

# Session1 : Git알아보기

부제 : 지옥에서 올라온 Git

# 목차

1. Git이란?
2. 왜 Git을 써야 하는가?
3. Git 주요 명령어
4. 실습
5. Q&A

**Git이란?**

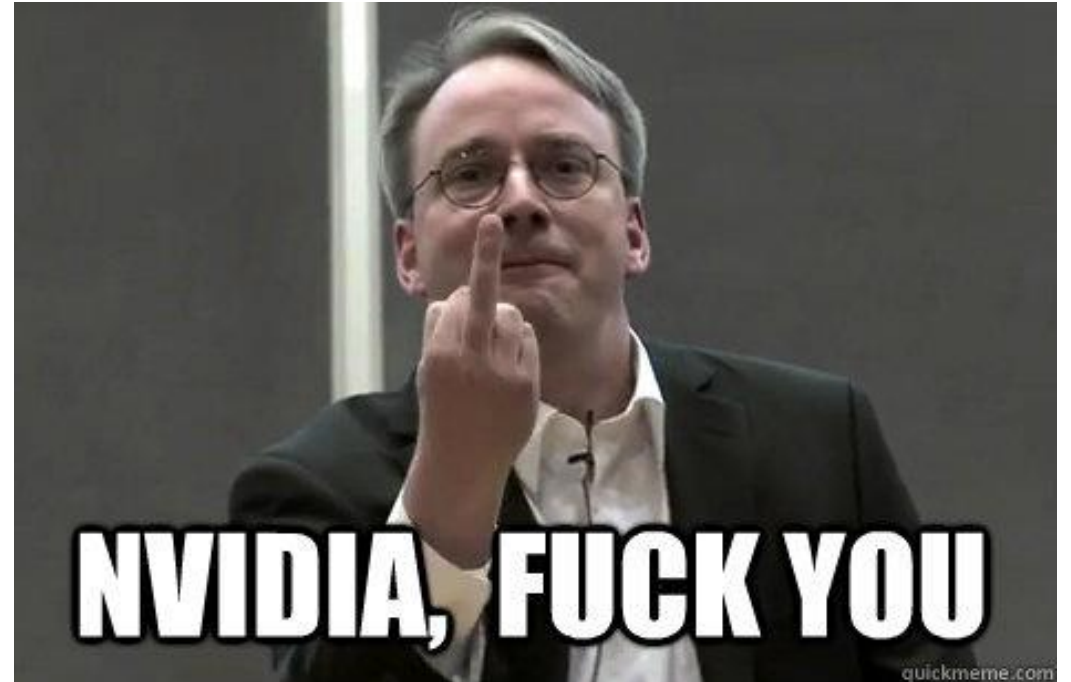
# Git이란?

리누스 토르발스가 개발한 분산형 버전 관리 시스템(VCS)

속도가 다른 버전 관리 시스템에 비해 빠르고  
분산형 저장소를 지원한다

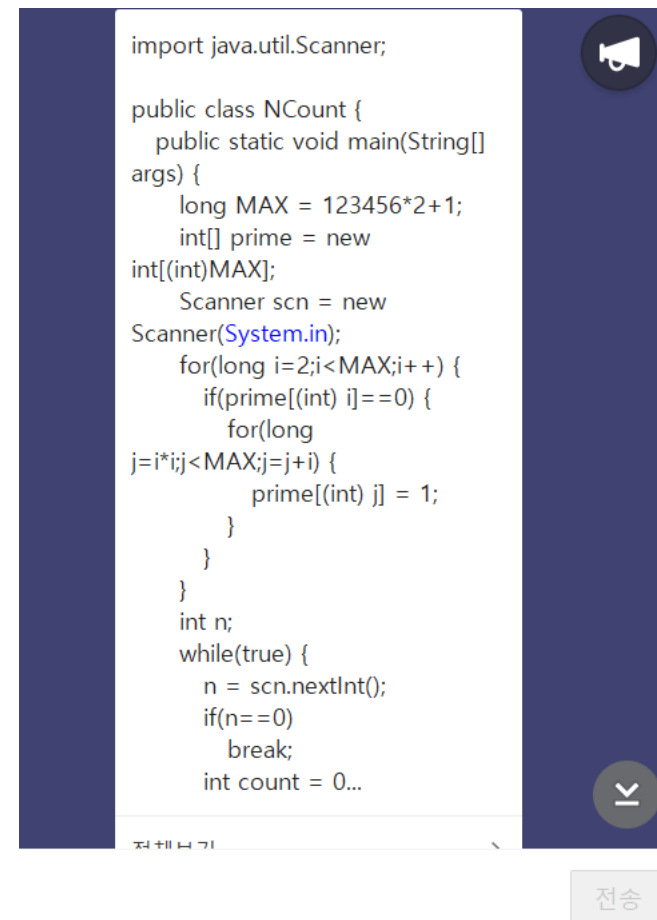
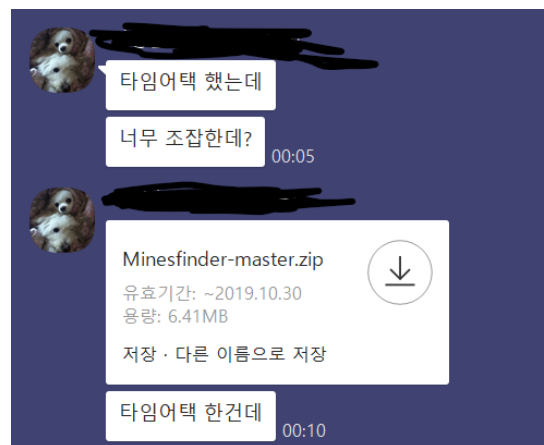
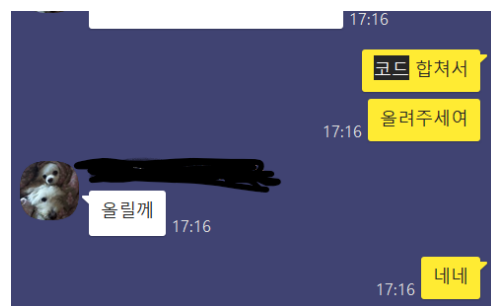
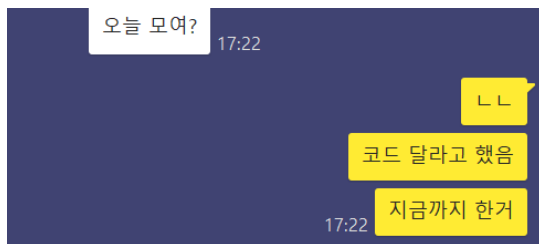
현재 Github가 가장 큰 호스팅 업체이며, bitbucket,  
Gitlab등의 다른 호스팅 사이트들도 존재한다.

Github는 개발 코드를 넘어서 현재 이력서 처럼 쓰이는중  
이다.



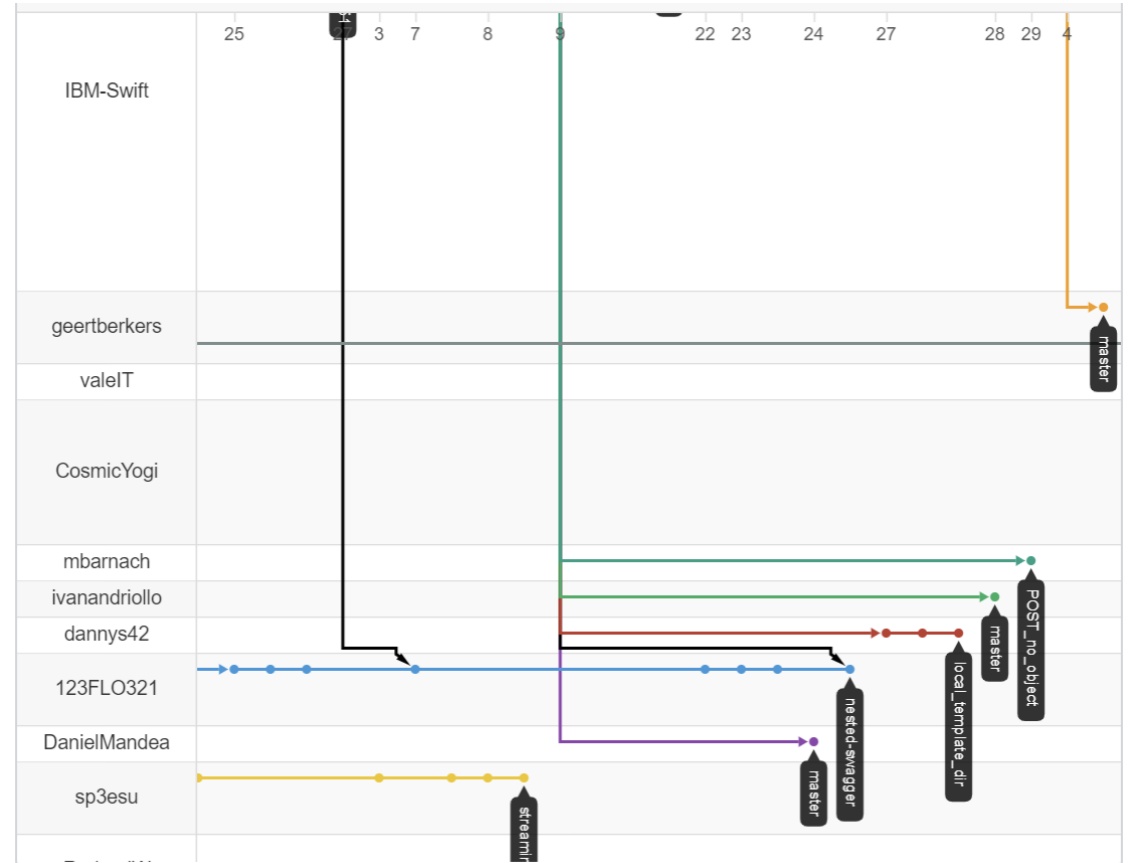
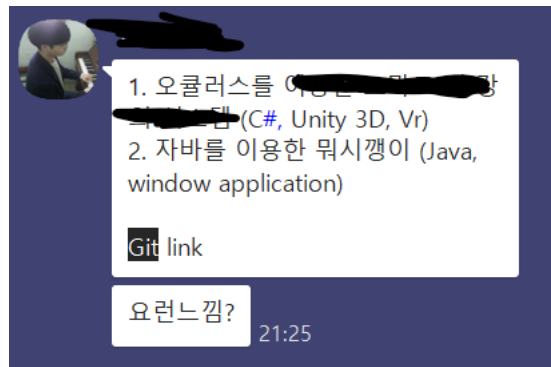
**왜 Git을 써야하는가?**

# 왜 Git을 써야 하는가?



우리가 프로젝트를 하면 맨날 보는 상황

# 왜 Git을 써야 하는가?



실제로 원하는 코드 설명

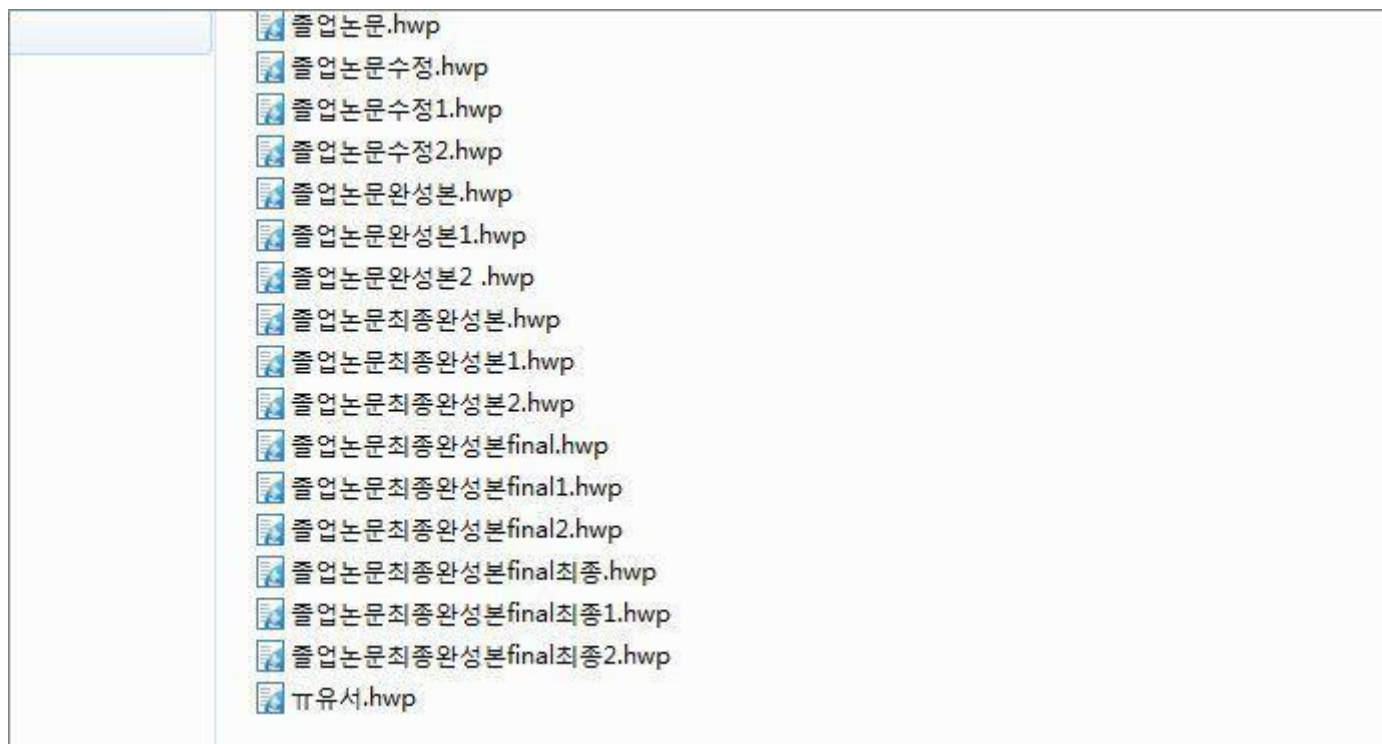
# 왜 Git을 써야 하는가?

버전관리를 함으로써 변경점 확인, 백업&복구,  
협업 등을 할 수 있음

Commits on Nov 4, 2019	<div>Add Nested Types support to Swagger Generator (#1488) 123FLO321 authored and djones6 committed 10 days ago ✓</div> <div>bbb7462</div>
Commits on Oct 9, 2019	<div>feat: Ability to limit request size and connection count (#1481) djones6 committed on 9 Oct ✓</div> <div>Verified728acab</div>
Commits on Sep 27, 2019	<div>ci: Update to Swift 5.1 djones6 committed on 27 Sep ✓</div> <div>5c3589f</div>
Commits on Sep 18, 2019	<div>tests: Remove workaround for SR-11012 (#1486) djones6 committed on 18 Sep ✗</div> <div>Verified093fbe4</div> <div>feat: Allow customization of the 'SwaggerDocument' (#1483) ... mbarnach authored and djones6 committed on 18 Sep ✓</div> <div>0bab2d2</div>
Commits on Sep 6, 2019	<div>fix: Correctly register Codable routes with trailing `:id` (#1484) ... djones6 committed on 6 Sep ✗</div> <div>Verified0777165</div>
Commits on Aug 13, 2019	<div>chore: Generate API docs for Kitura 2.8 (#1479) djones6 committed on 13 Aug ✓</div> <div>Verified788b4ec</div>
Commits on Aug 7, 2019	<div>ci: Update to Swift 5.0.2 djones6 committed on 7 Aug ✓</div> <div>6595bc9</div>
Commits on Aug 2, 2019	<div>feat: Convenience API to create and add cookies. (#1468) RudraniW authored and djones6 committed on 2 Aug ✓</div> <div>9f9e4ba</div>
Commits on Aug 1, 2019	<div>fix: Add ios availability (#1475) makleso6 authored and djones6 committed on 1 Aug ✓</div> <div>ab7a1d3</div>



# 왜 Git을 써야 하는가?



버전관리 시스템은 의외로 중요하다.

# 왜 Git을 써야 하는가?



버전관리 시스템은 의외로 중요하다.

# Git 주요 명령어

# Git 주요 명령어

저거 말고도 더 명령어가 많다!

```
PS C:\Users\NamYounSu> git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  commit     Record changes to the repository
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  reset      Reset current HEAD to the specified state
  switch     Switch branches
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

# Git 주요 명령어

6개만 기억하자,

Git init

Git add

Git commit

Git remote

Git push

Git pull

Git clone

# Git 주요 명령어

## Git init

Git 저장소를 새로 생성하는 명령어

반드시 “빈” 폴더에서 명령어를 입력해야 오류가 나지 않는다. (빈 폴더의 기준은 .git 폴더가 없는 폴더)

쓰는 예 ) git init

# Git 주요 명령어

## Git add

Git 저장소에 파일을 지켜보게 한다(추가가 아니다!)

Git은 파일을 지켜보며 변경사항이 발생하면 이를 기록한다.

쓰는 예 ) git add \*  
          git add HelloWorld.c

# Git 주요 명령어

## Git commit

Git 저장소에 변경사항의 “스냅샷” 을 만든다.

Git은 이전에 add한 파일들 중 변경사항이 발생한 파일의 정보를 저장하고, “스냅샷” 을 생성한다

쓰는 예 ) `git commit -m "initial commit"`  
`git commit -m "fix: 이 커밋 이후로 퇴사합니다"`



# Git 주요 명령어

## Git remote

Git 저장소에 원격 저장소를 추가/삭제 한다

지금까지 생성 한 Git은 내부저장소에 저장 했으므로, 저 명령어를 통해 원격 저장소와 연결한다.

쓰는 예 ) `git remote add origin https://github.com/NamSu/namsu.github.io.git`

# Git 주요 명령어

## Git push

현재까지의 커밋 내역을 원격 저장소를 추가

지금까지 생성 한 커밋은 내부저장소에 저장 했으므로, 저 명령어를 통해 원격 저장소에 커밋 내역을 추가할 수 있다.

쓰는 예 ) `git push origin master`

# Git 주요 명령어

## Git pull

원격 저장소의 커밋 내역을 내부 저장소에 추가

원격 저장소의 새버전을 원하면, 저 명령어를 통해 원격 저장소의 커밋을 내부 저장소에 추가할 수 있다.

쓰는 예 ) `git pull origin master`

# Git 주요 명령어

## Git clone

원격 저장소의 내용을 복제한다

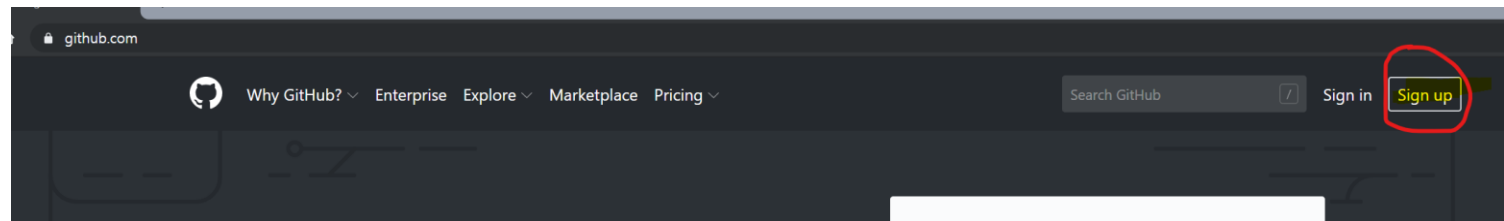
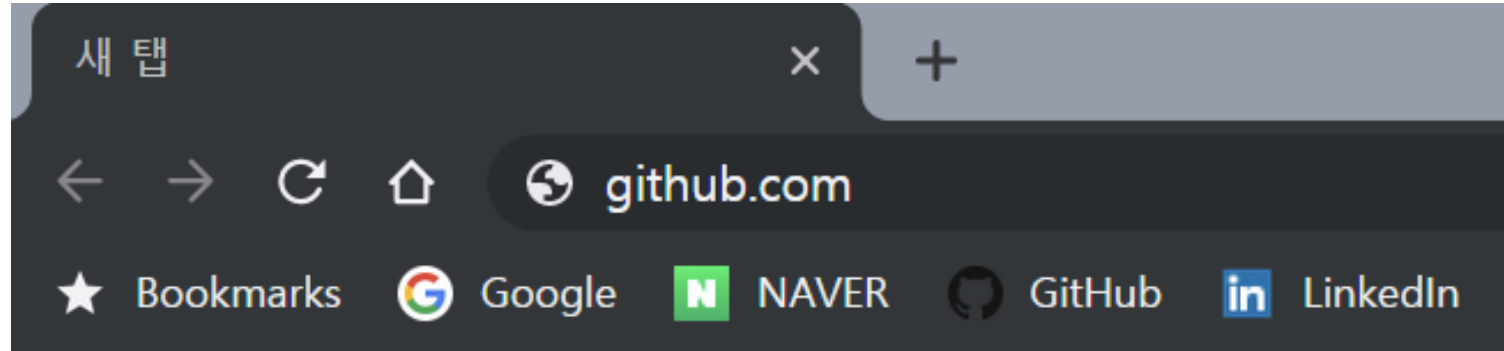
원격 저장소의 내용 그대로 현재 디렉토리에 복제한다

쓰는 예 ) `git clone https://github.com/NamSu/namsu.github.io.git`

실습

# 실습

1. Github에 접속해서 계정을 만든다.  
계정이 있으면 4번으로 가자.



<https://github.com>

# 실습

2. Username, Email, Password를 입력하고  
Next버튼을 클릭한다.

Username은 닉네임이나 아이디로 설정하면 편하다.  
ex) Namsu

Join GitHub

## Create your account

Username \*

nvidiafuckyou ✓

Email address \*

nvidiafuckyou@gmail.com ✓

Password \*

.....

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.  
[Learn more.](#)

Email preferences

☒ Send me occasional product updates, announcements, and offers.

Verify your account

당신이 진짜 사람인지 알 수 있도록 이  
퍼즐을 풀어주세요

검증하기

🔊

Next: Select a plan

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.

# 실습


3. Free Plan을 선택하고 Email 인증을 진행한다.

안 왔으면 스팸 메일함을 보자.

Choose a plan  
Pick the plan that's right for you

Individuals


Teams



Free  
\$0<sub>USD</sub>

The basics of GitHub for every developer

Choose Free



Pro  
\$7<sub>USD</sub>  
Per month

Pro tools for developers with advanced requirements

Choose Pro

- ✓ Unlimited public repositories
- ✓ Unlimited private repositories
- ✓ Limited to 3 collaborators for private repositories
- ✓ 2,000 total Action minutes/month  
See pricing details
- ✓ 500MB of GitHub Packages storage  
See pricing details
- ✓ Advanced vulnerability scanning for public repositories
- ✓ Automated security updates
- ✓ GitHub Security Advisories
- ✓ Issues and bug tracking
- ✓ Project management

← Includes everything in Free

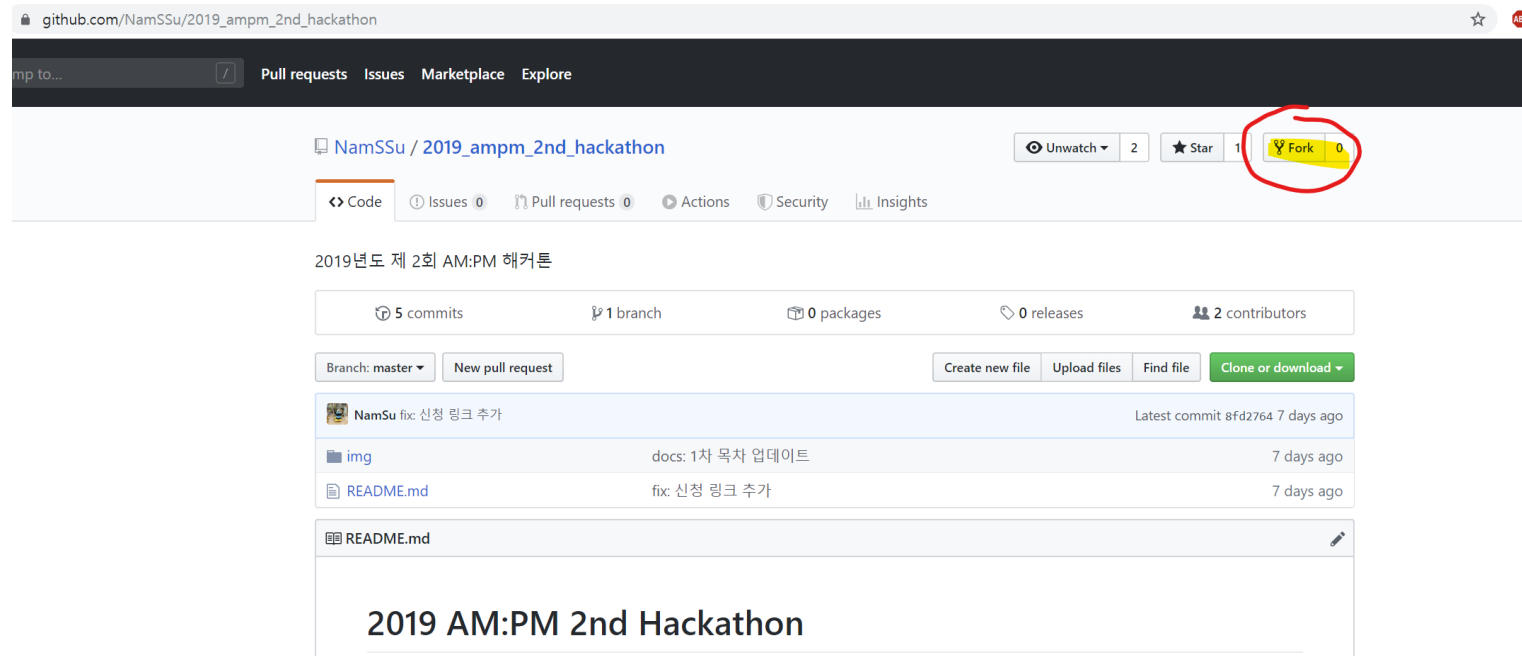
- ✓ Unlimited collaborators
- ↑ 3,000 total Action minutes/month  
See pricing details
- ✓ 500MB of GitHub Packages storage  
See pricing details
- ✓ Private GitHub Pages and Wikis
- ✓ Private protected branches
- ✓ Code owners
- ✓ Repository insights



# 실습

4. 해커톤 리포지토리에 접속한다.

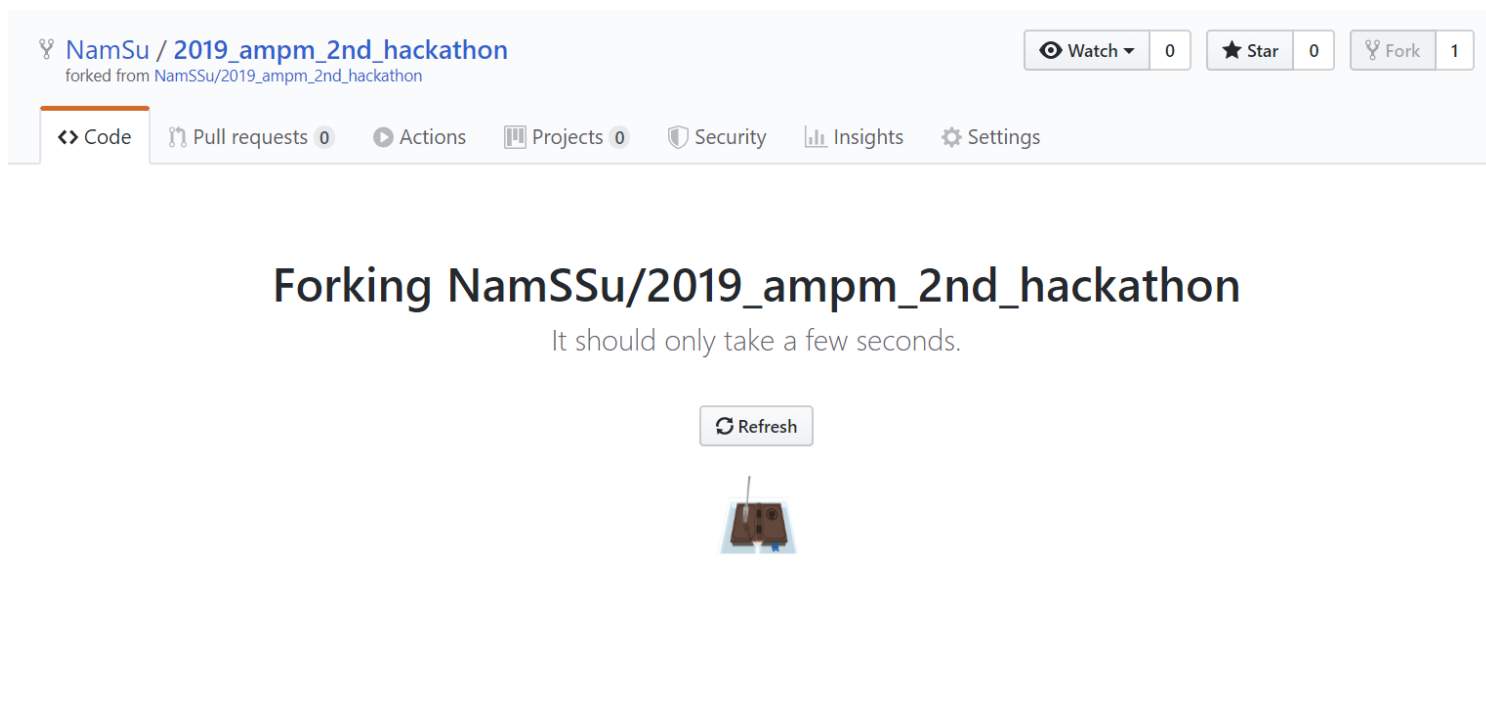
접속 후 옆의 Fork버튼을 클릭한다.



[https://github.com/NamSSu/2019\\_ampm\\_2nd\\_hackathon](https://github.com/NamSSu/2019_ampm_2nd_hackathon)

# 실습

5. Fork 되면 nickname/repo name  
이런 형식으로 뜰 것이다.



# 실습

## Git 설치

6. git-scm에 접속해 Git 클라이언트를 다운로드한다.

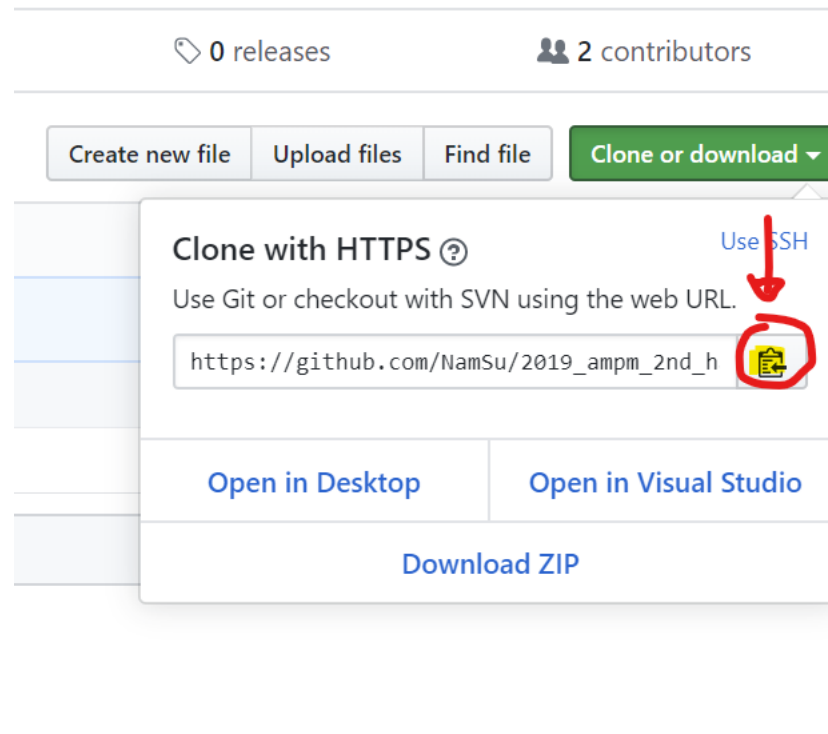
설치는 다음다음다음 완료만 누르면 된다.



<https://git-scm.com/download>

# 실습

7. 우측의 초록버튼(Clone or Download)를 클릭해 URL을 복사한다.



# 실습

8. Git bash 를 열어 우측과 같이 입력한다

 Windows PowerShell

```
PS C:\Users\NamYounSu\Desktop> git clone https://github.com/NamSu/2019_ampm_2nd_hackathon.git
```

# 실습

9. 완료되면 cd 명령어를 사용해 폴더 안으로 들어간다.

 Windows PowerShell

```
PS C:\Users\NamyounSu\Desktop> cd .\2019_ampm_2nd_hackathon\  
PS C:\Users\NamyounSu\Desktop\2019_ampm_2nd_hackathon>
```

# 실습

10. 우측과 같이 git 세팅을 진행한다.

이는 자신의 커밋 내역에 남는 내용이다.

User.name에는 github계정 이름을,  
User.email에는 github계정 이메일을  
적는다.

Windows PowerShell

```
PS C:\Users\NamYounSu\Desktop> cd .\2019_ampm_2nd_hackathon\  
PS C:\Users\NamYounSu\Desktop\2019_ampm_2nd_hackathon> git config --global user.name "NamSu"
```

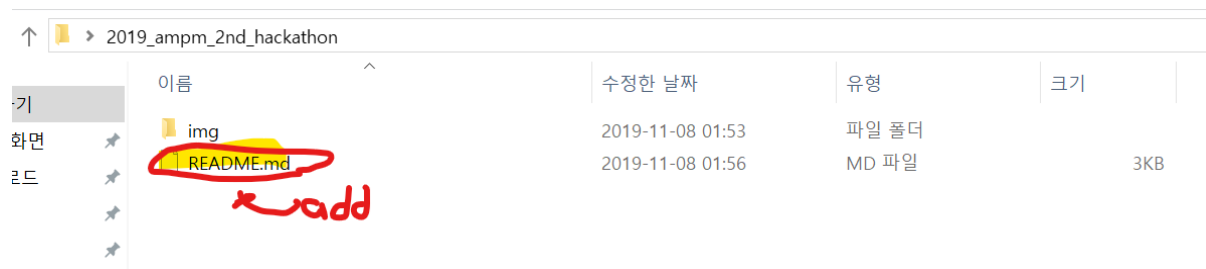
Windows PowerShell

```
PS C:\Users\NamYounSu\Desktop> cd .\2019_ampm_2nd_hackathon\  
PS C:\Users\NamYounSu\Desktop\2019_ampm_2nd_hackathon> git config --global user.email "nys6635@gmail.com"
```

# 실습

11. 파일을 추가하고 우측과 같이 add를 진행한다.

전체를 추가하거나 특정 파일만 추가 하는건 선택이다.



	이름	수정한 날짜	유형	크기
기	img	2019-11-08 01:53	파일 폴더	
화면	README.md	2019-11-08 01:56	MD 파일	3KB
드				

```
Windows PowerShell
PS C:\Users\NamYounSu\Desktop> cd .\2019_ampm_2nd_hackathon\
PS C:\Users\NamYounSu\Desktop\2019_ampm_2nd_hackathon> git add *
```

```
Windows PowerShell
PS C:\Users\NamYounSu\Desktop> cd .\2019_ampm_2nd_hackathon\
PS C:\Users\NamYounSu\Desktop\2019_ampm_2nd_hackathon> git add .\README.md
```



# 실습

12. 우측과 같이 commit을 진행한다.

 Windows PowerShell

```
PS C:\Users\NamYounSu\Desktop> cd .\2019_ampm_2nd_hackathon\  
PS C:\Users\NamYounSu\Desktop\2019_ampm_2nd_hackathon> git commit -m "add: README.md"
```

# 실습

13. 우측과 같이 원격 저장소에 push한다.





Remote과정은 생략한다(이미 연결됨)

 Windows PowerShell

```
PS C:\Users\NamYounSu\Desktop> cd .\2019_ampm_2nd_hackathon\  
PS C:\Users\NamYounSu\Desktop\2019_ampm_2nd_hackathon> git push origin master
```

# 실습

14. Github에 접속해서 커밋이 되었는지 확인한다.

Commits on Nov 8, 2019		
fix: 신청 링크 추가 NamSu committed 7 days ago	 8fd2764	
fix: 목차 네비게이션 bonomoon committed 7 days ago	 f6136f4	

# Q&A



**감사합니다**