

GitHub Tutorials

for 2021 ASME-CIE Hackathon: Identifying, Extracting,
Analyzing Value from Large Unstructured Data Sets in
Mechanical Engineering

Anh Tran

Optimization and Uncertainty Quantification Department
Computer Science Research Institute
Sandia National Laboratories

August 13, 2021

Outline

- 1 Introduction
- 2 Hello World
- 3 Setup
- 4 Code Development
- 5 README.md
- 6 GitHub Desktop

What is GitHub?

Software development

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer.

- This cheat sheet is your friend,
- but other official guides are also available

Hello World

- go to <https://www.github.com>
- create an account (it's free!)
- create a new repository ("repo")
 - choose your favorite license for your implementation (you own your codes!)
 - write a README.md
- invite others to join your GitHub repo

GitHub Setup

- config to `~/.config`

```
git config --global user.name "FirstName LastName"
git config --global user.email "a@b.com"
git config --global color.ui auto
git config --global core.editor "nano" # or your favorite editor
```

- (optional) create a long-term key to your own device (i.e. if somebody uses your laptop then they can get to your GitHub repo WITHOUT logins)

```
ssh-keygen -t rsa -b 4096 -C "a*****@gmail.com"
# when prompt, type id_rsaGitHub
# will generate id_rsaGitHub and id_rsaGitHub.pub
eval `ssh-agent -s` # start an ssh agent
ssh-add ~/.ssh/id_rsaGitHub # add ssh-key to ~/.ssh/config
# then add ssh public key into Git account through web interface
ssh -vT git@github.com

# expect a message like this
Hi ***! You've successfully authenticated, but GitHub does not provide shell access.
```

Git in action

- clone

```
git clone https://github.com/pytorch/pytorch.git # https  
# git clone git@github.com:pytorch/pytorch.git # ssh - SSH required  
# you can also switch mode in ./git/config in the local GitHub repo
```

- typical workflow (this is what you will use the most)

```
git add # be specific, e.g. git add testABC.py  
# git add * # this is ok, but beware of your colleagues' concurrent work  
# NEVER USE: git add * -f  
  
git commit # or git commit -m "write some notes", git commit --amend  
git pull --rebase # git fetch # optional: only if there are conflicts  
  
git push # or git push origin master  
# or git stash: https://git-scm.com/docs/git-stash
```

- read logs from your teammates

```
git log
```

- remove, copy, move

```
git rm file.txt  
git mv file.txt test/
```

Git in action

- see what have been changed between commits

```
git diff  
# OR  
git diff SHA1 SHA2
```

- check status

```
git status
```

Git in action (advanced)

- reset to previous version (advanced)

```
git reset --hard <SHA> # e.g. commit a5fdab97d911414660683c89b6cecd965b55ce16
```

- create a branch

```
git branch my_debug_branch  
git checkout my_debug_branch  
  
git checkout master  
git merge my_debug_branch  
  
# git branch -v  
# git branch -list
```

- other helpful sources: [here](#)

README.md

- Pandoc/markdown style; official guide [here](#)
- text

```
It's very easy to make some words **bold**  
and other words *italic* with Markdown.  
You can even [link to Google!] (http://google.com)
```

- headers

```
# This is an <h1> tag  
## This is an <h2> tag  
##### This is an <h6> tag
```

- emphasis

```
*This text will be italic*  
_This will also be italic_  
**This text will be bold**  
__This will also be bold__  
_You **can** combine them_
```

- strikethrough

```
~~this~~
```

README.md

- list: unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

- list: ordered

```
1. Item 1
1. Item 2
1. Item 3
  1. Item 3a
  1. Item 3b
```

- images

```
![GitHub Logo] (/images/logo.png)
Format: ! [Alt Text] (url)
```

- hyperlink

```
http://github.com - automatic!
[GitHub] (http://github.com)
```

- inline code

```
I think you should use an
`<addr>` element here instead.
```

README.md

- GitHub flavored markdown

```
```python
def foo():
 if not bar:
 return True
...
```

```

```
```matlab
function y = foo(x)
 y = x
end
...
```

```

GitHub Desktop (GUI): <https://desktop.github.com/>

The screenshot shows the GitHub Desktop application interface. At the top, there's a navigation bar with File, Edit, View, Repository, Branch, and Help. Below the navigation bar, there are three main sections: "Current repository" (set to "desktop"), "Current branch" (set to "esc-pr" with PR #3972), and "Fetch origin" (last fetched 2 minutes ago).

The main area displays a list of commits from the "esc-pr" branch:

- Add event handler to dropdown component** (Co-Authored-By: Markus Olsson <niik@users.noreply.github.com>)
 - app\src\ui\toolbar\dropdown.tsx
 - 145 145 this.state = { clientRect: null }
 - 146 146 }
 - 147 147
 - 148 + private get isOpen() {
 - 149 + return this.props.dropdownState === 'open'
 - 150 + }
 - 151 +
 - 148 152 private dropdownIcon(state: DropdownState): OcticonSymbol {
 - 149 153 // @TODO: Remake triangle octicon in a 12px version,
 - 150 154 // right now it's scaled badly on normal dpi monitors.
 - 148 152 @@ -249,6 +253,13 @@ export class ToolbarDropdown extends React.Component<
 - 249 253 }
 - 250 254 }
 - 251 255
 - 256 + private onFoldoutKeyDown = (event: React.KeyboardEvent<HTMLElement>) => {
 - 257 + if (!event.defaultPrevented && this.isOpen && event.key === 'Escape') {
 - 258 + event.preventDefault()

Below the commit list, there are buttons for Changes and History, and a sidebar with links to other pull requests and repository details.

GitHub Desktop (GUI): <https://desktop.github.com/>

- official docs:
<https://docs.github.com/en/free-pro-team@latest/desktop>
- install and configure GitHub Desktop:
<https://docs.github.com/en/free-pro-team@latest/desktop/installing-and-configuring-github-desktop>
- contributing and collaborating using GitHub Desktop:
<https://docs.github.com/en/free-pro-team@latest/desktop/contributing-and-collaborating-using-github-desktop>