Working with..

Ratchapong Tantipantarak

Resource

https://git-scm.com/doc

https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

https://medium.com/@pakin/git-คืออะไร-git-is-your-friend-c609c5f8efea

https://www.slideshare.net/avilayparekh/git-undo

git cloud comparison

https://www.git-tower.com/blog/git-hosting-services-compared/

Version Control System & Git

Version Control System (VCS)

Software tools that helps in recording changes made to files by keeping a track of modifications done to the code.

Tools for Group of developers coding the same project (folder)

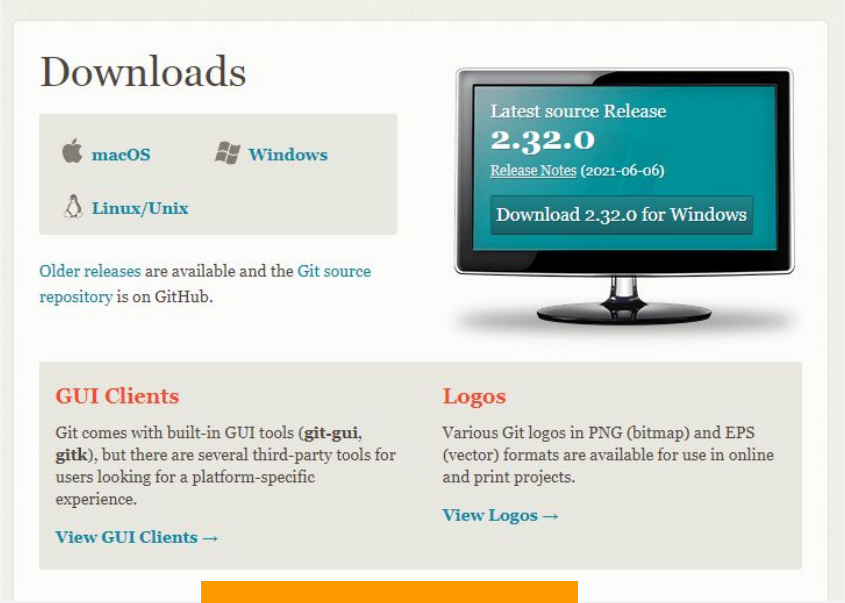
Git (General information Tracker)

คือ Version Control System ตัวหนึ่ง มีจุดเริ่มต้นจาก Linus Trovald (ผู้สร้าง Linux) ต้องการโปรแกรมแทน VCS เดิม(BitKeeper) เพื่อ support บรรดา developer ในการแก้ไข source code ของ Linux kernel ซึ่งเป็นโครงการ Opensource ที่มี developer ร่วมกันทำงาน 5-6 พันคน และเนื่องจาก git เป็น VCS ที่ใช้งานได้ฟรี และมีประสิทธิภาพสูง Git จึงเป็น VCS ได้รับความนิยมในการใช้งานทั่วโลก

Download git

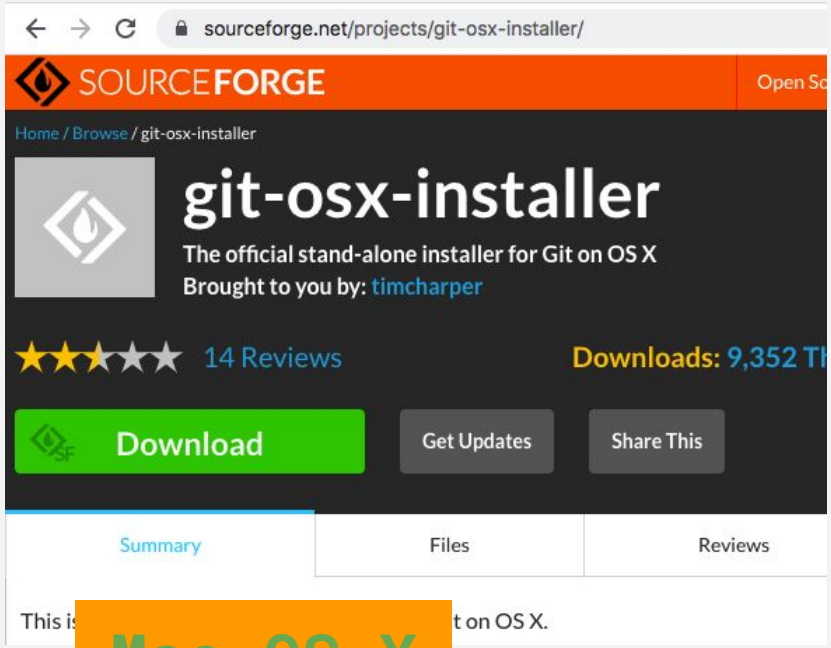
Software Park Thailand
</Code Camp>

https://git-scm.com/downloads



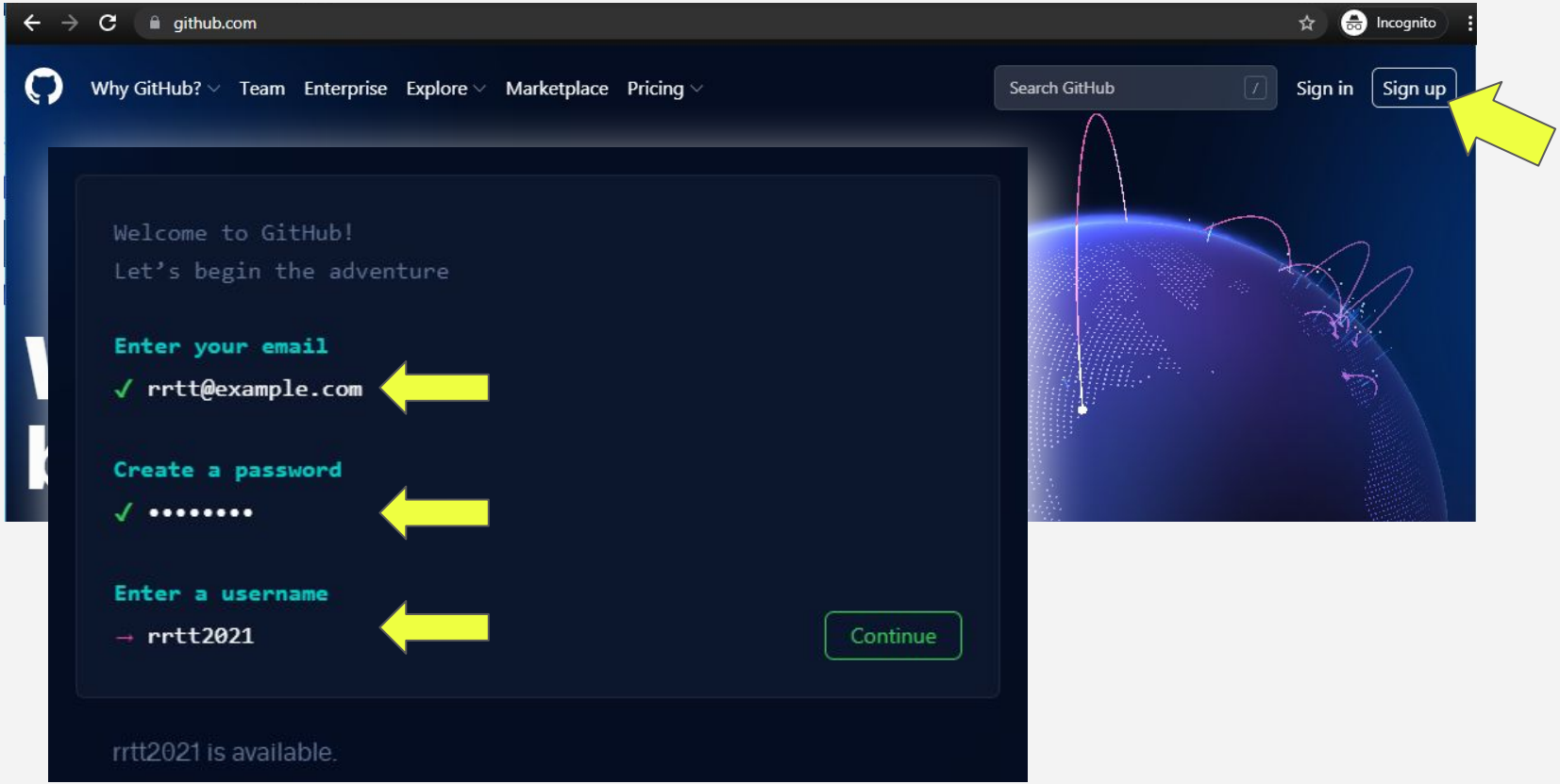
Windows

https://sourceforge.net/projects/git-osx-installer/



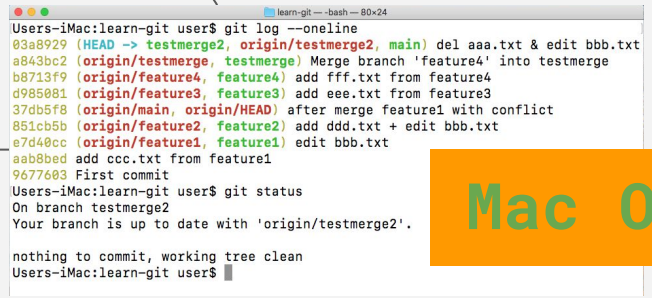
Mac OS X

Register : github.com



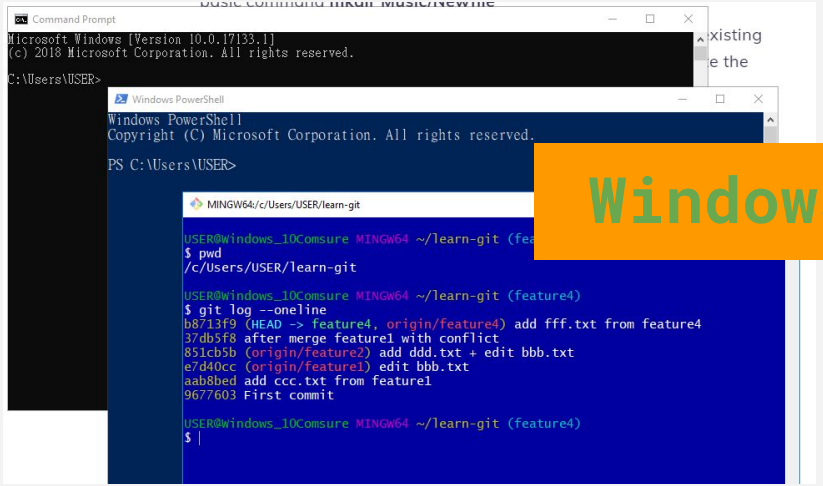
The Terminal (OS)

Command Line Interface (CLI)
vs. GUI (Graphic User Interface)

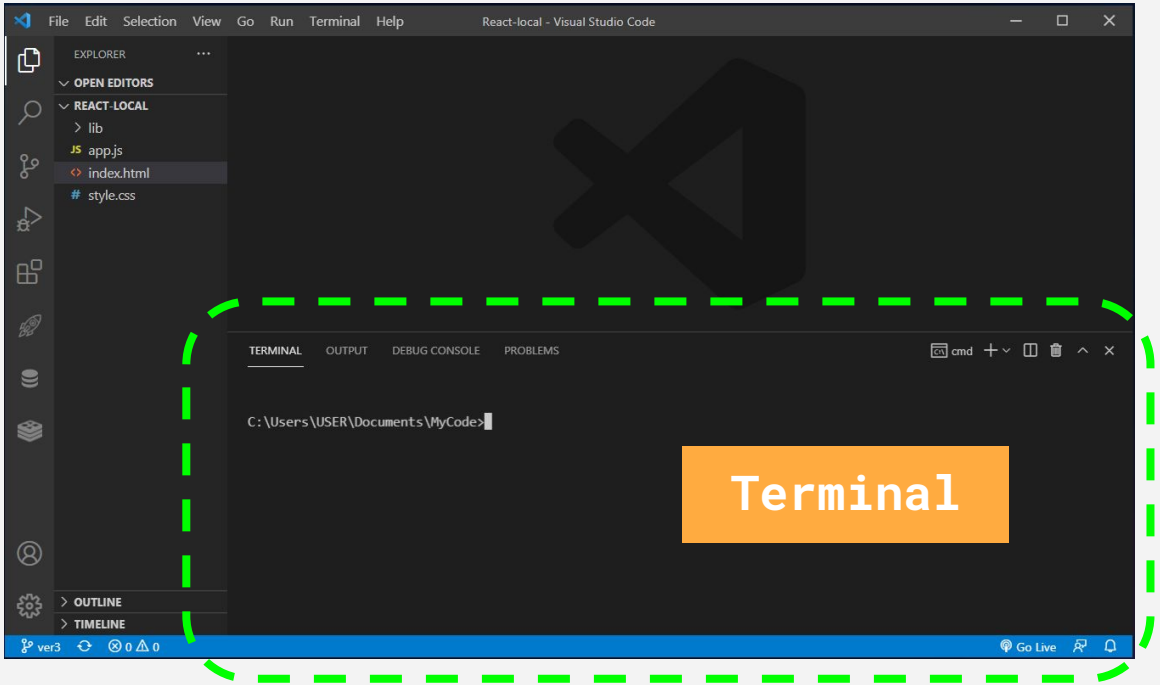


Git command with

* Linux : Bash Shell
* Mac OS : Bash Shell / Z Shell
* MS Windows :
  - CMD (from history...
    - MS DOS Prompt)
  - PowerShell (> windows 7)
  - Bash Shell from Git Bash



VSCode : terminal

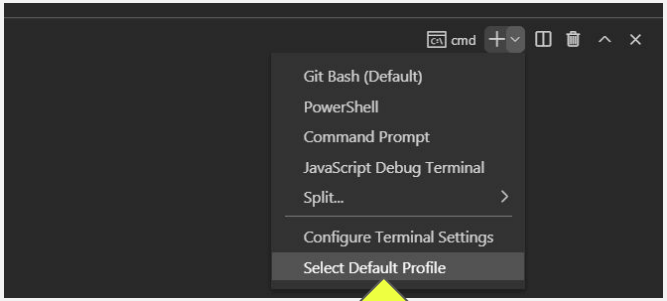


Shortcut Keys

CTRL+J : เปิด / ปิด Terminal

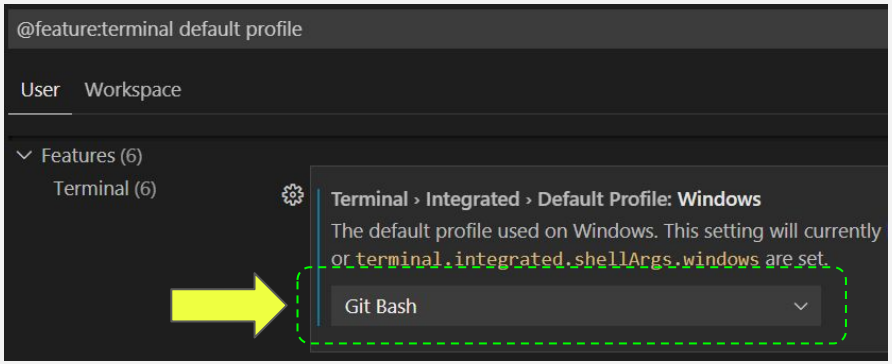
CTRL+Shift+C :

เปิด external Terminal



เปลี่ยน default shell

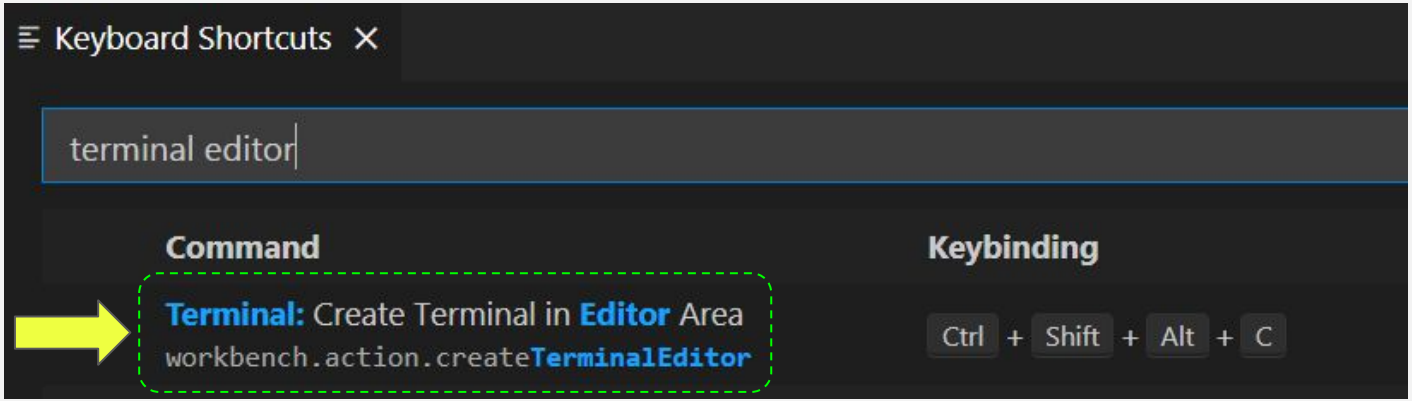
VSCode : terminal settings



Shortcut Keys

CTRL+Shift+P : พิมพ์คำสั่งใน VSCode

CTRL+K,CTRL+S : ตั้ง shortcut-key



Setup : git config (ทำรอบแรกรอบเดียว)

Software Park Thailand
</Code Camp>

...หลังติดตั้ง git (และ register ที่ github.com แล้ว)

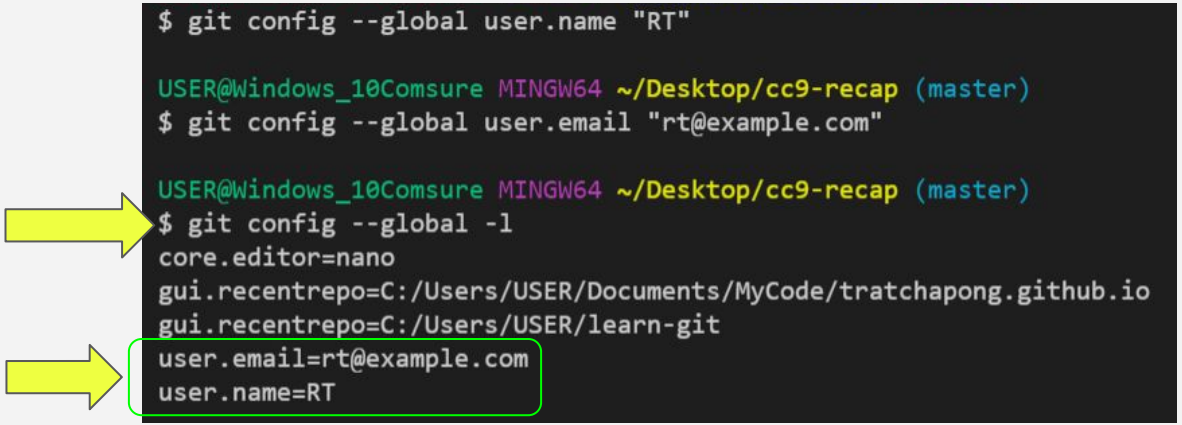
พิมพ์คำสั่ง

```
git config --global user.name “ชื่อ user”
git config --global user.email “email ที่ลงทะเบียน”
```

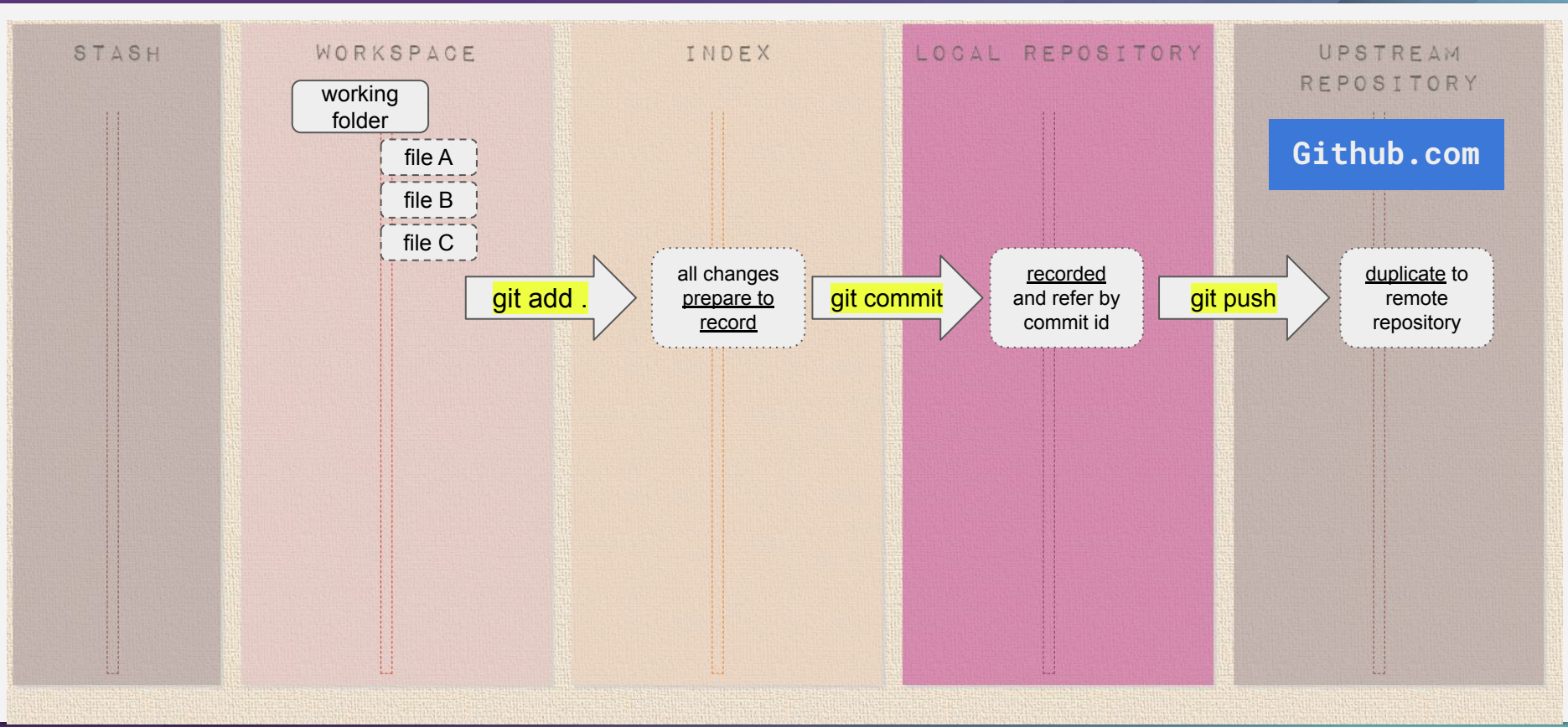
พิมพ์คำสั่ง

```
git config --global -l
```

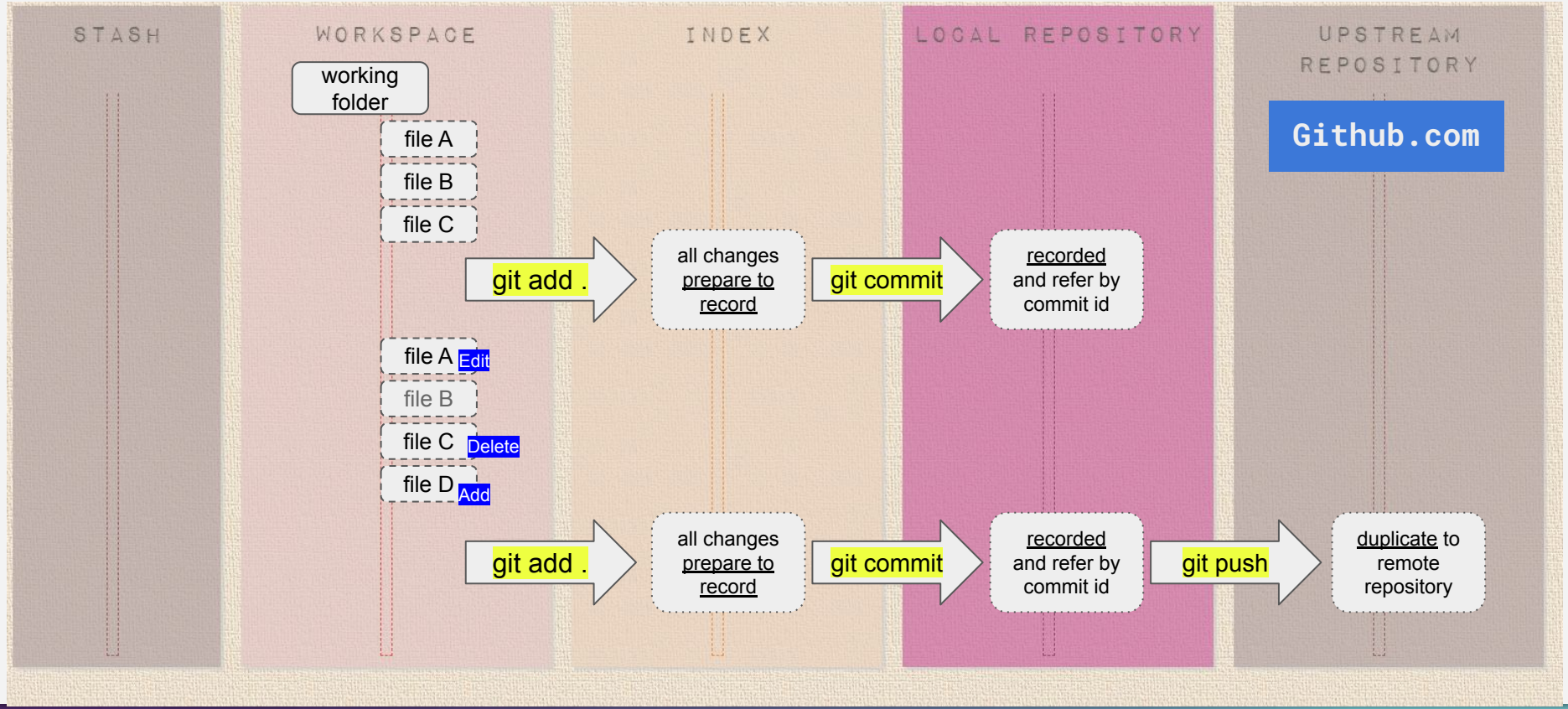
เพื่อดูสิ่งที่ config



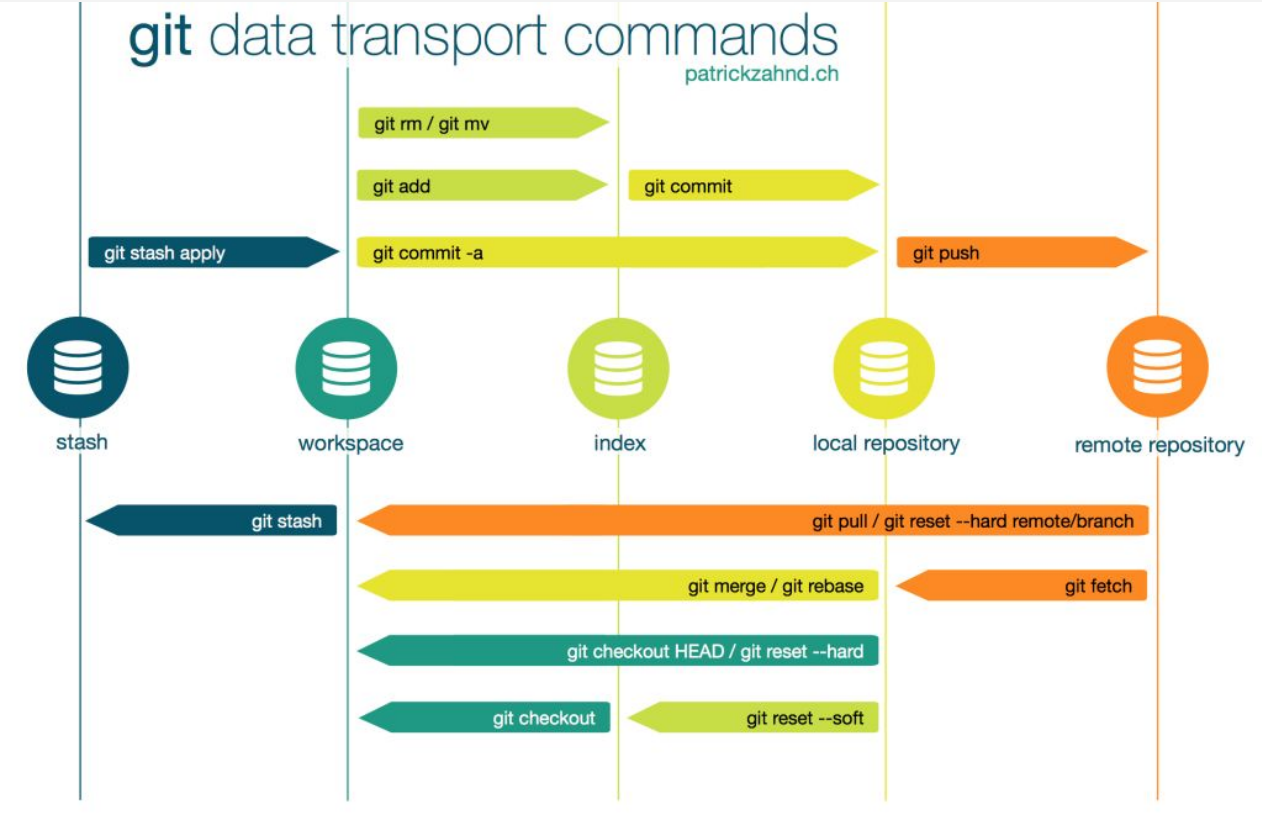
Git Concept


STASH
WORKSPACE
working folder
file A
file B
file C
git add .
INDEX
all changes prepare to record
git commit
LOCAL REPOSITORY
recorded and refer by commit id
git push
UPSTREAM REPOSITORY
Github.com
duplicate to remote repository

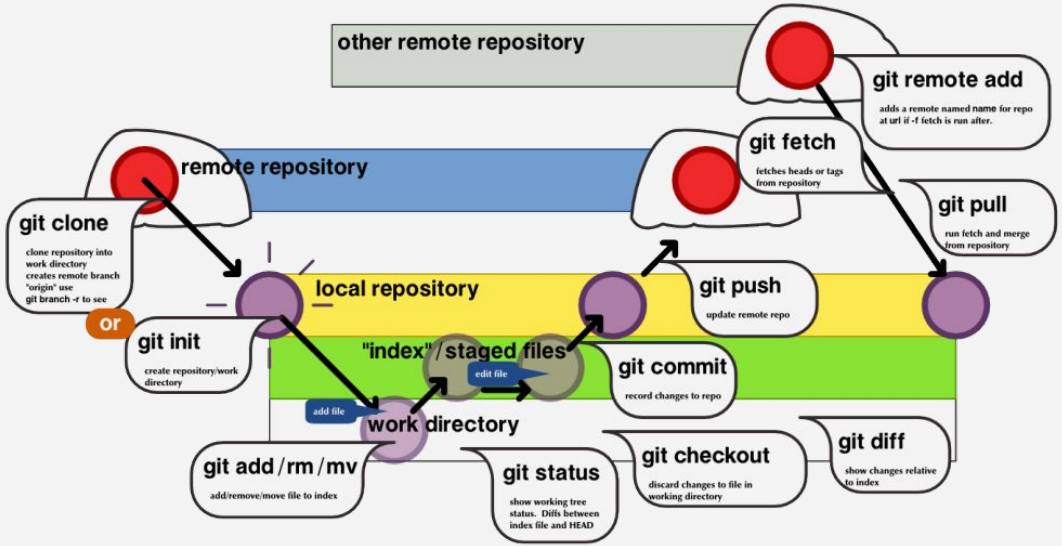
Git Concept


STASH
WORKSPACE
INDEX
LOCAL REPOSITORY
UPSTREAM REPOSITORY
working folder
file A
file B
file C
git add .
all changes prepare to record
git commit
recorded and refer by commit id
Github.com
file A Edit
file B
file C Delete
file D Add
git add .
all changes prepare to record
git commit
recorded and refer by commit id
git push
duplicate to remote repository

Git Concept


git data transport commands
patrickzahnd.ch
git rm / git mv
git add
git commit
git stash apply
git commit -a
git push
stash
workspace
index
local repository
remote repository
git stash
git pull / git reset --hard remote/branch
git merge / git rebase
git fetch
git checkout HEAD / git reset --hard
git checkout
git reset --soft

Git Concept



GIT Visual cheat sheet

https://github.com/mattharrison/Git-Supervisual-Cheatsheet

Overview all basic git commands..

Create a Repository

From scratch -- Create a new local repository
```
$ git init [project name]
```

Download from an existing repository
```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed
```
$ git status
```

Show the changes to files not yet staged
```
$ git diff
```

Show the changes to staged files
```
$ git diff --cached
```

Show all staged and unstaged file changes
```
$ git diff HEAD
```

Show the changes between two commit ids
```
$ git diff commit1 commit2
```

List the change dates and authors for a file
```
$ git blame [file]
```

Show the file changes for a commit id and/or file
```
$ git show [commit]:[file]
```

Show full change history
```
$ git log
```

Show change history for file/directory including diffs
```
$ git log -p [file/directory]
```

Working with Branches

List all local branches
```
$ git branch
```

List all branches, local and remote
```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory
```
$ git checkout my_branch
```

Create a new branch called new_branch
```
$ git branch new_branch
```

Delete the branch called my_branch
```
$ git branch -d my_branch
```

Merge branch_a into branch_b
```
$ git checkout branch_b
$ git merge branch_a
```

Tag the current commit
```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit
```
$ git add [file]
```

Stage all changed files, ready for commit
```
$ git add .
```

Commit all staged files to versioned history
```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history
```
$git commit -am "commit message"
```

Unstages file, keeping the file changes
```
$ git reset [file]
```

Revert everything to the last commit
```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)
```
$ git fetch
```

Fetch the latest changes from origin and merge
```
$ git pull
```

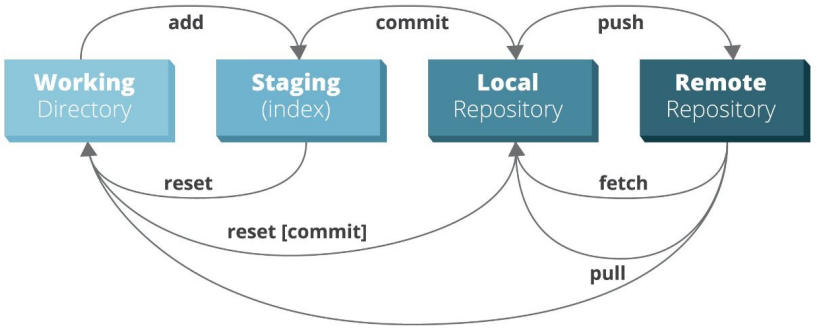
Fetch the latest changes from origin and rebase
```
$ git pull --rebase
```

Push local changes to the origin
```
$ git push
```

Finally!

When in doubt, use git help
```
$ git command --help
```

Or visit https://training.github.com/ for official GitHub training.



Start : git init (ทำตอนเริ่ม project ครั้งเดียว)

Software Park Thailand
</Code Camp>

cd ไปที่โฟลเดอร์ของ project

พิมพ์คำสั่ง

git init

ให้เริ่มเก็บข้อมูลระบบ git ในโฟลเดอร์นี้

```
$ git init
Initialized empty Git repository in C:/Users/USER/Desktop/cc9-recap/.git/
```

ลองสั่ง ls -al จะพบโฟลเดอร์ .git

```
$ ls -al
total 18
drwxr-xr-x 1 USER 197121   0 Jul 14 16:53 ./
drwxr-xr-x 1 USER 197121   0 Jul 14 15:06 ../
drwxr-xr-x 1 USER 197121   0 Jul 14 16:53 .git/
-rw-r--r-- 1 USER 197121 477 Jul 14 16:02 index.html
-rw-r--r-- 1 USER 197121 416 Jul 14 15:31 notes.html
```

เก็บข้อมูลของ git ทั้งหมด

git status : สถานะของ git

cd ไปที่โฟลเดอร์ของ project

พิมพ์คำสั่ง

git status

แสดงสถานะปัจจุบันของไฟล์ (ที่ถูกเปลี่ยนแปลง) ใน project

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
        notes.html

nothing added to commit but untracked files present (use "git add" to track)
```

ไฟล์ที่ยังไม่ขึ้น staging (สีแดง)

git add : จัดไฟล์ขึ้นเวที (staging)

Software Park Thailand
</Code Camp>

cd ไปที่โฟลเดอร์ของ project

พิมพ์คำสั่ง

git add <file>

เตรียมเก็บข้อมูลการเปลี่ยนแปลงของไฟล์ต่อไปนี้..

```
$ git add .
```

*** ส่วนมากใช้ git add . ซึ่งจะรวมทุกไฟล์ในโฟลเดอร์ + โฟลเดอร์ย่อยทั้งหมดด้วย*

```
USER@Windows_10Comsure MINGW64 ~/Desktop/cc9-recap (master)
$ git status
On branch master

No commits yet

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

สั่ง git status เพื่อดูข้อมูลสถานะของ git

ยกเลิกการ Add
โดยทียังไม่เคยมีการ commit

```
git rm -r --cached .
```

```
  (use "git rm --cached <file>..." to unstage)
```

```
  (use "git restore --staged <file>..." to unstage)
```

```
git restore --stage .
```

ยกเลิกการ Add
โดยมีการ commit แล้วอย่างน้อย 1 ครั้ง

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   notes.html
```

git commit : จัดเก็บข้อมูล (local-repo)

Software Park Thailand
</Code Camp>

cd ไปที่โฟลเดอร์ของ project

พิมพ์คำสั่ง

เก็บข้อมูลการเปลี่ยนแปลงของไฟล์ที่ถูก add ไว้ใน staging

git commit -m “ข้อความอธิบาย”

ใส่ -m “ข้อความ” ทุกครั้งที่ commit

```
$ git commit -m "First init"
[master (root-commit) 2f28211] First init
 2 files changed, 35 insertions(+)
 create mode 100644 index.html
 create mode 100644 notes.html

USER@Windows_10Comsure MINGW64 ~/Desktop/cc9-recap (master)
$ git status
On branch master
nothing to commit, working tree clean
```

หลัง commit ลอง git status ดู

`git reset --soft HEAD^1`

วิธียกเลิกการ commit ล่าสุด

```
$ git log --oneline
48afa5f (HEAD -> main) Edit about.html line 01
8ecc87a add line03 index.html
8cd562c edit index.html line01
ff622d4 add css + note.html
84654ae First Commit 2 files

USER@Windows_10Comsure MINGW64 ~/D
$ git reset --soft HEAD^1

USER@Windows_10Comsure MINGW64 ~/Desktop/Learn-git (main)
$ git log --oneline
8ecc87a (HEAD -> main) add line03 index.html
8cd562c edit index.html line01
ff622d4 add css + note.html
84654ae First Commit 2 files

USER@Windows_10Comsure MINGW64 ~/Desktop/Learn-git (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   about.html
```

ย้อนหลัง Head ไป 1 step

วงจรการทำงานกับ Git

Normal flow of git (local)

แก้ไขไฟล์ => git add . => git commit -m "..."

การ commit 1 ครั้ง คือการเก็บสภาพโฟลเดอร์นั้นไว้ 1 ชุด

จะได้เลข commit id เพื่อใช้อ้างอิง commit นั้นได้

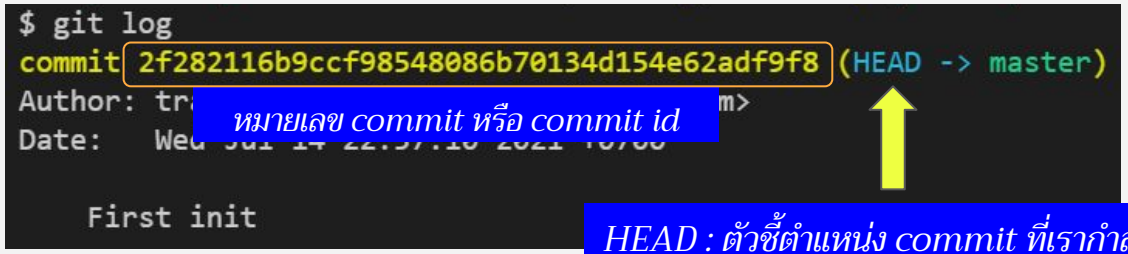
git log : ดูประวัติการ commit (1)

cd ไปที่โฟลเดอร์ของ project

พิมพ์คำสั่ง

git log

แสดงประวัติการ commit ทั้งหมดของโฟลเดอร์นั้น



git log : ดูประวัติการcommit (2)

Software Park Thailand
</Code Camp>

git log --oneline

แสดง log แบบ 1 line : 1 commit

```
$ git log
commit 6eb3d3a8debad72e64ca80a9f489f36f79f400e9 (HEAD -> master)
Author: tratchapong <tratchapong@gmail.com>
Date:   Thu Jul 15 00:20:22 2021 +0700

    add link to notes.html

commit 2f282116b9ccf98548086b70134d154e62adf9f8
Author: tratchapong <tratchapong@gmail.com>
Date:   Wed Jul 14 22:57:10 2021 +0700

    First init
```

```
$ git log --oneline
6eb3d3a (HEAD -> master) add link to notes.html
2f28211 First init
```

git diff : ดูความแตกต่างในเนื้อหาไฟล์

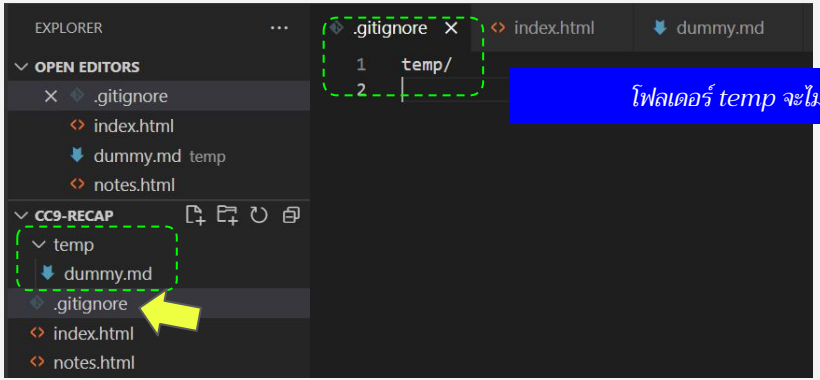
git diff	เปรียบเทียบระหว่าง working - staging
git diff <commit id>	เปรียบเทียบระหว่าง working - commit id
git diff <commit id> <commit id>	เปรียบเทียบระหว่าง 2 commit id

```
$ git diff
diff --git a/index.html b/index.html
index 6de2489..5821d57 100644
--- a/index.html
+++ b/index.html
@@ -11,5 +11,6 @@
    <p>line01 Lorem ipsum dolor sit amet.</p>
    <p>line02 Lorem ipsum dolor sit amet.</p>
    <p>line03 Lorem ipsum dolor sit amet.</p>
+   <hr>
  </body>
  </html>
\ No newline at end of file
```

```
$ git diff HEAD~2 HEAD~1
diff --git a/about.html b/about.html
index 2da28a1..762f57a 100644
--- a/about.html
+++ b/about.html
@@ -8,5 +8,7 @@
  </head>
  <body>
    <h2>About Page</h2>
+   <p>line01 Lorem ipsum dolor sit amet consectetur adipisicing.</p>
+   <p>line02 Lorem ipsum dolor sit amet consectetur adipisicing.</p>
  </body>
  </html>
\ No newline at end of file
diff --git a/css/style2.css b/css/style2.css
index eebf6a0..ba4d45c 100644
--- a/css/style2.css
+++ b/css/style2.css
@@ -1,4 +1,5 @@
 h1 {
    color: blue;
    background-color: white;
+   border: 1px solid lime;
```

.gitignore : ไฟล์

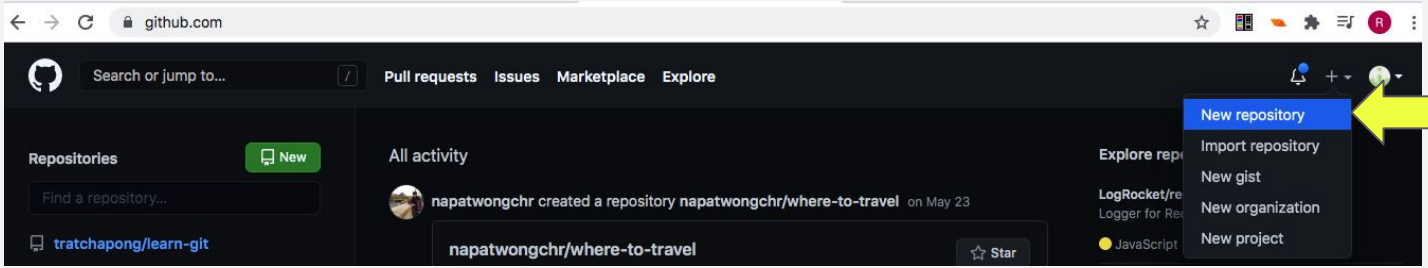
สร้างไว้ในโฟลเดอร์ของ project เพื่อระบุไฟล์ & โฟลเดอร์ที่จะไม่ส่งขึ้น staging ตลอดไป (ยกเว้นสั่ง git add -f <file>)



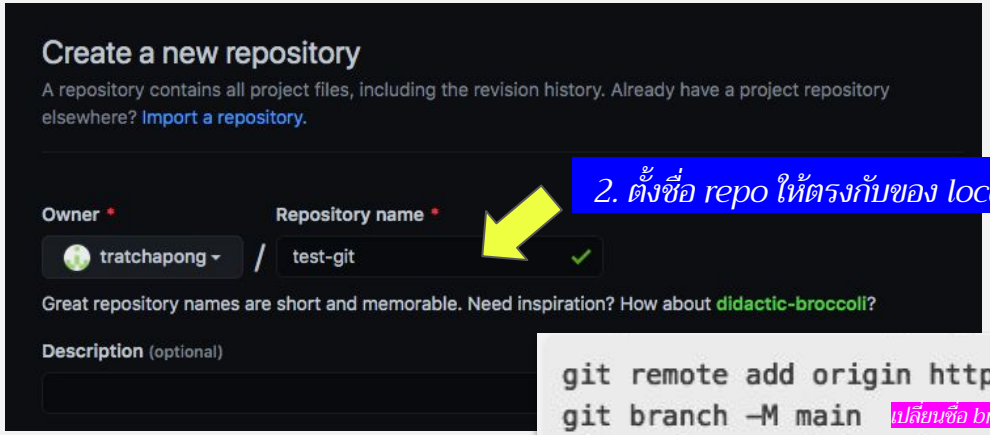

```
$ git add .

USER@Windows_10Comsure MINGW64 ~/Desktop/cc9-recap (master)
$ git add temp
The following paths are ignored by one of your .gitignore files:
temp
hint: Use -f if you really want to add them.
hint: Turn this message off by running
hint: "git config advice.addIgnoredFile false"
```

Remote repo : *login ที่ github.com*



1. สร้าง New repository



2. ตั้งชื่อ repo ให้ตรงกับของ local

3. พิมพ์คำสั่งเหล่านี้ที่เครื่องของเรา

เชื่อม remote repo ด้วยชื่อ origin

```
git remote add origin https://github.com/tratchapong/test-git.git
git branch -M main
git push -u origin main
```

เปลี่ยนชื่อ branch เป็น main

push ข้อมูลไปที่ remote repo

git push : ส่งข้อมูลขึ้น remote repo.

Software Park Thailand
</Code Camp>

```
git push -u origin main
```

ส่งข้อมูล branch main ขึ้น remote repo

upstream ชื่อ origin

ชื่อ branch

ถ้าเป็น branch main สามารถละได้

```
$ git log -oneline
c4ebf95 (HEAD -> main) edit about.html
d0cb3a0 (origin/main) add gitignore
7f4dc05 edit notes.html
ab7f653 add css
cdf516b add about.html
938298a add script
bd684c3 edit notes.html
574a682 edit index.html
21fa5bc First init
```

```
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objec
To https://github.com/tratchapong/git-recap.git
   d0cb3a0..c4ebf95  main -> main
```

```
$ git log --oneline
c4ebf95 (HEAD -> main, origin/main) edit about.html
d0cb3a0 add gitignore
7f4dc05 edit notes.html
ab7f653 add css
cdf516b add about.html
938298a add script
bd684c3 edit notes.html
574a682 edit index.html
21fa5bc First init
```

git checkout (commit id)

Software Park Thailand
</Code Camp>

ย้าย HEAD ไปที่ commit id อื่น (detached HEAD)

git checkout <commit id>

HEAD คือตัวชี commit ที่ active อยู่

```
$ git log --oneline
c4ebf95 (HEAD -> main, origin/main) edit about.html
d0cb3a0 add gitignore
7f4dc05 edit notes.html
ab7f653 add css
cdf516b add about.html
938298a add script
bd684c3 edit notes.html
574a682 edit index.html
21fa5bc First init
```

ปกติ HEAD จะชี้ที่ commit ล่าสุด (ของ local branch)

commit id

```
$ git checkout 938298a
Note: switching to '938298a'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 938298a add script
```

สร้าง branch ใหม่ ที่ commit นี้

ย้าย HEAD กลับที่เดิม

```
$ git log --oneline
938298a (HEAD) add script
bd684c3 edit notes.html
574a682 edit index.html
21fa5bc First init

USER@Windows_10Comsure MINGW64 ~/Desktop/git-recap
$ git switch -
Previous HEAD position was 938298a add script
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

ย้าย HEAD กลับที่เดิม

git checkout -b : สร้าง branch

ตั้งชื่อ HEAD กำกับ Commit นั้นเพื่อเตรียมแตก branch (สาขา)

git checkout -b <ชื่อ branch>

*** เราสามารถย้าย HEAD ไปสร้าง branch ในจุดที่ต้องการได้ ***

```
$ git log --oneline
c4ebf95 (HEAD -> main, origin/main) edit about.
d0cb3a0 add gitignore
7f4dc05 edit notes.html
ab7f653 add css
cdf516b add about.html
938298a add script
bd684c3 edit notes.html
574a682 edit index.html
21fa5bc First init
```

```
$ git log --oneline
c4ebf95 (HEAD -> test-branch, origin/main, main) edit about.
d0cb3a0 add gitignore
7f4dc05 edit notes.html
ab7f653 add css
cdf516b add about.html
938298a add script
bd684c3 edit notes.html
574a682 edit index.html
21fa5bc First init
```

```
$ git checkout -b test-branch
Switched to a new branch 'test-branch'
```

git branch : การจัดการ branch

Software Park Thailand
</Code Camp>

git help branch เรียกดู Help ของคำสั่ง branch

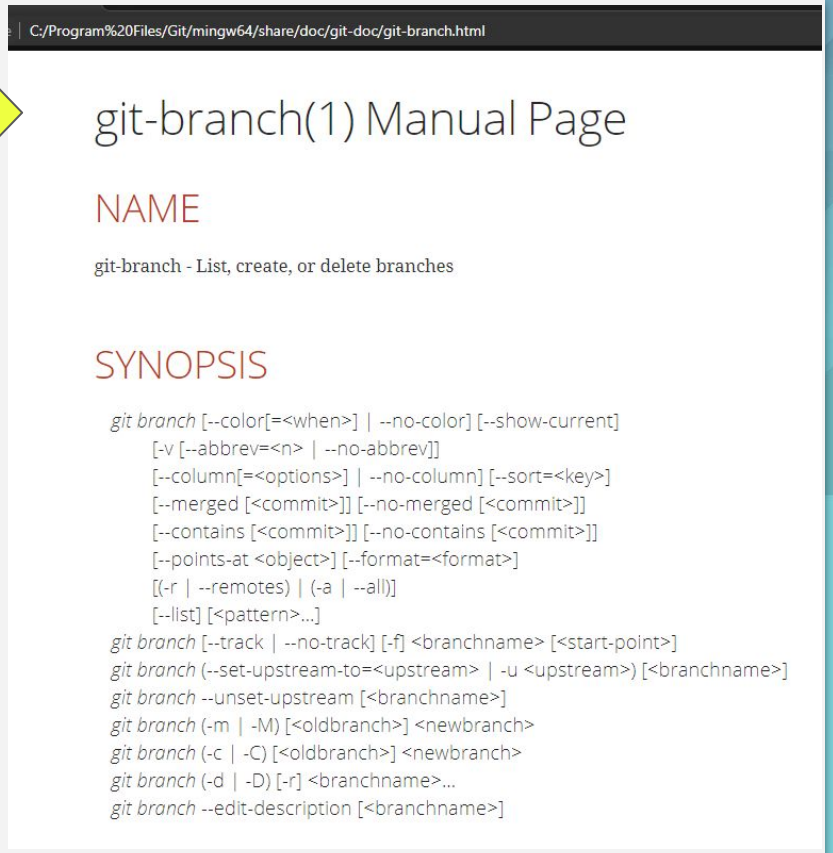
---

git branch -d <ชื่อ branch>

ลบ branch

git branch -m <ชื่อเดิม> <ชื่อใหม่>

เปลี่ยนชื่อ branch



https://www.nobledesktop.com/learn/git/git-branches

git merge : รวมงานจาก branch อื่น

git checkout / switch ไปยืนรอที่ branch หลักก่อน

git merge <ชื่อ branch> merge จาก branch ที่ระบุมาสู่ branch หลัก

```
USER@Windows_10Comsure MINGW64 ~/Desktop/Learn-git (b02)
$ git checkout main
Switched to branch 'main'

USER@Windows_10Comsure MINGW64 ~/Desktop/Learn-git (main)
$

USER@Windows_10Comsure MINGW64 ~/Desktop/Learn-git (main)
$ git merge b02
Merge made by the 'recursive' strategy.
 about.html | 1 +
 index.html | 1 +
 2 files changed, 2 insertions(+)
```

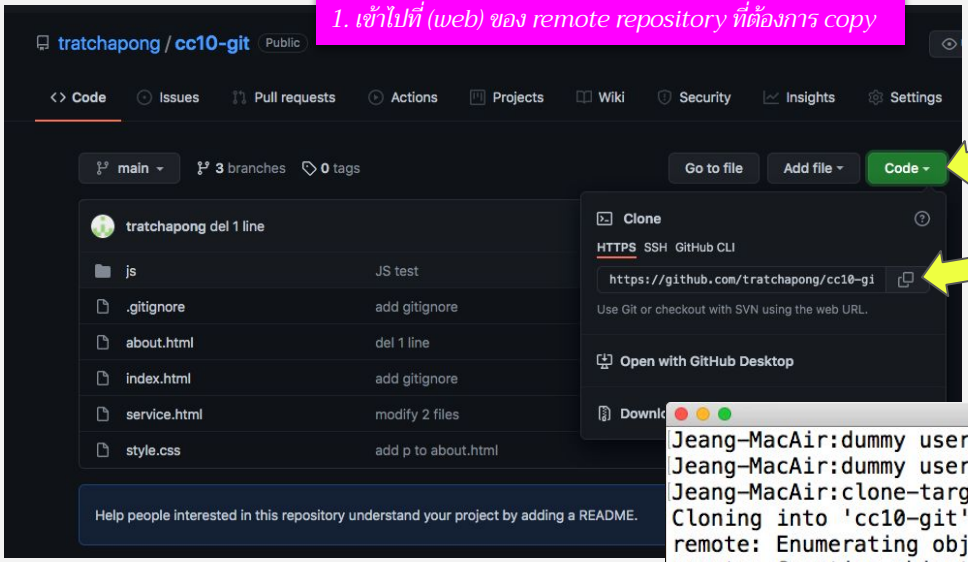
git clone : สำเนา remote repo.

Software Park Thailand
</Code Camp>

git clone <url>

copy Remote Repository มาไว้ที่ local

1. เข้าไปที่ (web) ของ remote repository ที่ต้องการ copy



2. Click ที่ Code

3. Click เพื่อ copy URL

4. cd ไปที่โฟลเดอร์ปลายทางก่อนสั่ง git clone

```
Jeang-MacAir:dummy user$ mkdir clone-target
Jeang-MacAir:dummy user$ cd clone-target
Jeang-MacAir:clone-target user$ git clone https://github.com/tratchapong/cc10-git.git
Cloning into 'cc10-git'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 47 (delta 23), reused 43 (delta 19), pack-reused 0
Unpacking objects: 100% (47/47), done.
Jeang-MacAir:clone-target user$ ls
cc10-git
```

git rebase : ย้าย base ของ branch

git stash : เก็บ working ไว้เรียกใช้ทีหลัง

git *when conflict* : เมื่อมี Conflict

git basic

https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

Software Park Thailand
</Code Camp>

GIT BASICS

`git init <directory>`	Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository.
`git clone <repo>`	Clone repo located at `<repo>` onto local machine. Original repo can be located on the local filesystem or on a remote machine via HTTP or SSH.
`git config user.name <name>`	Define author name to be used for all commits in current repo. Devs commonly use `--global` flag to set config options for current user.
`git add <directory>`	Stage all changes in `<directory>` for the next commit. Replace `<directory>` with a `<file>` to change a specific file.
`git commit -m "<message>"`	Commit the staged snapshot, but instead of launching a text editor, use `<message>` as the commit message.
`git status`	List which files are staged, unstaged, and untracked.
`git log`	Display the entire commit history using the default format. For customization see additional options.
`git diff`	Show unstaged changes between your index and working directory.

git branch

https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

Software Park Thailand
</Code Camp>

GIT BRANCHES

`git branch`	List all of the branches in your repo. Add a `<branch>` argument to create a new branch with the name `<branch>`.
`git checkout -b <branch>`	Create and check out a new branch named `<branch>`. Drop the `-b` flag to checkout an existing branch.
`git merge <branch>`	Merge `<branch>` into the current branch.

git log

https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

Software Park Thailand
</Code Camp>

GIT LOG

`git log -<limit>`	Limit number of commits by `<limit>`. E.g. `"git log -5"` will limit to 5 commits.
`git log --oneline`	Condense each commit to a single line.
`git log -p`	Display the full diff of each commit.
`git log --stat`	Include which files were altered and the relative number of lines that were added or deleted from each of them.
`git log --author="<pattern>"`	Search for commits by a particular author.
`git log --grep="<pattern>"`	Search for commits with a commit message that matches `<pattern>`.
`git log <since>..<until>`	Show commits that occur between `<since>` and `<until>`. Args can be a commit ID, branch name, `HEAD`, or any other kind of revision reference.
`git log -- <file>`	Only display commits that have the specified file.
`git log --graph --decorate`	`--graph` flag draws a text based graph of commits on left side of commit msgs. `--decorate` adds names of branches or tags of commits shown.

git undo..

https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

Software Park Thailand
</Code Camp>

UNDOING CHANGES

`git revert <commit>`	Create new commit that undoes all of the changes made in `<commit>`, then apply it to the current branch.
`git reset <file>`	Remove `<file>` from the staging area, but leave the working directory unchanged. This unstages a file without overwriting any changes.
`git clean -n`	Shows which files would be removed from working directory. Use the `-f` flag in place of the `-n` flag to execute the clean.

git reset..

https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

Software Park Thailand
</Code Camp>

GIT RESET

`git reset`	Reset staging area to match most recent commit, but leave the working directory unchanged.
`git reset --hard`	Reset staging area and working directory to match most recent commit and **overwrites all changes** in the working directory.
`git reset <commit>`	Move the current branch tip backward to `<commit>`, reset the staging area to match, but leave the working directory alone.
`git reset --hard <commit>`	Same as previous, but resets both the staging area & working directory to match. **Deletes** uncommitted changes, and **all commits after** `<commit>`.

git push & pull

https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

Software Park Thailand
</Code Camp>

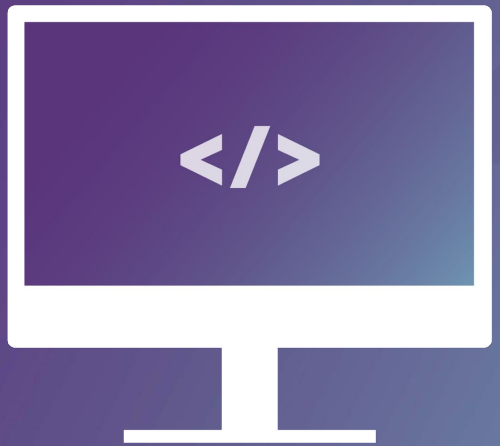
GIT PUSH

`git push <remote> --force`	Forces the `git push` even if it results in a non-fast-forward merge. Do not use the `--force` flag unless you're absolutely sure you know what you're doing.
`git push <remote> --all`	Push all of your local branches to the specified remote.
`git push <remote> --tags`	Tags aren't automatically pushed when you push a branch or use the `--all` flag. The `--tags` flag sends all of your local tags to the remote repo.

GIT PULL

`git pull --rebase <remote>`	Fetch the remote's copy of current branch and rebases it into the local copy. Uses git rebase instead of merge to integrate the branches.

[subject]



หัวข้อ

- What is Git?
- Download & Setup
- Basic file/folder command
  - Git Bash / Terminal
- Repository
  - local
  - remote
- Normal workflow with Git
- Stage in Git
  - *workspace*
  - *index*
  - *local repository*
  - *remote repository*
  - *stash*
- start project with Git
  - new from local
  - existing project
  - new from remote
- Undo changes

Basic File / Folder Command

Software Park Thailand
</Code Camp>

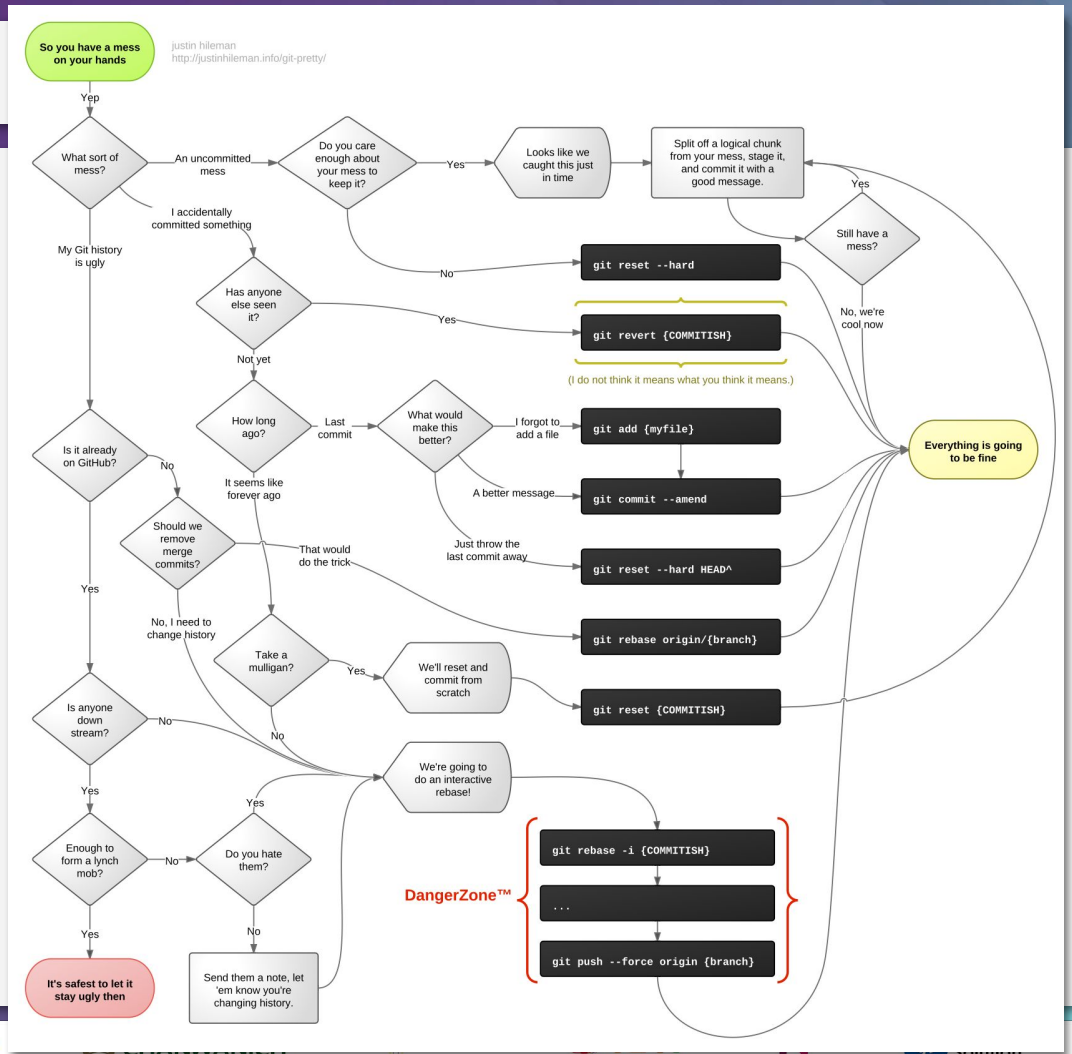
Based on Bash

https://www.educative.io/blog/bash-shell-command-cheat-sheet

- pwd
- ls
- cd
- mkdir
- rmdir
- cat
- echo
- pipe (|) and re-direction (>)
- mv
- cp

- nano (text editor)

git undo diagram



Staging : git add .

[หลังแก้ไข code, ไฟล์, โฟลเดอร์ ของ Project]
เมื่อพร้อมจะ commit สั่ง

git add . หรือ git add <filename list>

ลอง git status เพื่อดูสถานะ (ทั้งก่อนและหลัง git add)

Commit : git commit -m “the message”

Software Park Thailand
</Code Camp>

[เมื่อ staging พร้อม]
สั่ง

git commit -m “the commit message”

ลอง git status เพื่อดูสถานะ (ทั้งก่อนและหลัง git commit)

ลอง git log เพื่อดูรายละเอียดของ Head (index)