celine'.
## ✓ Added user 'ghclass-diego' to repo 'ghclass-demo/hw01-ghclass-diego'.
```

Create team assignments

```r
students = c("ghclass-anya", "ghclass-bruno", "ghclass-celine", "ghclass-diego")
teams = c("team01", "team01", "team02", "team02")

org_create_assignment(
  org = "ghclass-demo",
  repo = paste0("hw01-", teams),
  team = teams,
  user = students,
  source_repo = "rundel/hw1"
)
```

```
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-anya'.
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-bruno'.
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-celine'.
## ✓ Mirrored repo 'rundel/hw1' to repo 'ghclass-demo/hw01-ghclass-diego'.
## ✓ Added user 'ghclass-anya' to repo 'ghclass-demo/hw01-ghclass-anya'.
## ✓ Added user 'ghclass-bruno' to repo 'ghclass-demo/hw01-ghclass-bruno'.
## ✓ Added user 'ghclass-celine' to repo 'ghclass-demo/hw01-ghclass-celine'.
## ✓ Added user 'ghclass-diego' to repo 'ghclass-demo/hw01-ghclass-diego'.
```

Fixing mistakes

```r
repo_modify_file(
  repo = org_repos("ghclass-demo", filter = "hw01-"),
  path = "README.md",
  pattern = "Due 20/00/00 by 5:00 pm",
  content = "Due 2020/07/17 by 5:00 pm",
  method = "replace"
)
```

```
## ✓ Modified file 'ghclass-demo/hw01-ghclass-anya/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-ghclass-bruno/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-ghclass-celine/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-ghclass-diego/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-team01/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-team02/README.md'.
```

```r
repo_get_readme("ghclass-demo/hw01-team01", include_details = FALSE) %>%
  substr(1, 80) %>%
  cat()
```

```
##
## Statistical Programming - Homework 1
## -------------
##
## Due 2020/07/17 by 5:00 pm.
```

These changes are tracked by Git - to get them students will need to pull.

Collecting and Grading

Repo details

```r
org_repos("ghclass-demo")
```

```
## [1] "ghclass-demo/hw01-ghclass-anya"   "ghclass-demo/hw01-ghclass-bruno"
## [3] "ghclass-demo/hw01-ghclass-celine" "ghclass-demo/hw01-ghclass-diego"
## [5] "ghclass-demo/hw01-team01"         "ghclass-demo/hw01-team02"
```

```r
org_repos("ghclass-demo", filter = "hw01-team")
```

```
## [1] "ghclass-demo/hw01-team01" "ghclass-demo/hw01-team02"
```

```r
org_repo_stats("ghclass-demo")
```

```
## # A tibble: 6 x 6
##   repo                              private commits last_update         open_issues closed_issues
##   <chr>                             <lgl>     <int> <dttm>                    <int>         <int>
## 1 ghclass-demo/hw01-team02          TRUE          2 2020-07-17 08:42:50           0             0
## 2 ghclass-demo/hw01-team01          TRUE          2 2020-07-17 08:42:48           0             0
## 3 ghclass-demo/hw01-ghclass-diego   TRUE          2 2020-07-17 08:42:47           0             0
## 4 ghclass-demo/hw01-ghclass-anya    TRUE          2 2020-07-17 08:42:41           0             0
## 5 ghclass-demo/hw01-ghclass-bruno   TRUE          2 2020-07-17 08:42:43           0             0
## 6 ghclass-demo/hw01-ghclass-celine  TRUE          2 2020-07-17 08:42:45           0             0
```

Collecting

```r
local_repo_clone(
  repo = org_repos("ghclass-demo", filter = "hw01-team"),
  local_path = "hw1/"
)
```

```
✓ Cloned 'ghclass-demo/hw01-team01'.
✓ Cloned 'ghclass-demo/hw01-team02'.
```

```r
fs::dir_tree("hw1/", recurse = TRUE)
```

```
hw1/
├── hw01-team01
│   ├── README.md
│   ├── hw1.Rmd
│   └── hw1.Rproj
└── hw01-team02
    ├── README.md
    ├── hw1.Rmd
    └── hw1.Rproj
```

Options for giving feedback on GitHub

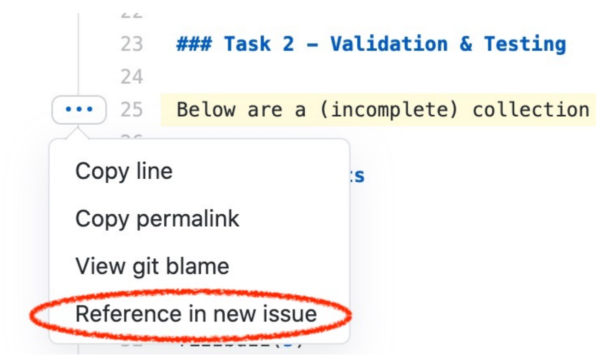
- Use the GitHub UI to review and add issues to each repo
- Use the `issue_create()` function to post issues to all repos at once
- Create pull requests with explicit revisions to student code
- Clone repos locally and add feedback in a file, push back to GitHub

More on giving feedback in issues

- Instructors (and TAs) can view all repositories within the course organization.
- Builtin tools for referencing specific commits, lines of code, etc.
- `@` mention students so that they are notified when an issue is opened.
- You may want to consider keeping grades / marks out of issues.

Your turn!

- Pick one person from the team to be the "instructor" and share their screen.
- Go to https://github.com/ghclass-demo/hw01-everyone (public repo)
- Go to the issues tab, open a new issue, and provide mock feedback. Tag someone from your team by using the `@` sign in front of their GitHub.
- Go to `hw1.Rmd`, pick a line of code, click on the `...` next to the numbers, click on *Reference in new issue*, and add a comment on the issue that links to this line of code.



```
### Task 2 - Validation & Testing

Below are a (incomplete) collection
```

10:00

Peer review

- Once an assignment is completed you can let other students/teams into a repository and they can provide peer review.
- Peer review is an incredibly effective learning experience for both the reviewers and the reviewees, however it does require coordination and being able to carve out sufficient time in the course schedule.
- Tip: Do not solely count on peer review for feedback as some reviewers might be less diligent than others. Teams reviewing teams, as opposed to individual reviewing individuals, might address this issue partially.
- Functionality for coordinating this has been implemented in ghclass, and will be available in the next release. Available in the `peer_review` branch for the adventurous.

Automated feedback

```r
action_workflows("ghclass-demo/hw01-team01")
```

```
## # A tibble: 1 x 4
##   name       path                       state  badge_url
##   <chr>      <chr>                      <chr>  <chr>
## 1 check_knit .github/workflows/knit.y… active https://github.com/ghclass-demo/hw01-team01/…
```

```r
action_add_badge(
  repo = org_repos("ghclass-demo", "hw01-")
)
```

```
## ✓ Modified file 'ghclass-demo/hw01-ghclass-anya/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-ghclass-bruno/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-ghclass-celine/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-ghclass-diego/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-team01/README.md'.
## ✓ Modified file 'ghclass-demo/hw01-team02/README.md'.
```

closing thoughts

Git + GitHub lessons learned

- If you plan on using Git in class, start on day one, don't wait until the "right time"
- First assignment should be individual, not team based to avoid merge conflicts
- Remind students to remember to pull before starting work
- You will likely need to do shell intervention at some point - make it a teachable momentc and remember, there is a terminal pane in RStudio
- Remind students on that future projects should go on GitHub with PI approval

Q: What about data protection regulations (FERPA, GDPR)?

- Consider data privacy rules of institution / country (e.g. you may need to enter a data protection agreement for GDPR compliance)

- Make everything private by default (ghclass opts for this)

    - Private repos
    - Hidden team and org memberships
    - Disallow forking of private repos

Q: What about GitHub Classroom?

This is education tool created by GitHub manage repository sharing and collection.

- It is great and very usable and they continue to improve it
- ghclass and GitHub Classroom work together, pick the workflow that is best for you
- Different "membership" models

Q: How do you introduce Git & GitHub to students?

Introduce it early and often, and make it required.

Example materials:

- Git and GitHub intro - https://introds.org/labs/lab-01/lab-01-hello-r
- merge conflict activity - https://introds.org/labs/lab-04/lab-04-ugly-charts.html#merge-conflicts

thank you!

All materials at bit.ly/teach-r-online-mats