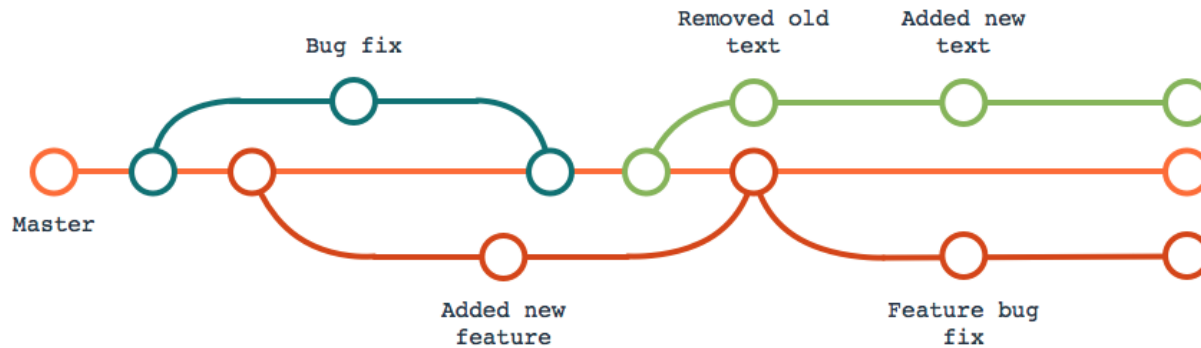


# GIT



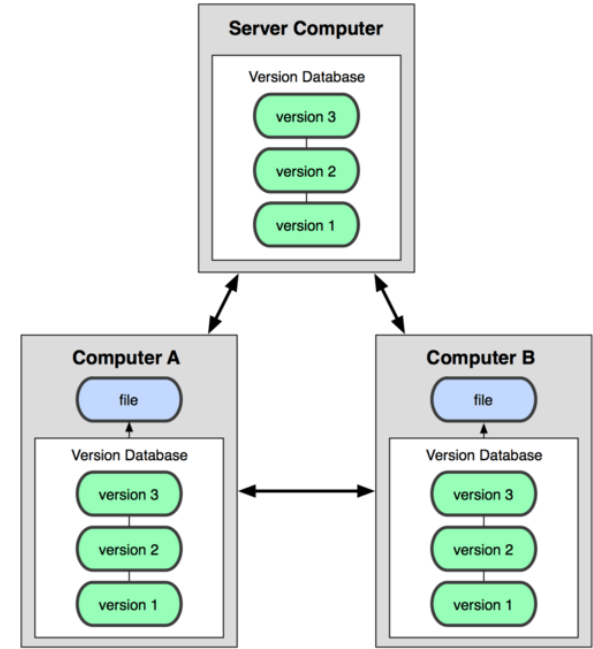
# GIT คืออะไร?

**GIT** คือ ระบบที่จัดการเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่งหรือหลายไฟล์เพื่อให้สามารถเรียกเวอร์ชันใดเวอร์ชันหนึ่งกลับมาดูเมื่อไรก็ได้



# ข้อดีของ GIT

- สามารถควบคุมเวอร์ชันได้ โดยสามารถย้อนกลับไปเวอร์ชันเก่าได้ถ้าหากเกิดปัญหาขึ้น
- สามารถติดตามการเปลี่ยนแปลงของ code ได้ผ่านการ commit
- ช่วยพัฒนาการทำงานเป็นทีม เนื่องจากทุกคนในทีมได้รับโค้ดที่อัปเดตล่าสุดได้



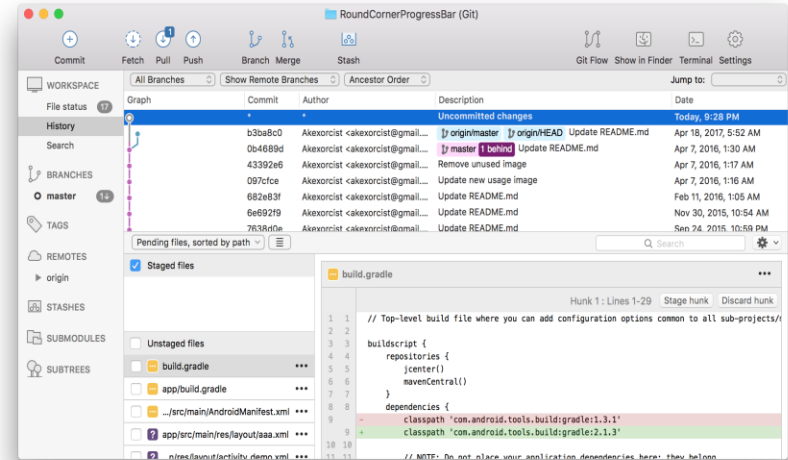
# การใช้งาน GIT

## GUI

- Sourcetree, Gitkraken, Github desktop

## Command Line

- Git scm



# Server สำหรับให้บริการ GIT



# คำที่ต้องรู้จักใน GIT

## Repository

คือ Folder ที่ใช้เก็บข้อมูลโดยที่ 1 Project จะมี 1 Repository

## Clone

คือการ Copy Repository จาก Server มาลงเครื่องของเรา

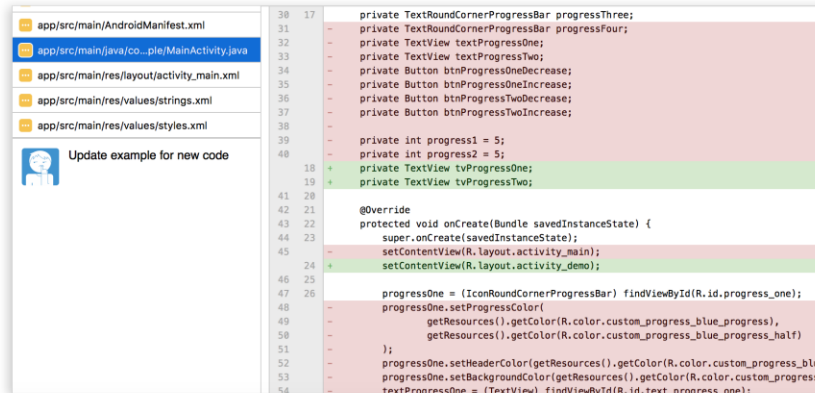


# คำที่ต้องรู้จักใน GIT

## Commit

คือการ Backup เก็บไว้ใน Server ซึ่งการ Commit จะสามารถเลือกได้ว่าจะเอาไฟล์ไหนบ้าง (ไม่จำเป็นต้องเลือกทุกไฟล์)

ซึ่งการ Commit ในแต่ละครั้งจะบันทึกว่ามีตรงไหนของข้อมูลที่ถูกเปลี่ยนแปลงไป แต่สามารถย้อนดู History ได้ว่ามีการแก้ไขอะไรบ้าง ทำให้รู้ว่าใน Commit นั้นๆ แต่ละไฟล์มีข้อมูล



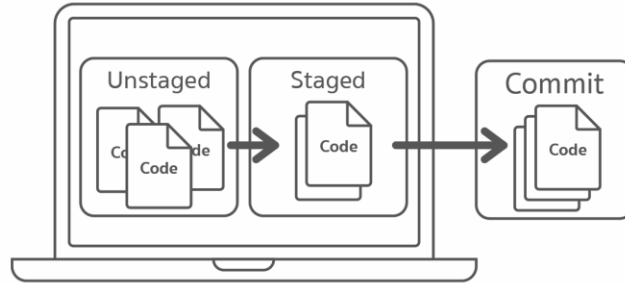
```
30 17 private TextRoundCornerProgressBar progressThree;
31 - private TextRoundCornerProgressBar progressFour;
32 - private TextView textProgressOne;
33 - private TextView textProgressTwo;
34 - private Button btnProgressOneDecrease;
35 - private Button btnProgressOneIncrease;
36 - private Button btnProgressTwoDecrease;
37 - private Button btnProgressTwoIncrease;
38 -
39 -
40 - private int progress1 = 5;
41 - private int progress2 = 5;
42 + private TextView tvProgressOne;
43 + private TextView tvProgressTwo;
44
45
46 20
47 21
48 22 @Override
49 23 protected void onCreate(Bundle savedInstanceState) {
50 24     super.onCreate(savedInstanceState);
51 25     setContentView(R.layout.activity_main);
52 26     setContentView(R.layout.activity_demo);
53
54 27     progressOne = (IconRoundCornerProgressBar) findViewById(R.id.progress_one);
55 28     progressOne.setProgressColor(
56 29         getResources().getColor(R.color.custom_progress_blue_progress),
57 30         getResources().getColor(R.color.custom_progress_blue_progress_half)
58 31     );
59 32     progressOne.setHeaderColor(getResources().getColor(R.color.custom_progress_blue)
60 33     );
61 34     progressOne.setBackgroundColor(getResources().getColor(R.color.custom_progress_
62 35     );
63 36     textProgressOne = (TextView) findViewById(R.id.text_progress_one);
```



# คำที่ต้องรู้จักใน GIT

## Unstaged และ Staged

ไฟล์ที่ถูกแก้ไขจะอยู่ในสถานะ **Unstaged** และเวลาที่ต้องการ Commit จะต้องเลือกไฟล์ที่ต้องการเพื่อย้ายเข้าสู่สถานะ **Staged** ก่อนถึงจะทำการ Commit ได้

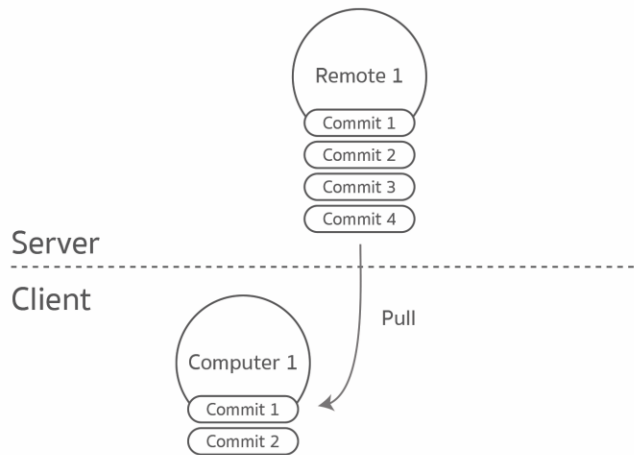




# คำที่ต้องรู้จักใน GIT

## Pull

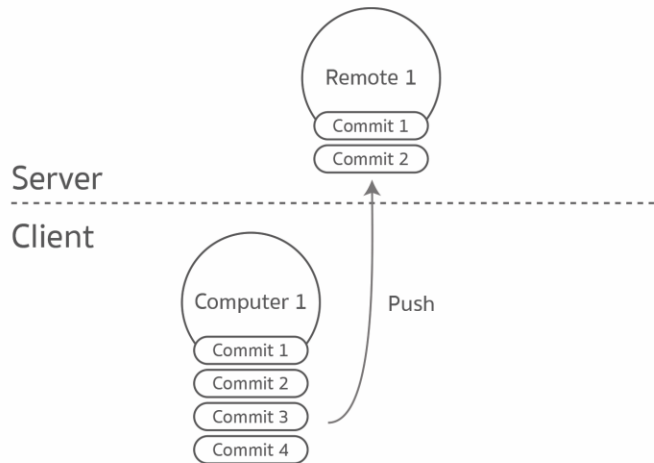
คือการ Sync จาก Server เพื่อดึงข้อมูล Commit ใหม่ ๆ ลงมาเก็บไว้ในเครื่อง



# คำที่ต้องรู้จักใน GIT

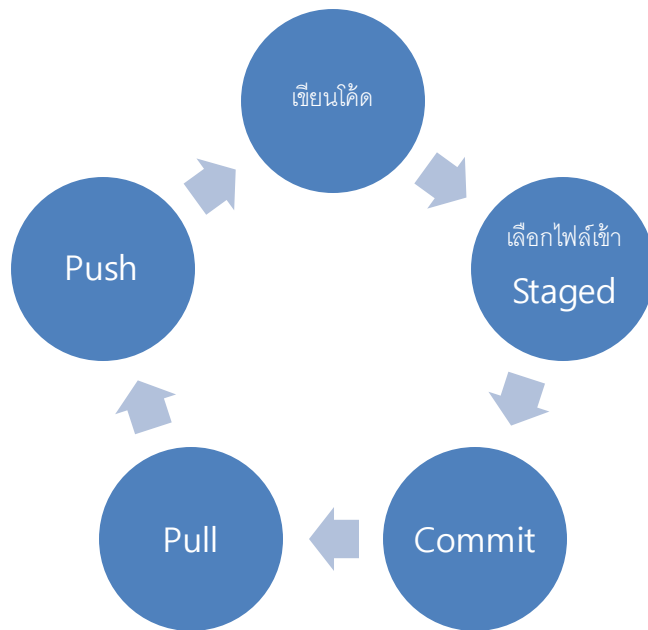
## Push

คือการ Sync commitdขึ้นไปเก็บไว้ใน Server



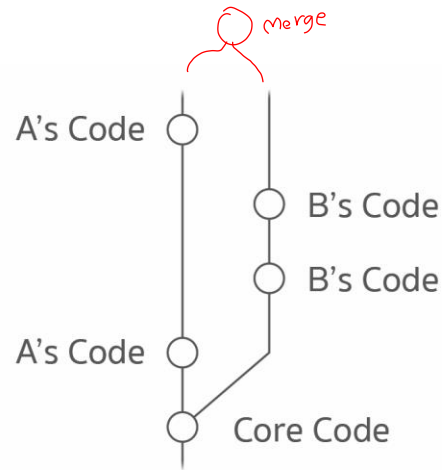
# GIT Loop

จากคำที่กล่าวไปในข้างต้นสามารถสรุปเป็นรูปได้ดังนี้



# Branch

คือ History ของการ Commit ในแต่ละครั้ง และนอกจากนี้ยังสามารถแตก Branch ออกมาได้หลายเส้นอีกด้วย



# GIT Flow

คือหนึ่งในมาตรฐานที่กำหนดรูปแบบของการแตก Branch ให้เป็นระบบระเบียบ เพื่อตอบโจทย์ใน Development Process โดยแบ่ง Branch ออกเป็น 3 กลุ่ม ใหญ่ๆดังนี้

- master
- develop
- feature





# Workshop

# สร้าง Repository

## 1. สร้าง Repository ใน GitLab หรือ GitHub โดยตั้งค่าให้เป็น Private

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner

Repository name \*

jirapatha

hdy-training

Great repository names are short and memorable. Need inspiration? How about **sturdy-system**?

Description (optional)

☐ Public

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

☒ Private

You choose who can see and commit to this repository.

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

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None

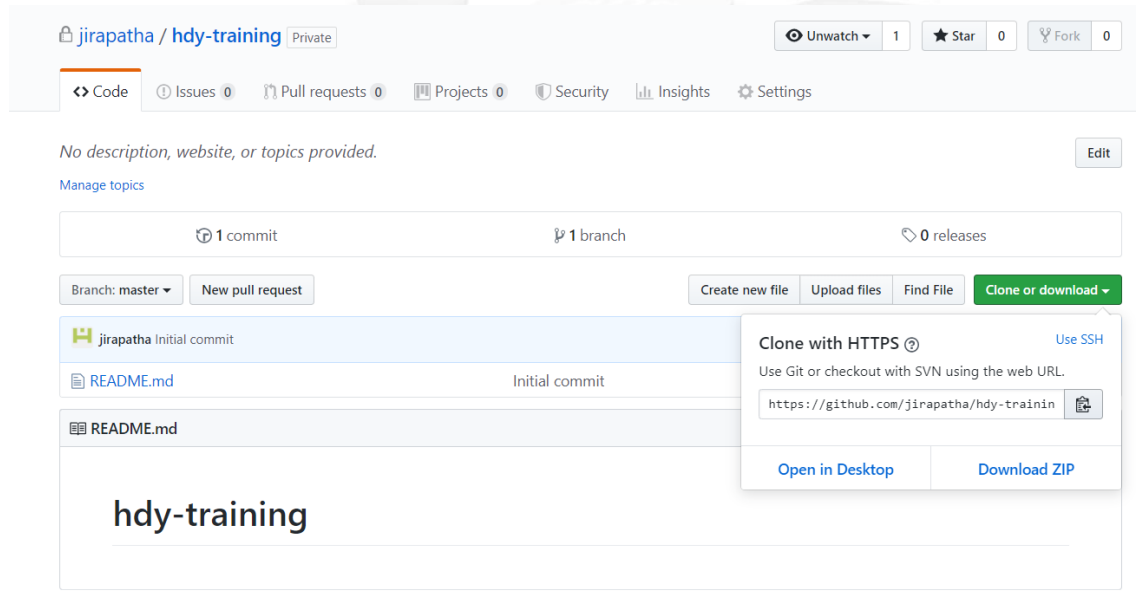
Add a license: None



Create repository

# สร้าง Repository

## 2. copy link httpsdไว้สำหรับ clonedลงในเครื่อง



The screenshot shows the GitHub interface for a repository named 'jirapatha/hdy-training'. The repository is marked as 'Private'. At the top, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Security', 'Insights', and 'Settings'. The main content area shows 'No description, website, or topics provided.' and 'Manage topics'. Below this, it indicates '1 commit', '1 branch', and '0 releases'. A 'Branch: master' dropdown and a 'New pull request' button are visible. A 'Clone or download' button is highlighted, and its dropdown menu is open, showing options to 'Clone with HTTPS' (with a link to 'https://github.com/jirapatha/hdy-training'), 'Use SSH', 'Open in Desktop', and 'Download ZIP'. The repository's README is visible at the bottom, showing the title 'hdy-training'.



# Sourcetree (GUI)

1. โหลดและติดตั้ง Sourcetree จาก <https://www.sourcetreeapp.com/>
2. เชื่อม Account GitLab หรือ GitHub กับ Sourcetree

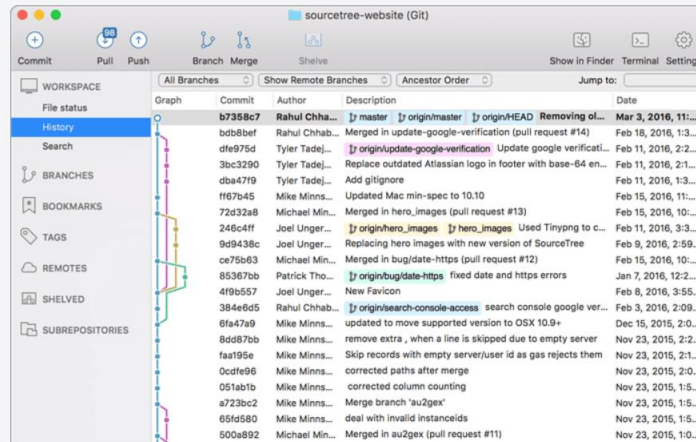
 Sourcetree

Download free

Simplicity and power in  
a beautiful Git GUI

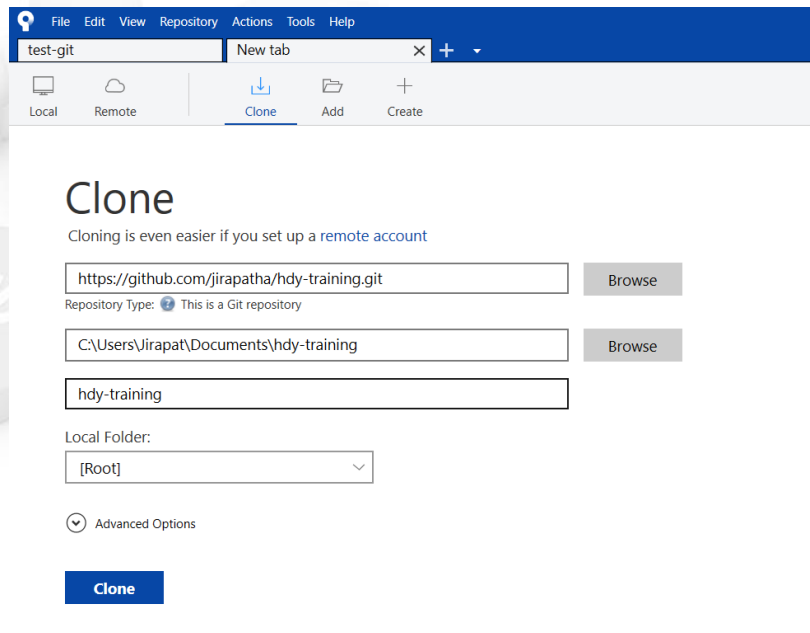
Download for Windows

Also available for Mac OS X



# Sourcetree (GUI)

## 3. Clone Repository ที่สร้างไว้



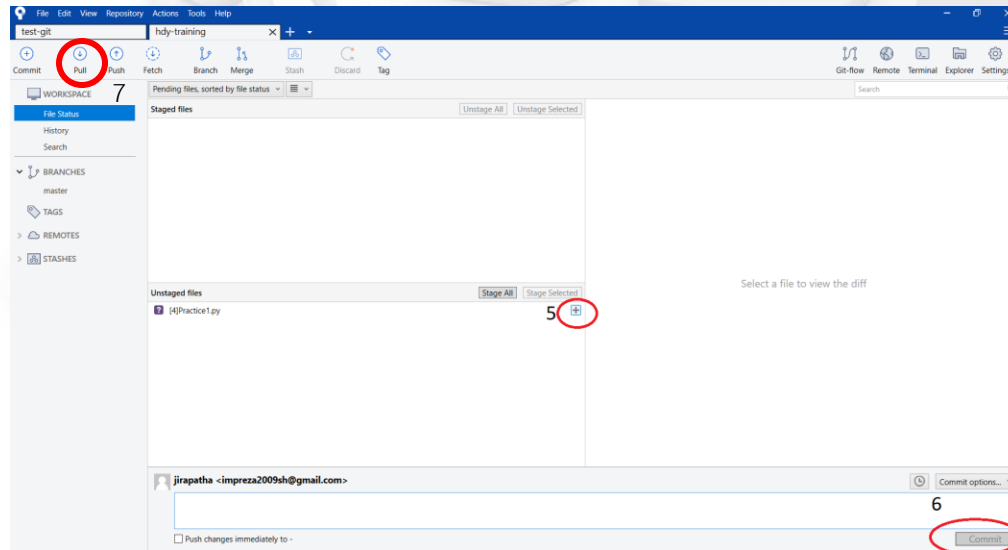
# Sourcetree (GUI)

## 4. เพิ่มไฟล์เข้าไปใน Folder ที่ Clone ไว้

Name	Date modified	Type	Size
[4]Practice1	1/3/2562 13:39	PY File	1 KB
README.md	5/9/2562 23:10	MD File	1 KB

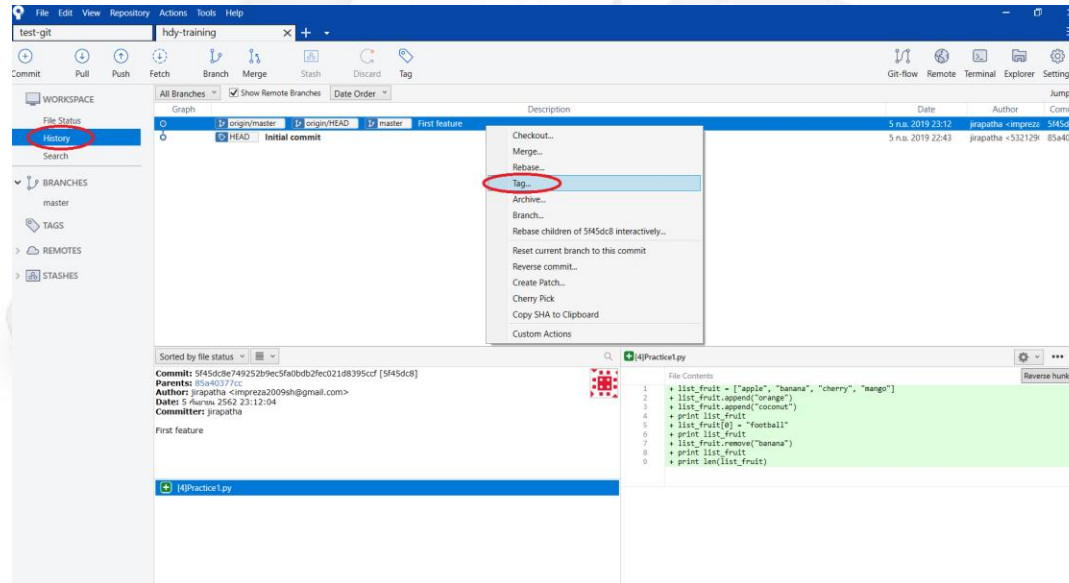
# Sourcetree (GUI)

5. เพิ่มไฟล์เข้า staged
6. ทำการ Commit
7. ทำการ Pull



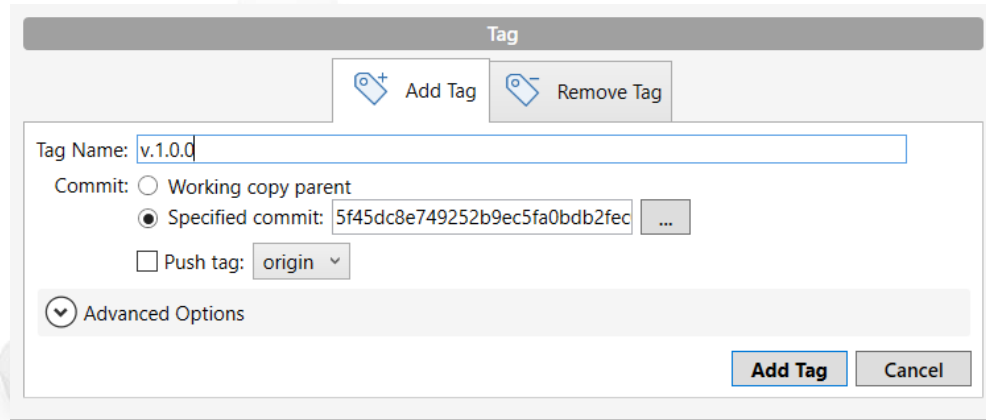
# Sourcetree (GUI)

## 8. ทำการใส่ Tag version



# Sourcetree (GUI)

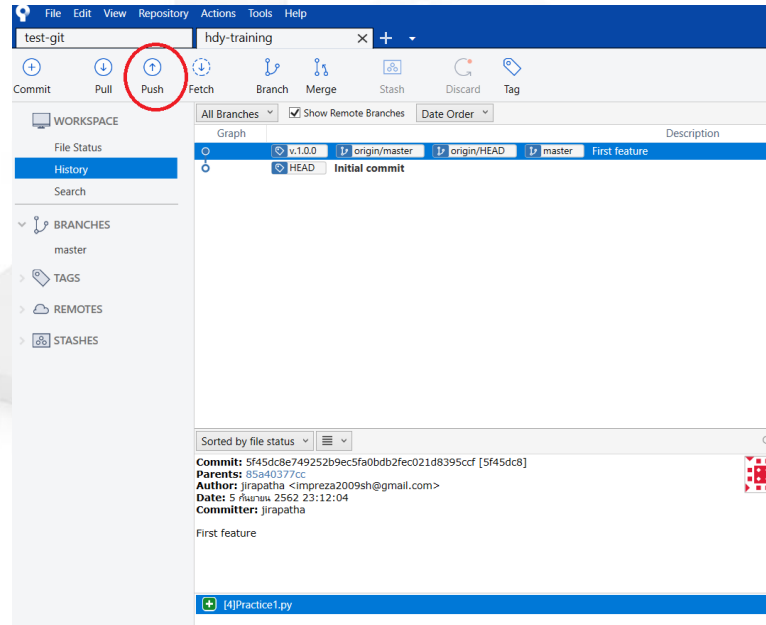
หลักการใช้ version



หลักที่ 1 version ของโครงสร้าง code  
หลักที่ 2 version ของ feature  
หลักที่ 3 version ของ fix bugs

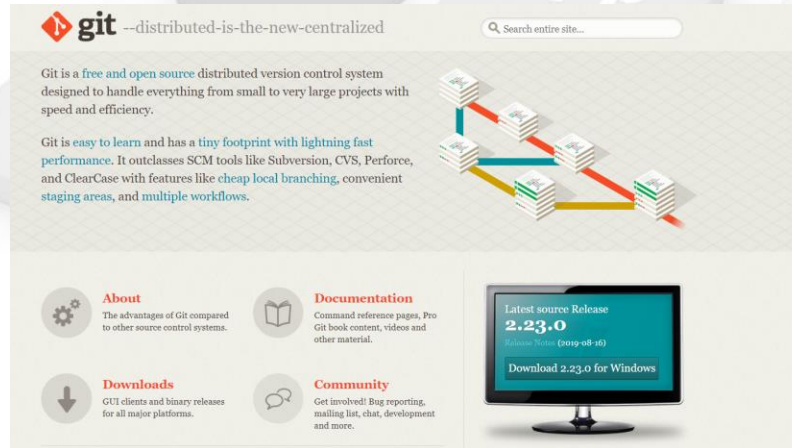
# Sourcetree (GUI)

## 9. ทำการ Push



# GIT SCM (Command Line)

1. โหลดและติดตั้ง GIT SCM จาก <https://git-scm.com/>
2. สร้าง folder ใหม่





# GIT SCM (Command Line)

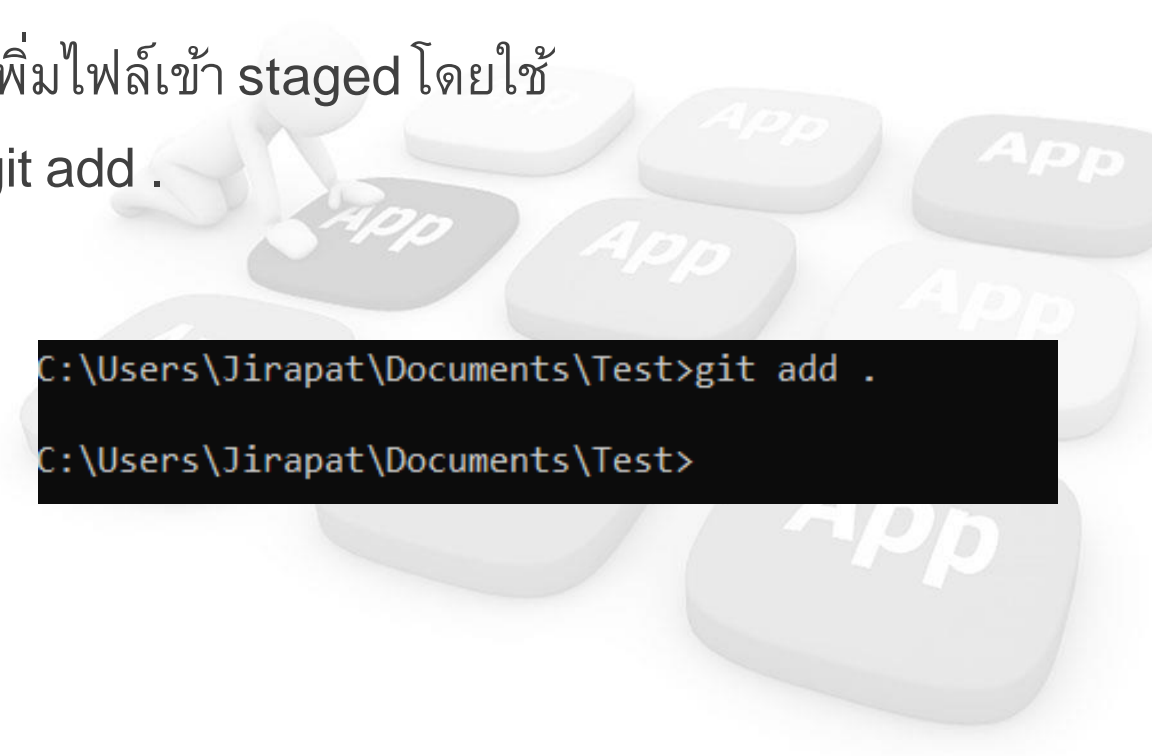
3. Clone Repository ไปยัง Folder ที่สร้างไว้ดโดยใช้  
`git clone <https link> <destination folder>`

```
C:\Users\Jirapat\Documents>git clone https://github.com/jirapatha/hdy-training.git Test
Cloning into 'Test'...
Fatal: HttpRequestException encountered.
Username for 'https://github.com': jirapatha
Password for 'https://jirapatha@github.com':
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.
```

# GIT SCM (Command Line)

4. เพิ่มไฟล์ใน Folder ที่ Clone ไว้

5. เพิ่มไฟล์เข้า staged โดยใช้  
git add .



```
C:\Users\Jirapat\Documents\Test>git add .  
C:\Users\Jirapat\Documents\Test>
```

# GIT SCM (Command Line)

6. ทำการ commit โดยใช้

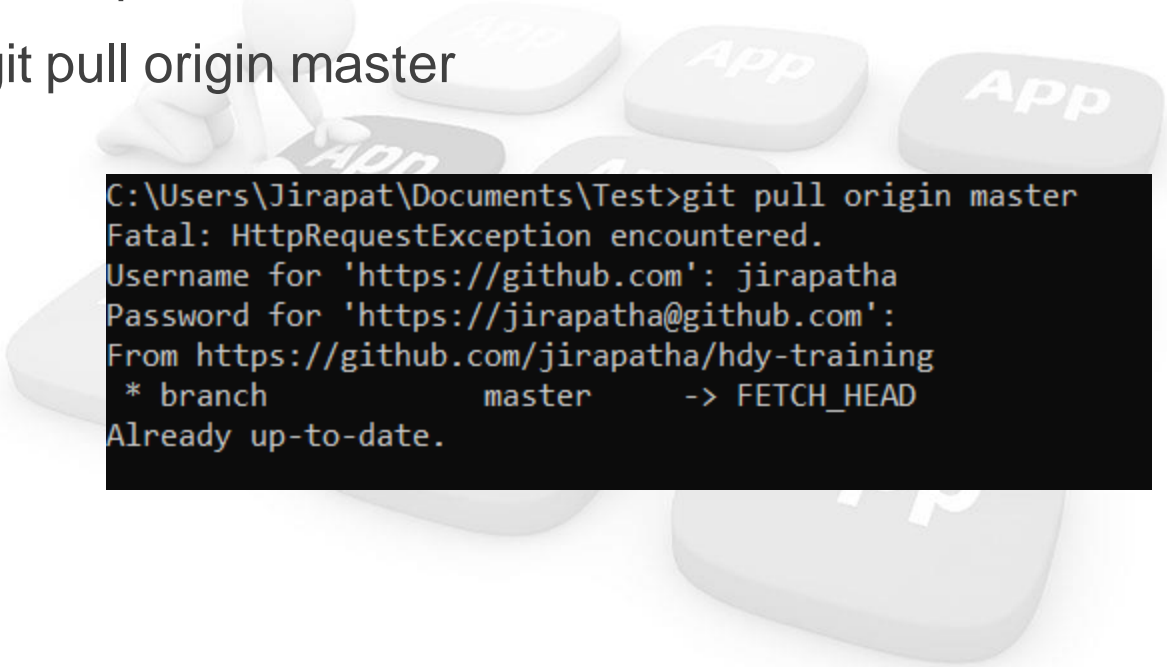
`git commit -m "comment"`

```
C:\Users\Jirapat\Documents\Test>git commit -m "Second feature"
[master 9b3edf4] Second feature
1 file changed, 8 insertions(+)
create mode 100644 [4]Practice2.py
```

# GIT SCM (Command Line)

7. ทำการ pull โดยใช้

`git pull origin master`



```
C:\Users\Jirapat\Documents\Test>git pull origin master
Fatal: HttpRequestException encountered.
Username for 'https://github.com': jirapatha
Password for 'https://jirapatha@github.com':
From https://github.com/jirapatha/hdy-training
 * branch          master      -> FETCH_HEAD
Already up-to-date.
```

# GIT SCM (Command Line)

## 8. ทำการ push

git push

```
C:\Users\Jirapat\Documents\Test>git push
Fatal: HttpRequestException encountered.
Username for 'https://github.com': jirapatha
Password for 'https://jirapatha@github.com':
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 411 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/jirapatha/hdy-training.git
5f45dc8..9b3edf4 master -> master
```



# Thank you