Git / GitHub

입 문

Git vs GitHub

Git 개요

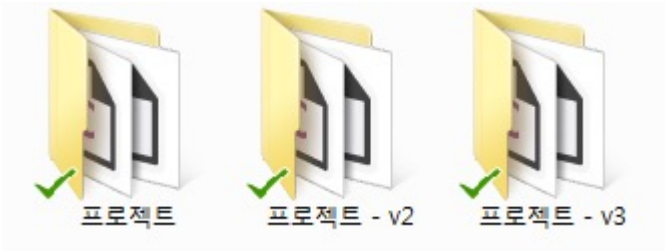
Git (깃) ? 분산 버전 제어 SW

리누즈 토발츠
Father of Linux

Git 개요

프로잭트에도 버전이 있지요 ?

Git 개요



…

Git 개요

아놔... 이것 좀 관리해주는 것 없을까?

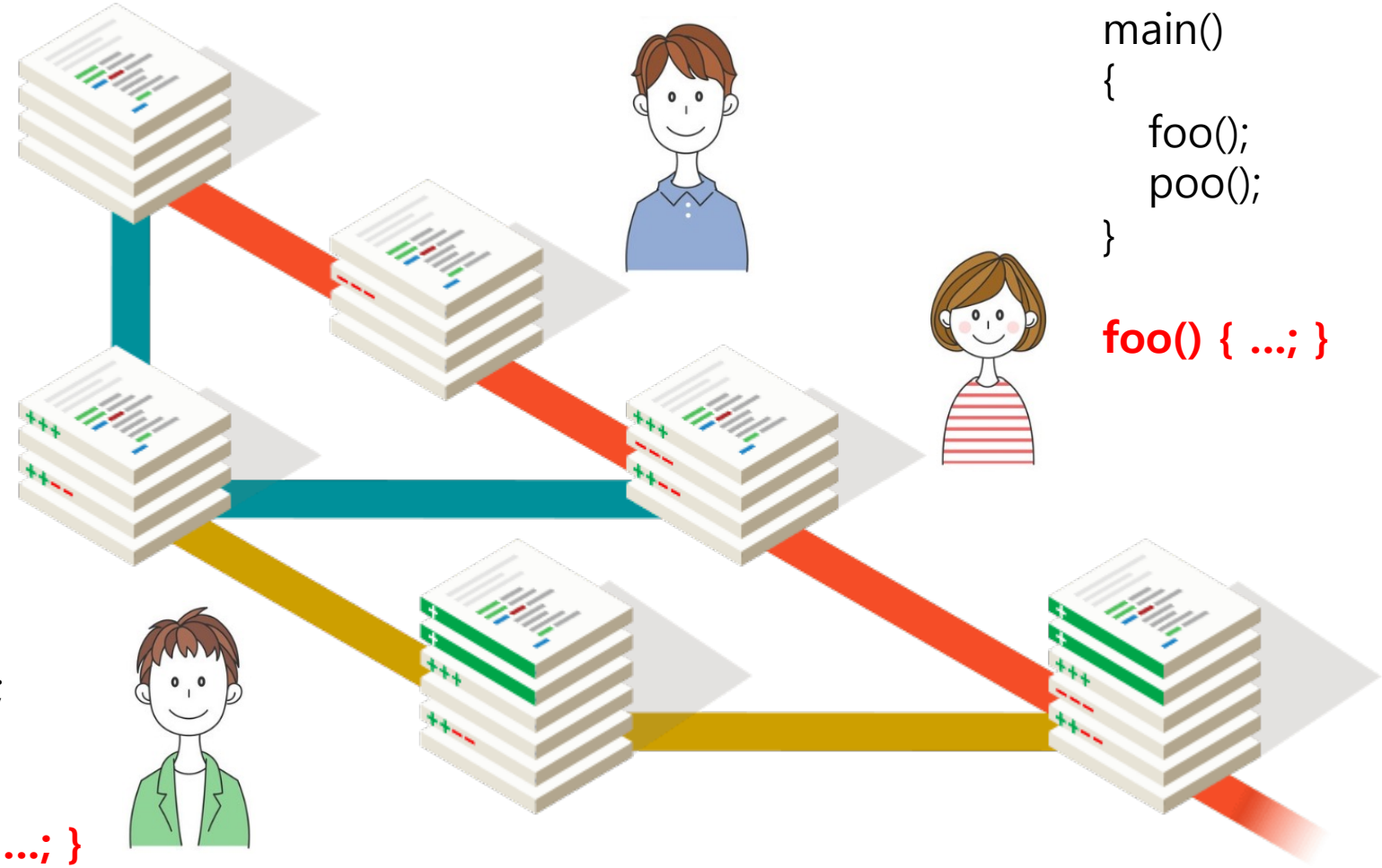
GitHub 개요

협업도 지원해줘요!

GitHub 개요

main()
{
    foo();
    poo();
}

협업도 지원해줘요!



main()
{
    foo();
    poo();
}

foo() { …; }

main()
{
    foo();
    poo();
}

poo() { …; }

main()
{
    foo();
    poo();
}

foo() { …; }
poo() { …; }

GitHub 개요

그리고 오픈소스 SW !!!

오픈하면 공짜! ㅋㅋ

GitHub

Github (깃허브) ?

Git을 위한 웹 저장소(Repository)

+ 커뮤니티 협의공간

MS, 8조원에 깃허브 인수

공유 74 댓글 0

언어 선택 Google 번역에서 제공

마이크로소프트(MS)가 '깃허브' 인수를 공식 발표했다. 인수 금액은 75억달러, 우리돈 8조원 규모다. MS는 75억달러에 이르는 주식을 주고 깃허브를 인수한다. 인수 후 신임 CEO는 자마란 설립자이자 MS 법인 부사장을 맡고 있는 냇 프리드먼이 맡는다. 깃허브의 현재 CEO인 크리스 원스트러스는 MS 기술 펠로우로 전략 SW 사업 부문을 맡게 된다.

깃허브는 세계 최대 소프트웨어 개발 플랫폼이다. 리눅스 개발자 리누스 토발즈가 만든 분산형 버전관리 도구 '깃'을 호스팅하는 서비스로 출발했다. 현재 2800만명 이상의 개발자가 8천개 넘는 소스코드 저장소를 운영하고 협업하는 개발자들의 놀이터로 성장했다.

Git vs GitHub

Git (깃) ?

분산 버전 제어 SW

Github (깃허브) ?

Git을 위한 웹 저장소(Repository)

+ 커뮤니티 협의공간

Git vs GitHub

- **Git은 버전 관리 시스템 : VCS** (Version control system)
  - 버전 관리란?
    - 버전 관리 시스템은 파일의 변화를 시간에 따라 기록하여 과거 특정 시점의 버전을 다시 불러올 수 있는 시스템이다. 소스 코드를 버전 관리하는 예를 들지만 실제로는 모든 컴퓨터 파일이 버전 관리의 대상이 될 수 있다.
- **GitHub**
  - GitHub은 Git repository를 위한 호스팅 플랫폼이다.
  - 만약 GitHub 이 없다면 서버를 따로 만들어서 올려야 하는 것이다. 이렇게 되면 다른 개발자와 협업을 하기가 어려워지고 코드 공유도 쉽지 않아진다.

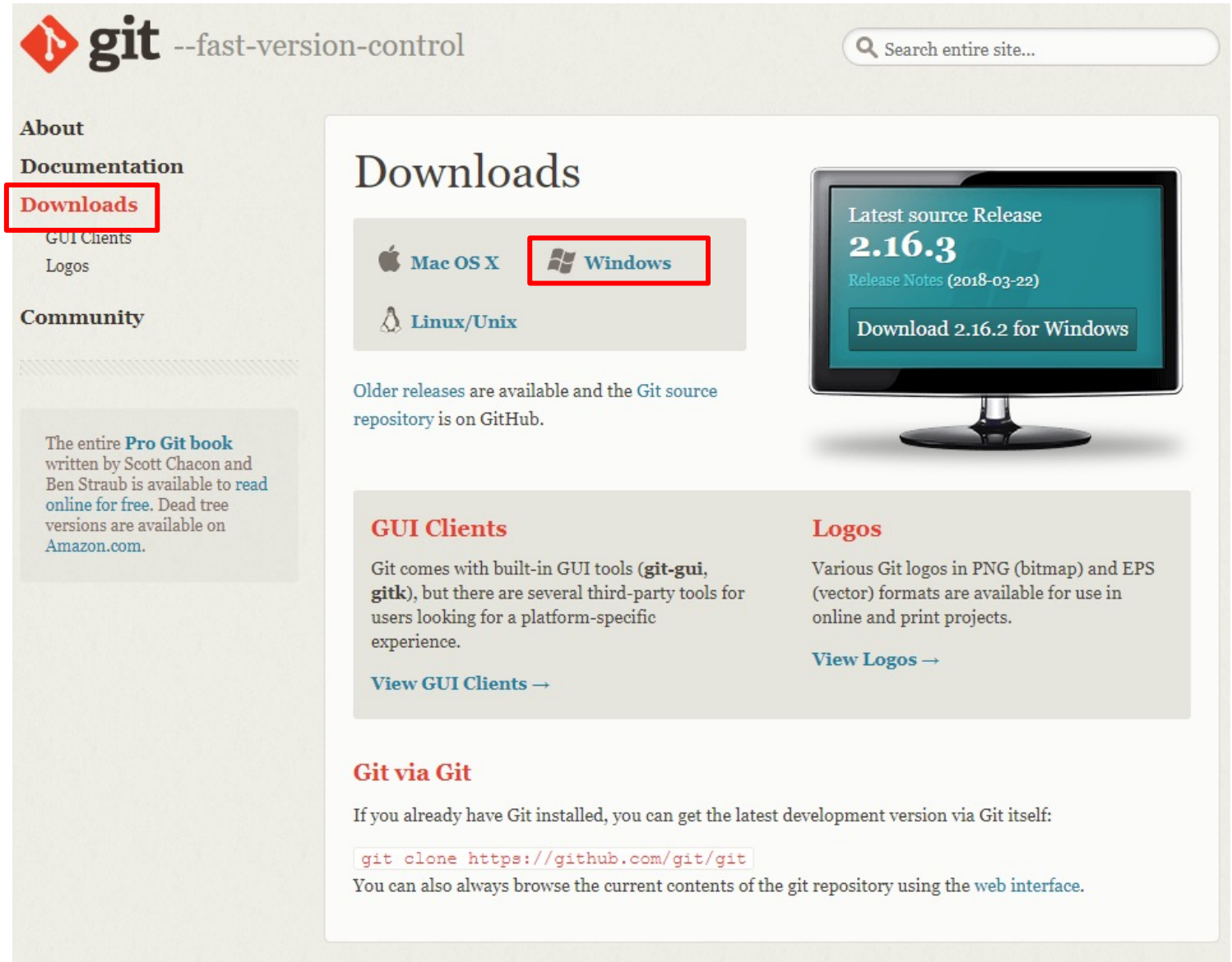
설치

- **1. Git 설치 (TUI)**
  - http://git-scm.com/

- **2. GitHub 회원가입(Romote Repository)**
  - https://github.com/

- **3. 소스트리 설치 (GUI)**
  - https://www.sourcetreeapp.com
  - 설치시 회원가입 필요

1. Git 설치

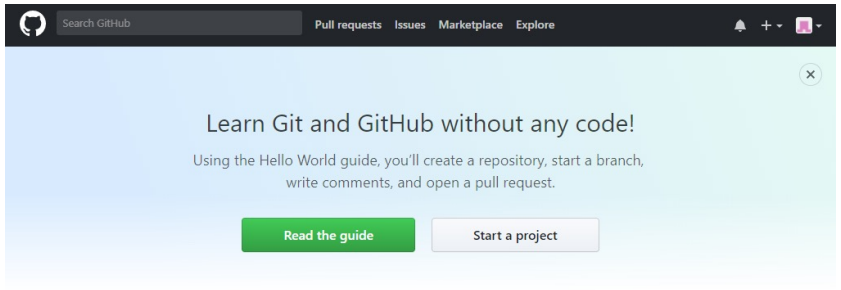


- **1. 구글에서 “Git " 검색**
  - Git 페이지에서 Git 다운로드 및 설치 진행
- **2. URL**
  - http://git-scm.com/

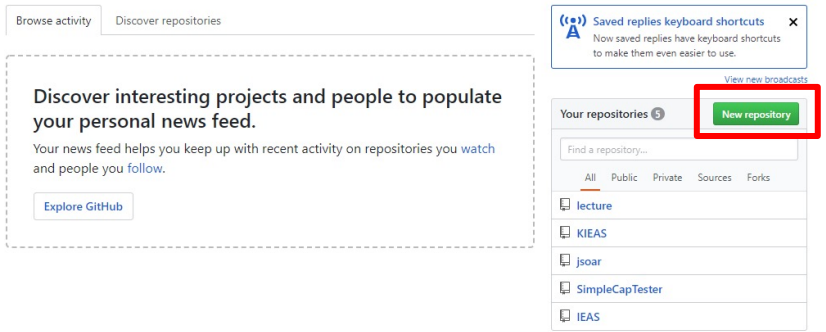
2. GitHub 가입

- https://github.com/
  - 위의 주소에서 깃허브 계정 생성

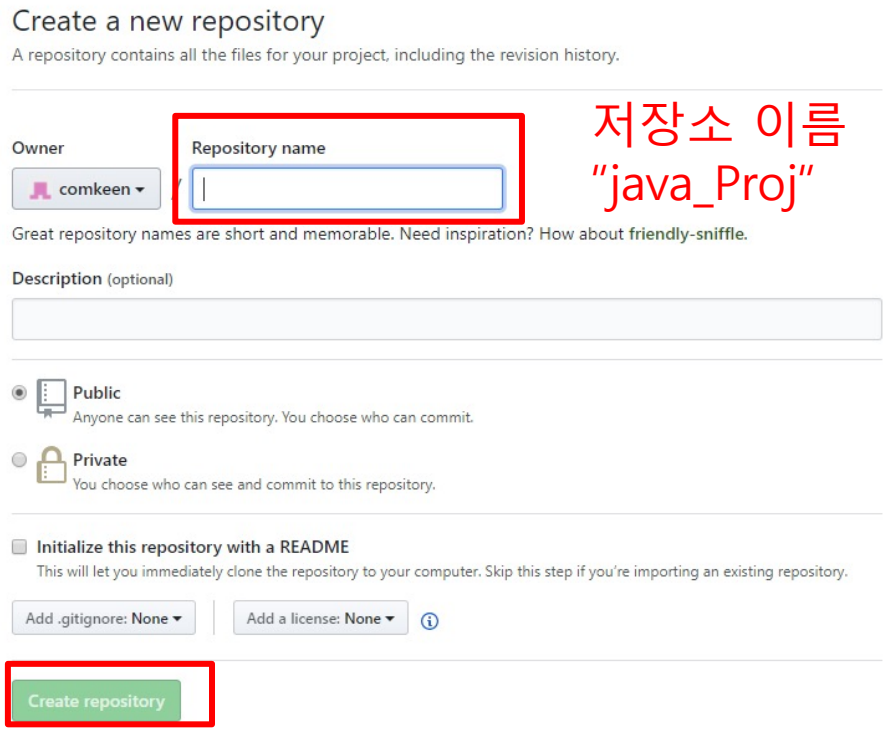
2-1 GitHub 새 저장소 만들기(Repository)



https://github.com/new



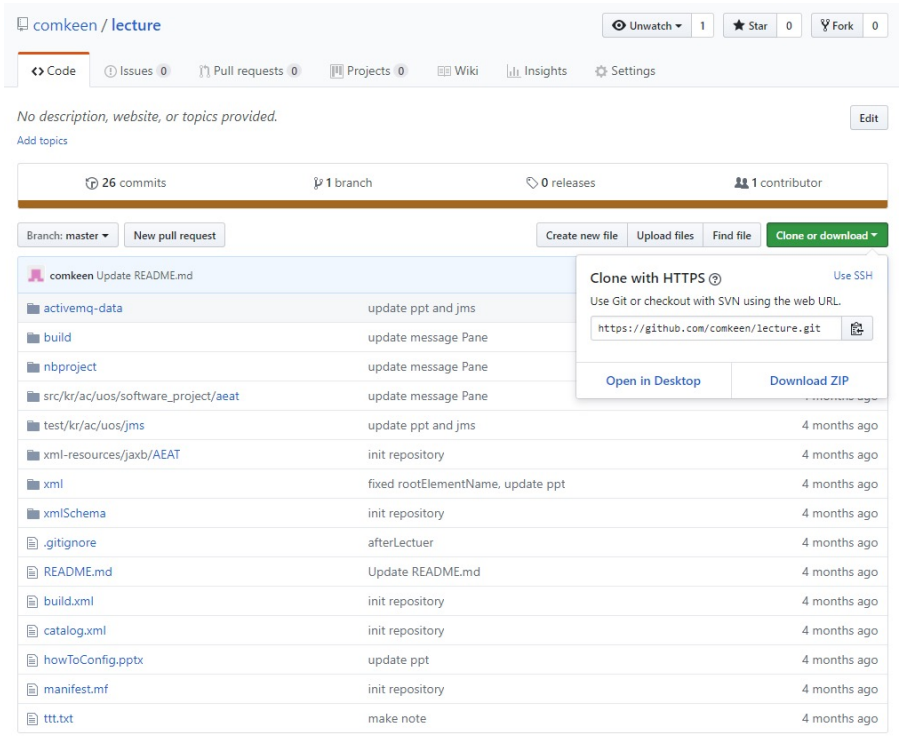
New repository ->
사용할 저장소 이름 작성 ->
Create repository



실습 - 로컬 저장소 만들기

- **'Win+R', cmd**
  - 커맨드라인 실행
- **mkdir java_Proj**
  - 커맨드라인에서 깃으로 관리할 workspace(java_Proj)로 이동
- **cd java_Proj**
  - (cd 디렉토리 이동)
- **echo "# java_Proj" >> README.md**
  - README.md 파일생성 및 내용 입력 (type "README.md" 내용 출력)
- **git init**
  - Git 사용을 위한 초기화작업, .git (폴더 생성)
- **git config –global user.name "anything(영문이름)"**
- **git config –global user.email "깃허브계정메일주소"**
- **git add .**
  - 모든 파일 git 사용준비, (" . " : 현재 디렉토리)
- **git commit –m "anything(변경사항설명)"**
  - 로컬 저장소에 변경사항 저장

로컬 저장소와 원격 저장소 연결



- **git remote add origin 원격저장소주소**
  - 예) git remote add origin https://github.com/username/myproject.git
  - 원격저장소주소 변경) git remote set-url origin 변경할주소
- **git push origin master**
  - 최초 push일 경우, master 브랜치로 지정

git 사용

- **내 소스코드 변경사항을 원격저장소에 저장**
  - 1. git add .
  - 2. git commit –m '***anything(변경사항설명)***'
  - 3. git remote add origin https://github.com/***username***/java_Proj.git
  - 4. git push --set-upstream origin main
  - 5. git push

- **원격 저장소 불러오기**
  - git clone
    - 원격저장소의 소스코드를 로컬에서 새로 사용 할 때
  - Git pull
    - 로컬의 소스코드를 원격저장소와 동기화 할 때

최초 Git 설정 1/4

- **이 단계는 원격 저장소에 연결하기 전에 최초로 Git 설정 할 때만 진행하자**
- **Git 사용을 위한 초기화작업**
  - git init

```
D:₩Repository₩lecture>git init
```

- **Git 사용자이름 설정**
  - git config user.name "anything(영문이름)"

```
D:₩Repository₩lecture>git config user.name "comkeen"
```

- **Git 이메일주소 설정**
  - git config user.email "깃허브계정메일주소"

```
D:₩Repository₩lecture>git config user.email "comkeen@naver.com"
```

깃허브 계정 생성시 사용했던 메일주소로 하자

최초 Git 설정 2/4

- **Git으로 관리할 파일 모두 추가**
  - git add .

```
D:₩Repository₩lecture>git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
```

Warning은 무시하자. 윈도우와 리눅스의 줄바꿈 방식 차이 때문에 발생하는 것

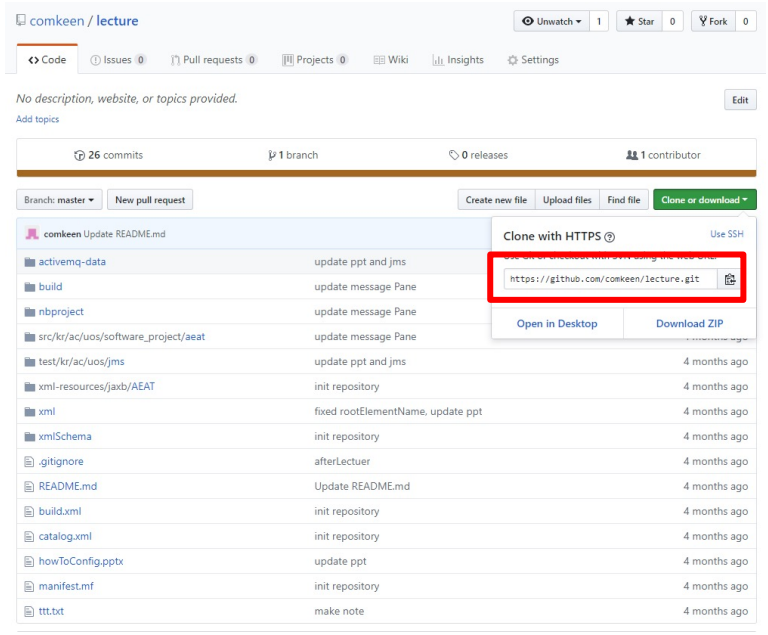
- **Git 로컬 저장소에 변경사항 저장**
  - git commit –m "anything(변경사항설명)"

```
D:₩Repository₩lecture>git commit -m "최초설정"
[master 3f5714d] 최초설정
 5 files changed, 30 insertions(+), 6 deletions(-)
 create mode 100644 .project
 create mode 100644 RemoteSystemsTempFiles/.project
```

변경사항에 대해 보여준다

최초 Git 설정 3/4

HTTPS 주소로 하자
SSH는 보안토큰을 이용해 접근하기 위해 사용하는 것



- **Git 원격 저장소 주소 설정**
  - GitHub의 원격저장소 주소 복사
  - git remote add origin 원격저장소주소

```
D:₩Repository₩lecture>git remote add origin https://github.com/comkeen/lecture.git
```

- 원격 저장소 주소를 잘못 입력했을 경우, 다음과 같이 주소를 다시 지정해 주자
- git remote set-url origin 원격저장소주소

```
D:₩Repository₩lecture>git remote set-url origin https://github.com/comkeen/lecture.git
```

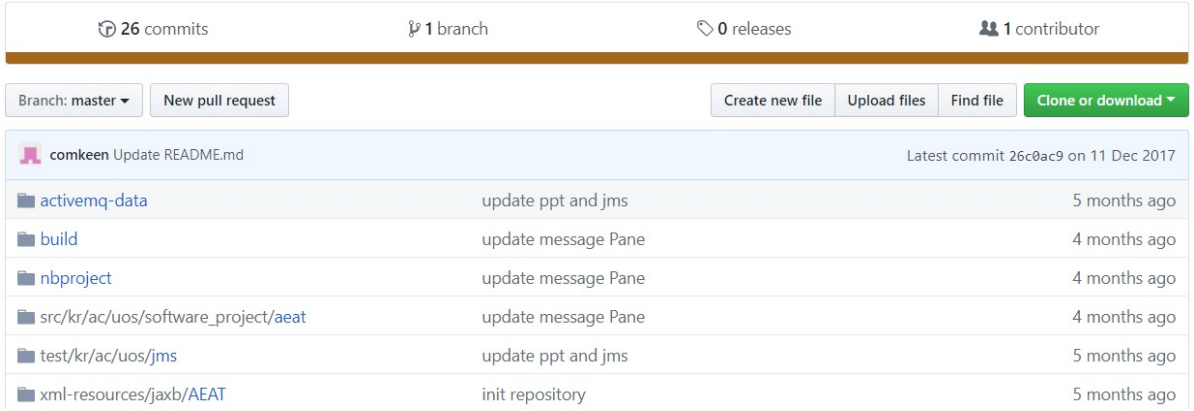
최초 Git 설정 4/4

- **원격 저장소에 변경사항 업로드**
  - git push –u origin master

이전 문서에서는 –u 옵션이 없었는데 넣자

```
D:₩Repository₩lecture>git push -u origin master
Username for 'https://github.com': comkeen
Password for 'https://comkeen@github.com':
Counting objects: 10, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 1024 bytes | 512.00 KiB/s, done.
Total 10 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To https://github.com/comkeen/lecture.git
   26c0ac9..3f5714d  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

깃허브 로그인 절차



푸시까지 정상적으로 이루어 졌다면 깃허브에 여러분이 작성한 코드가 업로드 되었을 것이다

Git 사용: 원격 저장소를 로컬로 불러오기

- **원격 저장소에 업로드 된 소스코드를 다른 컴퓨터에서 사용하고자 할 때**
  - git clone 원격저장소주소

```
D:₩>git clone https://github.com/comkeen/lecture.git
```

- **Clone 혹은 최초 git 설정한 프로젝트에서 원격 저장소에 업로드 된 변경사항을 불러올 때(동기화)**
  - git pull

```
D:₩lecture>git pull
remote: Counting objects: 10, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 10 (delta 5), reused 10 (delta 5), pack-reused 0
Unpacking objects: 100% (10/10), done.
From https://github.com/comkeen/lecture
   26c0ac9..3f5714d  master     -> origin/master
Updating 26c0ac9..3f5714d
Fast-forward
 .gitignore                              |  3 ++-
 .project                                | 11 ++++++++++
 RemoteSystemsTempFiles/.project         | 12 ++++++++++++
 nbproject/private/private.properties |  2 +-
 nbproject/private/private.xml        |  8 ++++----
 5 files changed, 30 insertions(+), 6 deletions(-)
 create mode 100644 .project
 create mode 100644 RemoteSystemsTempFiles/.project
```

- 이전 버전의 기존의 로컬 저장소에 원격 저장소의 최신 버전 동기화
- 항상 작업하기 전에 pull 명령을 통해 최신 상태로 동기화 시키고 작업을 수행하자
- 최신 상태로 동기화 하지 않은 상태에서 작업할 경우 변경사항이 충돌해서 merge를 해야 하는 일이 생길 수 있다

Git 사용: 변경사항 업로드 하기

- **Clone 혹은 최초 git 설정한 프로젝트에서 작업을 수행할 경우**
  - git add .

```
D:₩lecture>git add .
```

변경된 파일 추가

  - git commit –m ‘anything(변경사항설명)’

```
D:₩lecture>git commit -m '새기능추가'
```

로컬 저장소에 변경사항 저장

  - git push

```
D:₩lecture>git push
Username for 'https://github.com': comkeen
Password for 'https://comkeen@github.com':
Everything up-to-date
```

원격 저장소에 변경사항 업로드

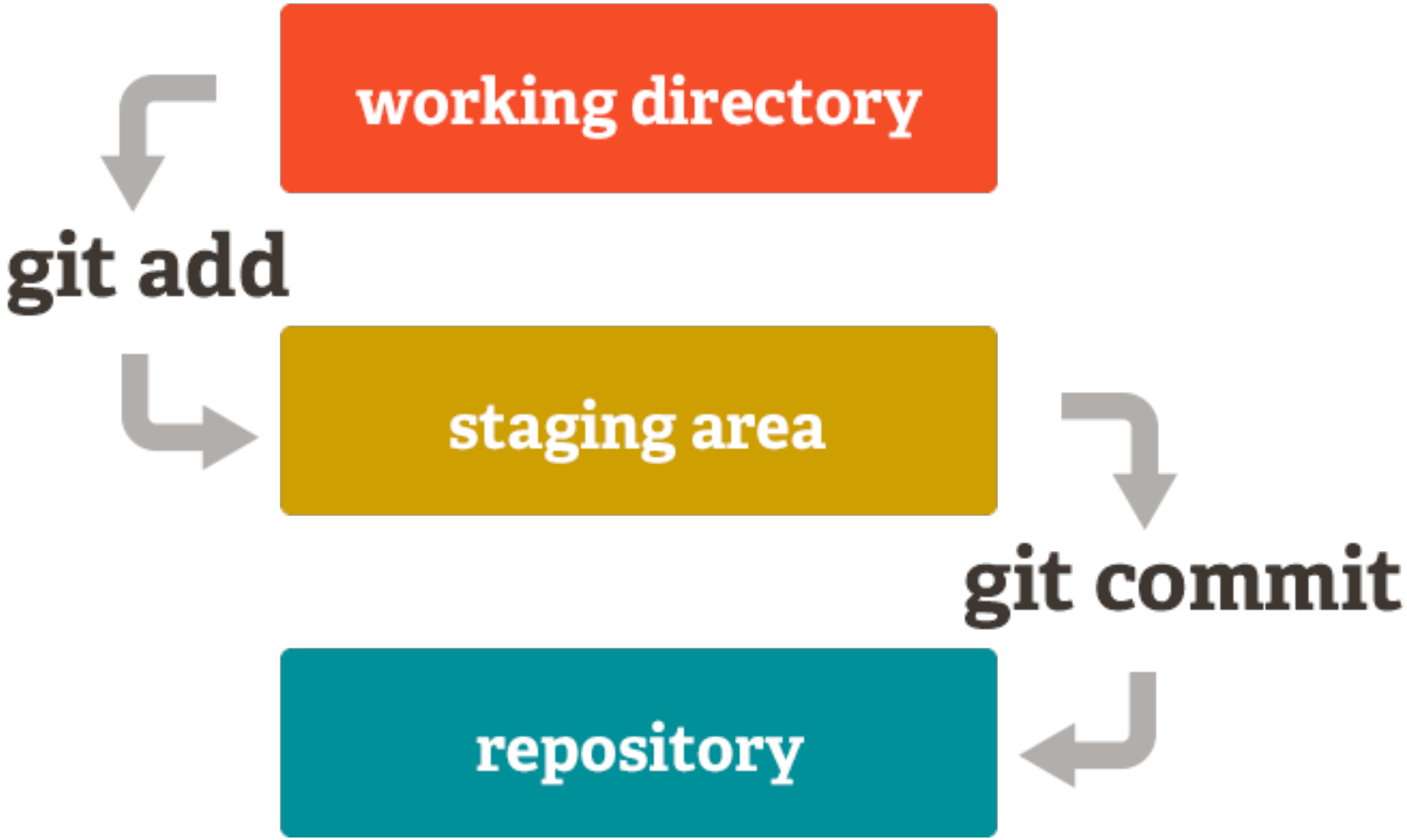
문제해결

- **Push에서 문제가 발생하는 경우**
  1. pull을 수행하고 commit, push를 수행해보자
  2. Git 사용자이름과 이메일 주소 설정에서 --global 옵션을 빼고 설정해보자

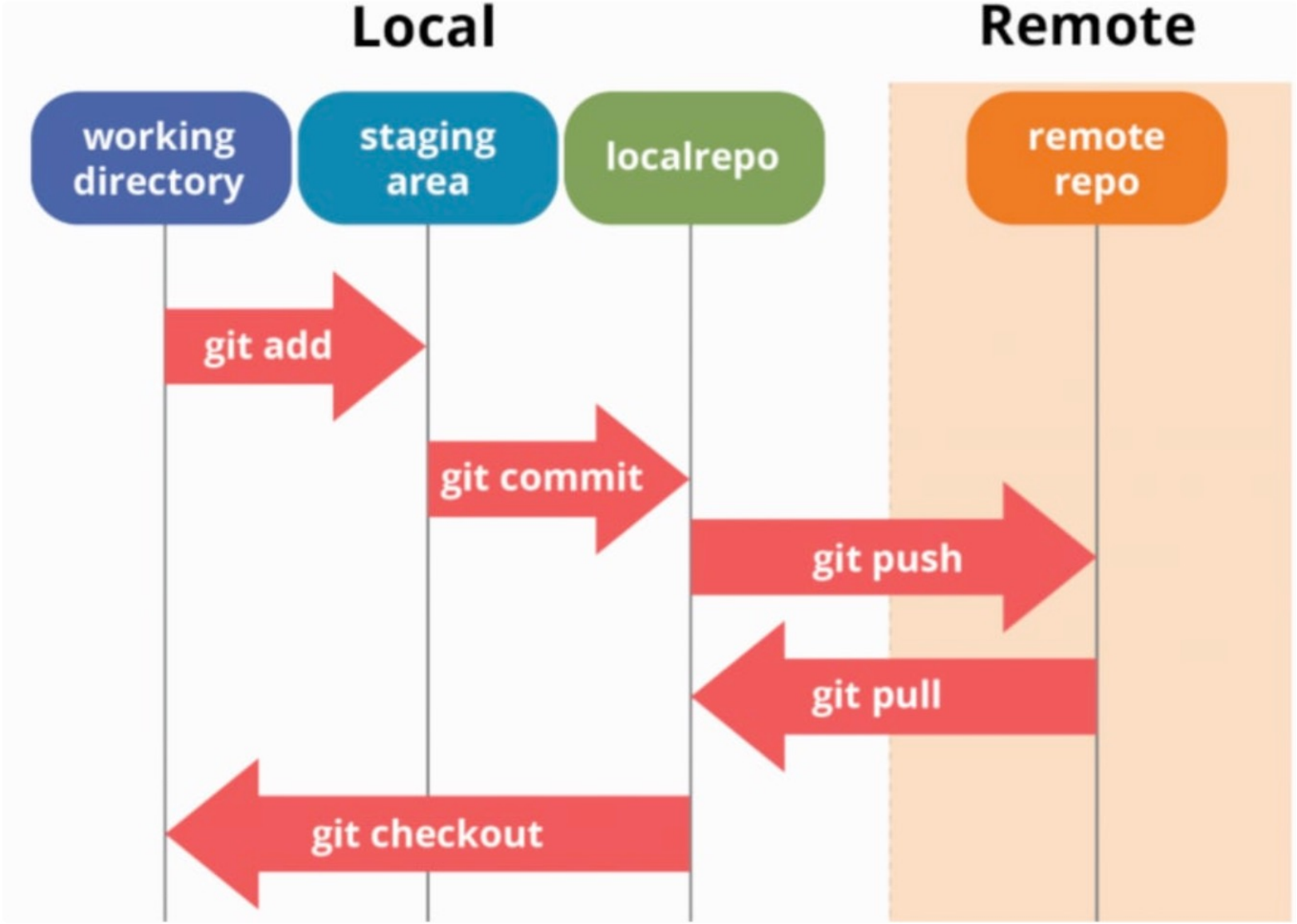
```
D:₩Repository₩lecture>git config user.email "comkeen@naver.com"
```

  3. 그래도 안되면 GitHub의 원격 저장소를 삭제하고 다시 생성해서 최초 Git 설정부터 다시 진행해보자
- **명령 프롬프트의 에러메시지를 구글에서 검색해보자**

Staging Area



Git vs GitHub



발 전

감사합니다.

끝