

버전관리 도구 Git

광명융합기술교육원

데이터분석과

버전관리 도구 - Git

- 0 버전관리
- 1 GitHub
- 2 Local Git 설치
- 3 Remote 저장소 연동하기

0. 버전관리

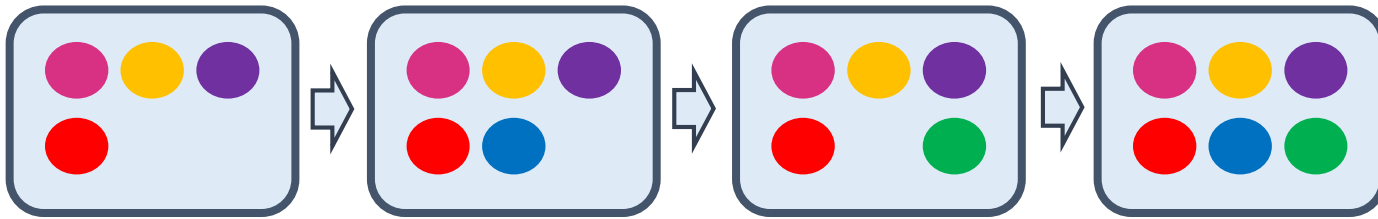
버전관리

• 버전 관리

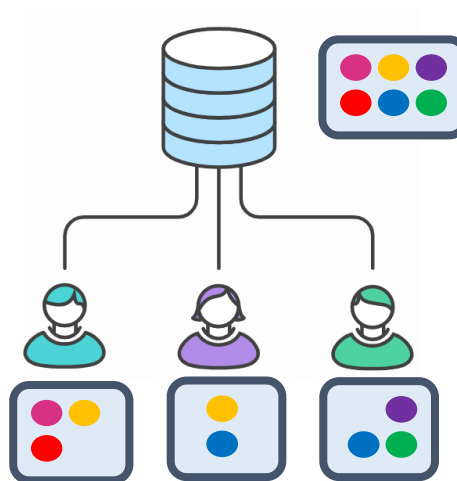
- ✓ 소프트웨어 개발시 매 버전마다 형상을 관리하는 것

• 버전관리의 필요성

- ✓ 코드의 변경사항의 추적과 관리



- ✓ 팀원들 간의 효율적인 협업
 - 내가 고친거 누가 덮어 썼어?
 - 길동씨. 어제 고친거 메일로 보내줘요



과제제출_최종
과제제출_최종_1
과제제출_최종_2
과제제출_최종_3
과제제출_최종_진짜최종
과제제출_최종_진짜최종_마지막

처음꺼가 좋았던 것 같아요.
다시 처음으로 갑시다!



누구인가?
누가 버그를 심었는가?



버전관리

- 여러 개발자가 함께 개발시 source code 수정, 추가, 삭제 관리
 - ✓ 서로 같은 code를 고치면 문제 발생
 - ✓ 버그가 발생시 어느 버전에서 버그 삽입이 되었는지 확인 및 그 이전버전으로 변경 필요
 - ✓ 누가 어떤 bug를 만들어 냈는지 확인 및 수정 가능
- 여러 버전 관리 도구
 - ✓ CVS, Subversion, Git 등

Q. 여러 명이 함께 작업할 때만 필요한 것 아닌가요? 혼자 할 때 굳이 필요해요?

➔ A. 스스로의 체계를 갖추면 필요 없음. 효율적인 관리를 위한 것임.
하지만 소프트웨어 규모가 커질수록 체계적인 버전 관리는 필수

Git 활용

- **Opensource 프로젝트 참여**

- ✓ 전세계 여러 개발자와 함께 코드 작성

- **개인 프로젝트 외부 공유**

- ✓ 취업시 포트폴리오 활용
- ➔ 지금부터 꾸준히 git에 저장 하기

- **단기 목표 – 개인별 Git을 활용하여 최종 프로젝트 관리 (9월 말)**

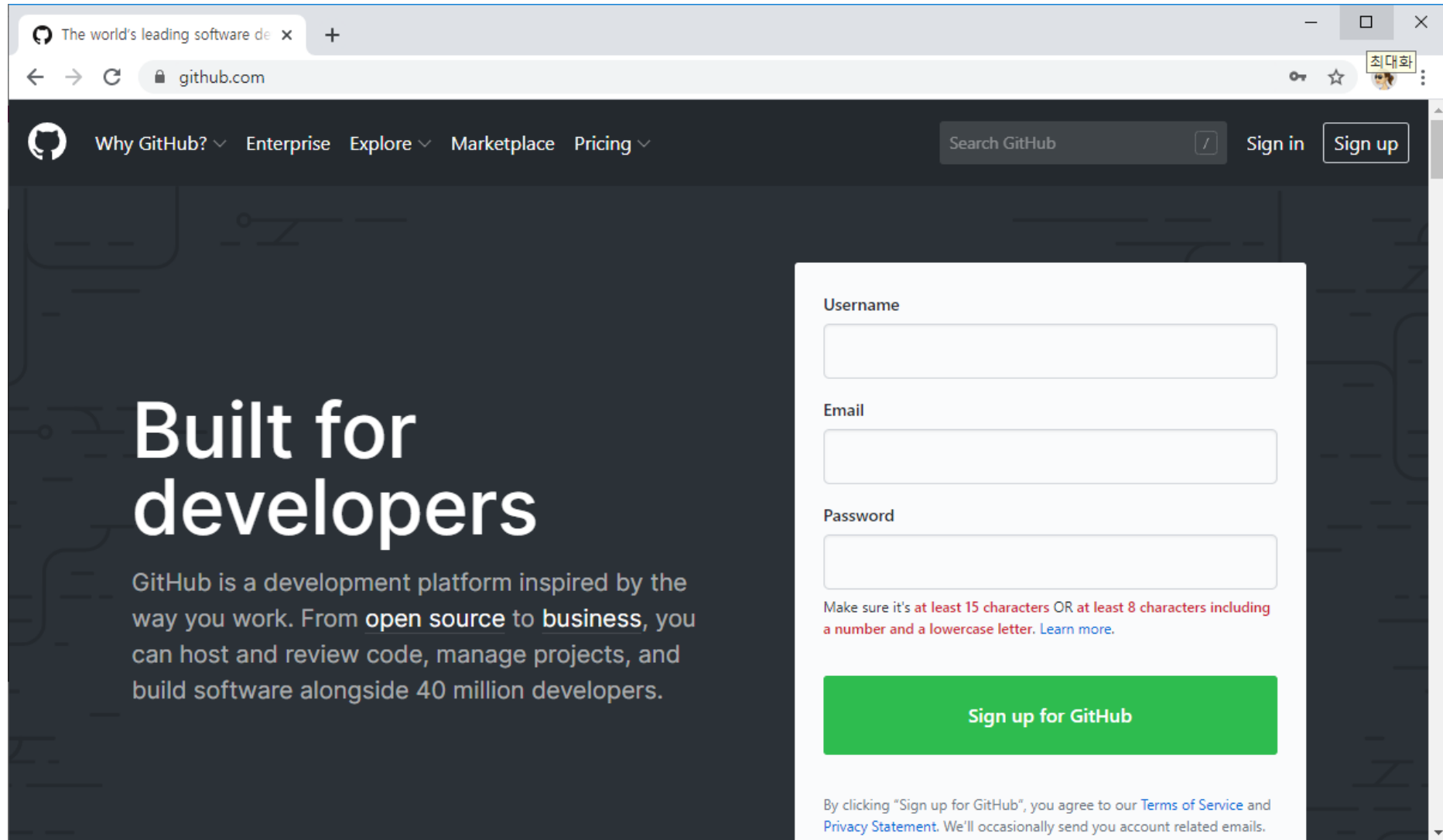
- ✓ "2020년 데이터 분석과 github"을 만들어서 학생 개별 공간에 최종 프로젝트 게시 예정
- ✓ 효과
 - Git 활용 역량 입증
 - 일정 수립, 중간 체크 가능
- ✓ 광명융합기술교육원 github
 - <https://github.com/koposoftware>
- ✓ 학생 개별 repository 및 홈페이지
 - <https://github.com/koposoftware/{repository명}>
 - <https://koposoftware.github.io/{repository명}>

정보시스템개요 과목 목표

- Github 가입 및 기초 사용
- Local에서 Git console 사용
- Local – remote (github) 연결
 - ✓ Console로 연동
 - ✓ Eclipse 설치 및 eclipse로 연동
 - ✓ Sourcetree(Git GUI tool)로 연동

I. GitHub

I.1 Github 가입하기



The screenshot shows the GitHub homepage in a web browser. The browser's address bar displays 'github.com'. The page features a dark header with the GitHub logo, navigation links ('Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', 'Pricing'), a search bar, and 'Sign in' and 'Sign up' buttons. The main content area has a dark background with the text 'Built for developers' and a description of GitHub as a development platform. On the right, a white sign-up form is overlaid, containing fields for 'Username', 'Email', and 'Password', along with a green 'Sign up for GitHub' button and a disclaimer at the bottom.

The world's leading software de x +

github.com

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub / Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

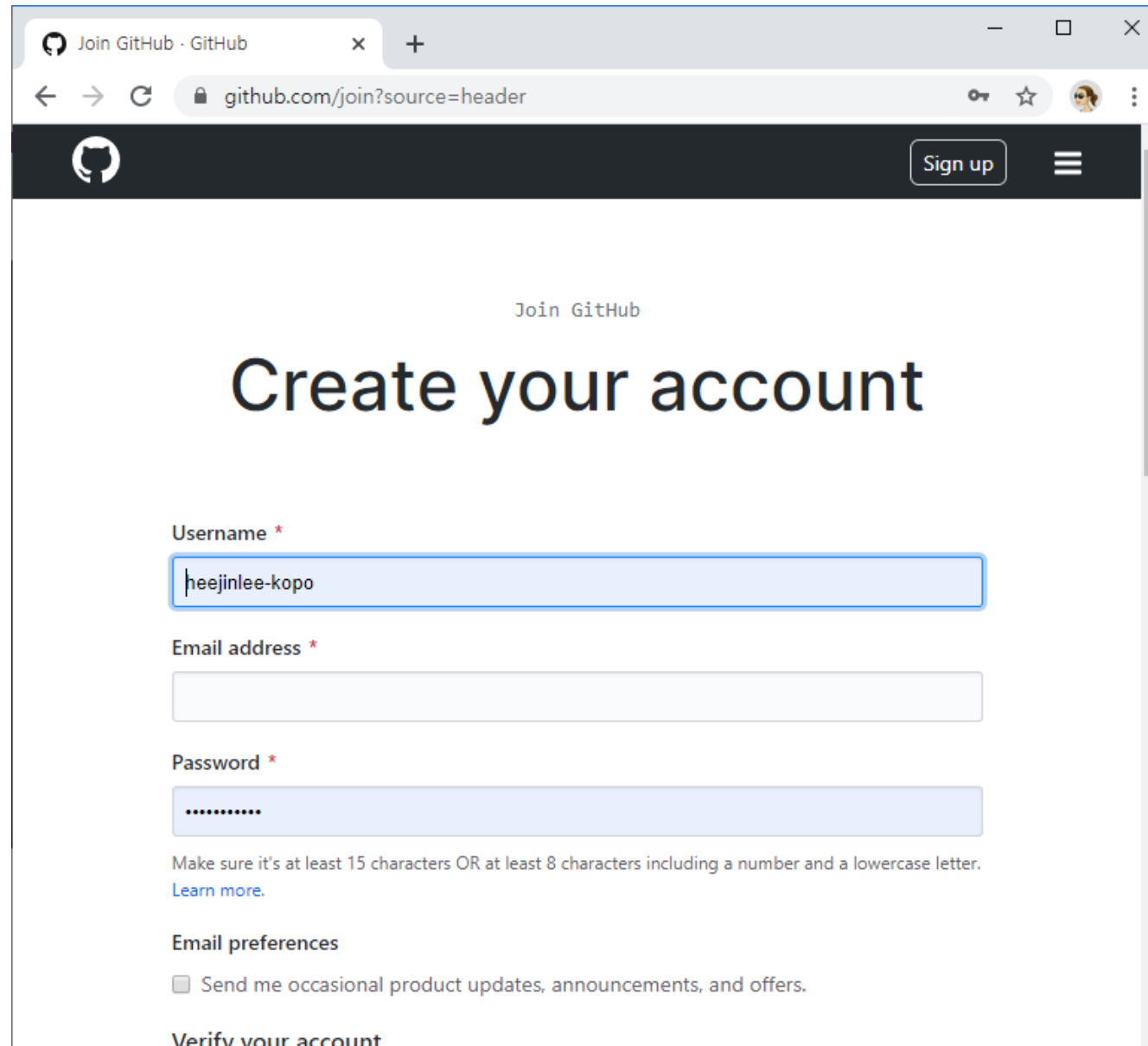
Email

Password

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

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.



The screenshot shows a web browser window with the GitHub sign-up page. The browser's address bar shows 'github.com/join?source=header'. The page has a dark header with the GitHub logo and a 'Sign up' button. The main content area is white and features the heading 'Join GitHub' followed by 'Create your account'. Below this are three input fields: 'Username' with the value 'heejinlee-kopo', 'Email address' which is empty, and 'Password' which contains masked characters. A note below the password field states: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. At the bottom, there is an 'Email preferences' section with an unchecked checkbox for 'Send me occasional product updates, announcements, and offers.' and a link to 'Verify your account'.

Join GitHub · GitHub

github.com/join?source=header

Sign up

Join GitHub

Create your account

Username *

heejinlee-kopo

Email address *

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](#)

Choose a plan(1/2)

- Individual – free 선택


Individual Pick the plan that's right for you, personally.	Team Choose a plan to help your team grow and collaborate.
 Free \$0 USD	 Pro \$7 USD Per month
The basics of GitHub for every developer	Pro tools for developers with advanced requirements
Choose Free	Choose Pro


Choose a plan(2/2)


- Team – owner만 하면 됨. 하지 마시오.
 - ✓ 사용자 별로 repository 권한 부여 가능

Individual
Pick the plan that's right for you, personally.

Team
Choose a plan to help your team grow and collaborate.


Team for Open Source
\$0 USD
Per user / month
Collaboration tools for teams who don't need private repositories
[Choose Team for Open Source](#)


Team
\$9 USD
Per user / month
Starts at \$25 and includes 5 users
Advanced collaboration and management tools for teams
[Choose Team](#)


Enterprise
\$21 USD
Per user / month
Security, compliance, and deployment controls for organizations
[Start your 14-day free trial](#)



Create a new repository

- Repository “hello-world”

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 *

 hjkopo / 

Great repository names are short and memorable. Need inspiration? How about [cuddly-funicular?](#)

Description (optional)

test repository

- ☒  **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

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

<https://guides.github.com/activities/hello-world/>

hjkopo /

hello-world

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

test repository

Edit

Manage topics

1 commit

1 branch

0 packages

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

hjkopo Initial commit

Latest commit ee74717 16 seconds ago

README.md

Initial commit

16 seconds ago

README.md

testrepository

test repository

Create a new branch

- Repository에 가서, new branch를 생성
 - ✓ Branch "readme-edits" – readme file을 고치는 용도
 - ✓ Branch "gh-pages" – github 에서 제공하는 웹 페이지 수정 branch

Just another repository — Edit

🔄 1 commit

🌿 1 branch



🌿 branch: master ▼

hello-world / +

Initial commit



hubot authored just now



README.md

Initial



README.md

- Commit – 저장소에 수정사항을 저장하는 것
- Commit 방법
 - ✓ Git client에서 하는 방법(사용자 pc)
 - ✓ 약식으로 해보자
- Branch “readme-edits”
 - ✓ Readme.md 수정
 - ✓ Commit

Your recently pushed branches:


🔗 readme-edits (1 minute ago) [Compare & pull request](#)

Branch: readme-edits ▼ [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download ▼](#)

This branch is 3 commits ahead of master. [Pull request](#) [Compare](#)

👤 heejinlee-kopo Update README.md Latest commit a84b95e 36 seconds ago

📄 README.md Update README.md 36 seconds ago

📄 README.md 

hello-world

hello world There is a change in readme.md\n

1

Commit a file (2/2)

- 간단히 홈페이지를 만들어보자.

- ✓ <https://{github-id}.github.io/{repository-name}> 에서 확인

- Branch "gh-pages"

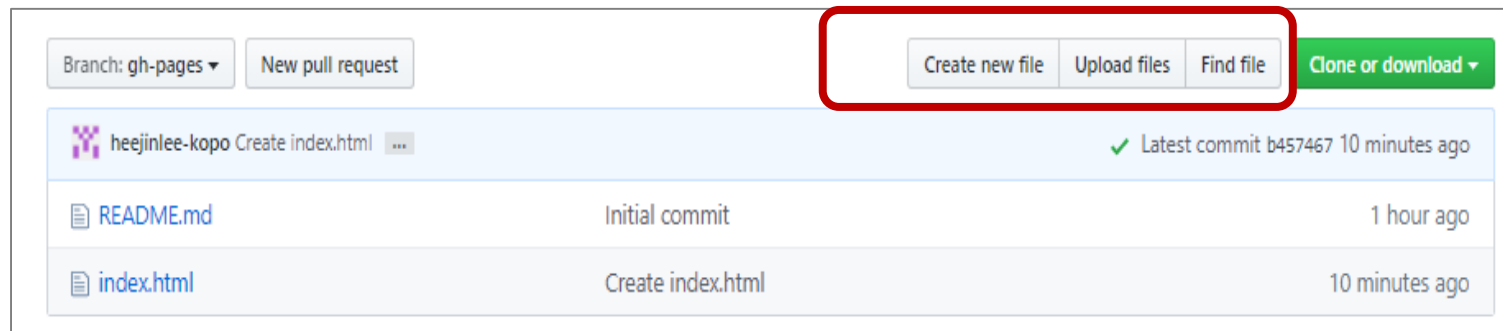
- ✓ Create new file

- Index.html

- ✓ Upload files

- "git.png" 업로드

- ✓ Commit



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Page Title</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <h1>첫 깃허브 페이지입니다</h1>
  <a href="http://naver.com"> 네이버 </a><br>
  <img src = "git.png" />
</body>
</html>
```

- **Github 기본 사용 방법 동영상 작성**

- ✓ Repository 생성, 삭제, branch 생성, 삭제, file upload, commit 진행
- ✓ Repository, branch, commit에 대한 기본 개념 설명 (read-me/ 홈페이지/ 동영상에 설명)
- ✓ Read-me 변경
- ✓ "gh-pages" branch 생성 및 홈페이지 작성

- **해당 과정을 동영상으로 제작하여 본인 repository 홈페이지에 링크 공유**

- ✓ 직접 파일 업로드, YouTube, Naver blog 다 괜찮음

- **제출 방법**

- ✓ Github 해당 repository 홈페이지 url 제출
 - 예) `https://{본인id}.github.io/{repository명}`
- ✓ 정보시스템개요/과제제출/Github 활용 과제1.xls 에 url 기재

- **제출 기한**

- ✓ 4월 7일 화요일 오전 10시

Ⅱ. local Git 설치

Ⅱ.1 Git 설치

Ⅱ.2 Git 기본기능

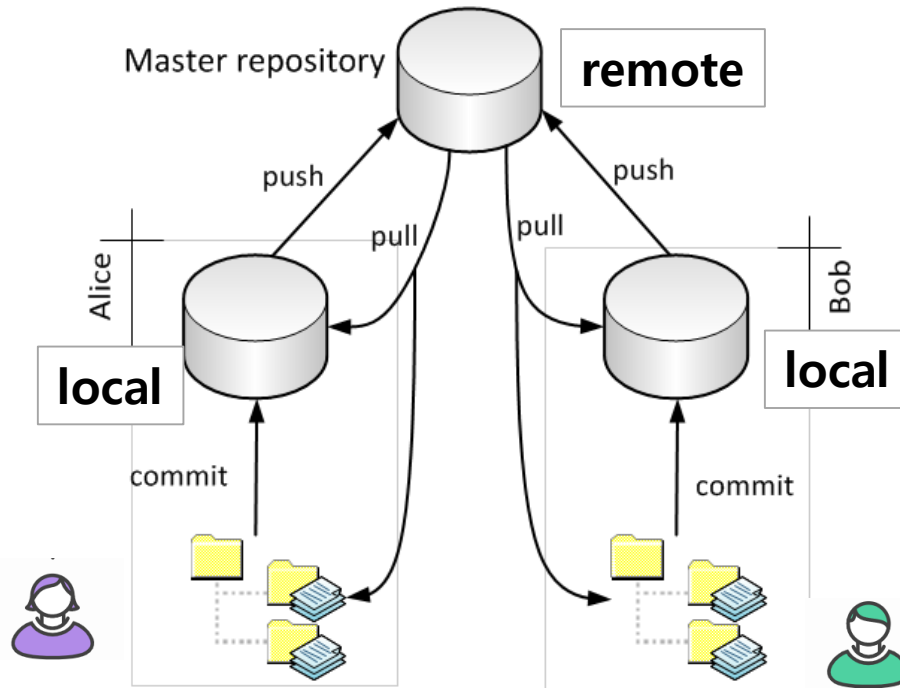
Ⅱ.3 Branch

- 내가 작업하는 컴퓨터에 설치 (local)

- ✓ Console 사용 (설치 필요)
- ✓ GUI console 사용 (설치 필요)
- ✓ Eclipse 를 사용 – eclipse내에 이미 설치되어 있음. (몇가지 설정으로 간단) *

- 공용 저장소에 설치 (remote)

- ✓ On premise – 서버실에 있는 서버에 설치 (private한 용도, 보안 중요한 경우)
- ✓ Cloud – 외부의 저장소를 두고 제공하고 있는 서비스를 사용(SaaS) *



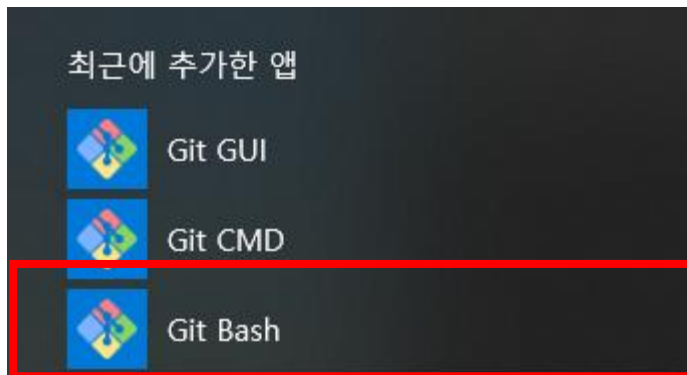
1. Git 설치

- <https://git-scm.com/>
- 다운로드 및 설치 - 각자 os에 맞는 설치파일 설치
- Windows - 2.26.0 - 64bit

2. Git-bash.exe 실행 (windows 기준)

- C:\Program Files\Git\Git-bash.exe

3. Console에서 사용



<기본 linux 명령어>

- 디렉토리 생성

```
>> mkdir MyGitRepository
```

- 디렉토리 들어가기

```
>> cd MyGitRepository
```

- 현재 디렉토리 확인

```
>> pwd
```

- 디렉토리 내용 확인 - 윈도우의 경우

```
>> dir
```

- Git 저장소로 만들기

- ✓ .git이라는 숨김 파일이 생김.
- ✓ MyGitRepository 디렉토리 안에 관리하고자 하는 code들을 넣고 git으로 관리

```
>> git init
```

이름

 .git

```
USER@Heejin MINGW64 /c/workspace/hello-world
$ git init
Initialized empty Git repository in C:/workspace/hello-world/.git/

USER@Heejin MINGW64 /c/workspace/hello-world (master)
$ |
```

- 저장소 상태를 확인

```
>> git status
```

- Git이 관리해야 할 파일을 지정

- ✓ git add를 하지 않으면 commit시에 git에 반영 안됨

```
>> git add 파일명  
예) git add a.txt
```

- 변경사항 최종 저장

- ✓ Add한 파일을 commit

```
>> git commit -m '설명'  
예) git commit -m [feature]로그인 기능 추가
```

- 파일 지정 + 최종 저장

```
>> git commit -am '설명'
```

Git 기본 사용 방법 (4/6)

- 기본 사용자 setting

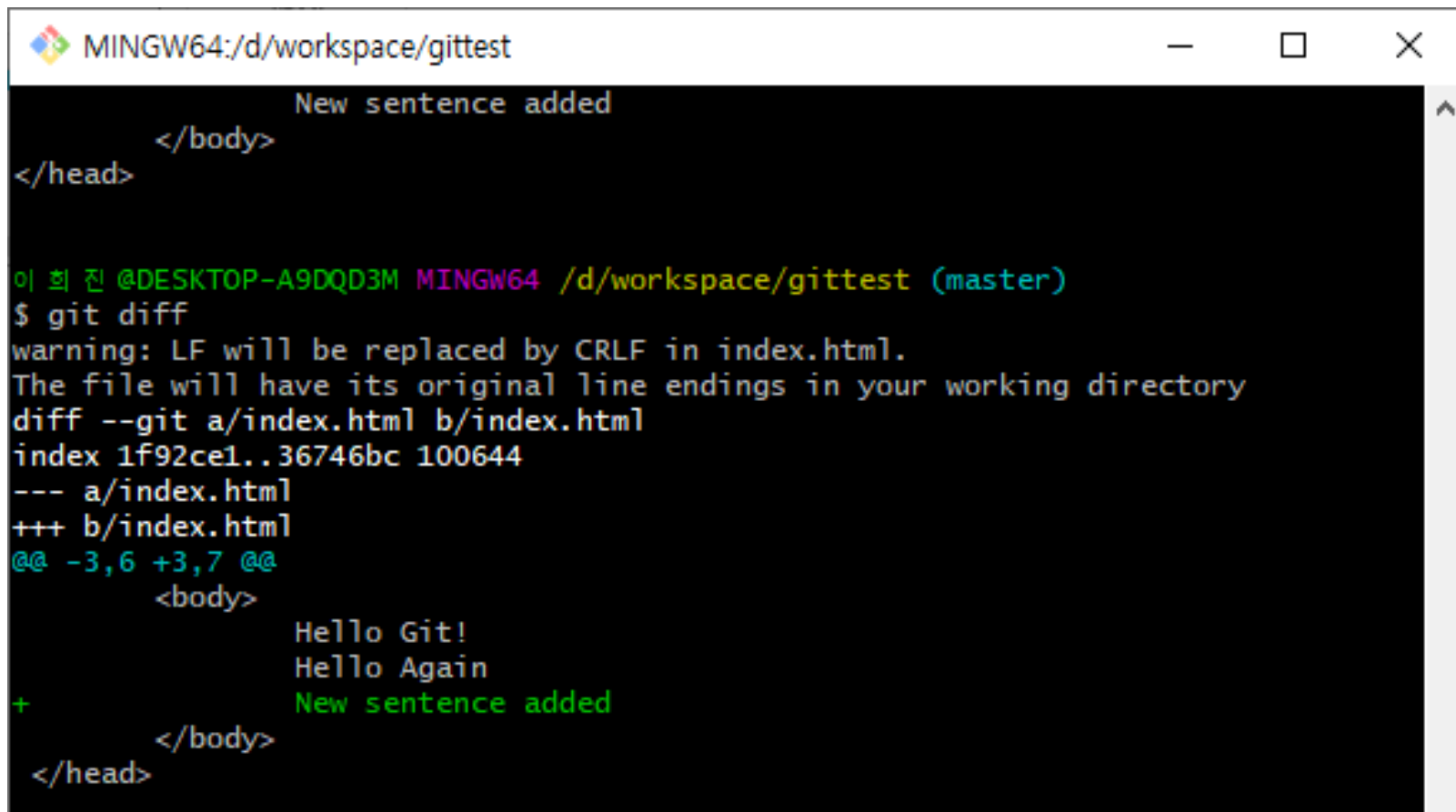
```
>> git config -global user.name "홍길동 "  
>> git config -global user.email "you@example.com"
```

```
USER@Heejin MINGW64 /c/workspace/hello-world (master)  
$ git commit -m "a.txt add first commit"  
  
*** Please tell me who you are.  
  
Run  
  
    git config --global user.email "you@example.com"  
    git config --global user.name "Your Name"  
  
to set your account's default identity.  
Omit --global to set the identity only in this repository.  
  
fatal: unable to auto-detect email address (got 'USER@Heejin.(none)')  
  
USER@Heejin MINGW64 /c/workspace/hello-world (master)  
$ |
```

- 변경사항 파악

- ✓ Commit전에 마지막 버전 대비 변경된 내용을 최종 확인 기능

```
>> git diff
```



```
MINGW64:/d/workspace/gittest

New sentence added
</body>
</head>

이 회 진 @DESKTOP-A9DQD3M MINGW64 /d/workspace/gittest (master)
$ git diff
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
diff --git a/index.html b/index.html
index 1f92ce1..36746bc 100644
--- a/index.html
+++ b/index.html
@@ -3,6 +3,7 @@
     <body>
         Hello Git!
         Hello Again
+        New sentence added
     </body>
 </head>
```


Git 기본 사용 방법 (6/6)

• 전체 변경 이력 조회

```
>> git log           //변경 이력 조회
>> git log -p        //전체 변경 이력 조회
```

```
MINGW64:/d/workspace/gittest
1 file changed, 1 insertion(+)

이희진@DESKTOP-A9DQD3M MINGW64 /d/workspace/gittest (master)
$ git log
commit 8a937492b0dece0565abb35f414d5a949562f966 (HEAD -> master)
Author: 이희진 <heejinlee@kopo.ac.kr>
Date: Thu Feb 27 12:54:21 2020 +0900

    3rd

commit b5286db33cf0f40e5f79e5a3f95ad8a02676a96c
Author: 이희진 <heejinlee@kopo.ac.kr>
Date: Thu Feb 27 12:43:59 2020 +0900

    2nd description

commit 2ee0b80ef5ee7df1fa2c1d684953ea378b65c48b
Author: 이희진 <heejinlee@kopo.ac.kr>
Date: Thu Feb 27 12:41:54 2020 +0900

    git test 첫 번째 commit 설명

이희진@DESKTOP-A9DQD3M MINGW64 /d/workspace/gittest (master)
$ git |
```

```
MINGW64:/d/workspace/gittest

Hello Again
New sentence added
</body>
</head>

commit b5286db33cf0f40e5f79e5a3f95ad8a02676a96c
Author: 이희진 <heejinlee@kopo.ac.kr>
Date: Thu Feb 27 12:43:59 2020 +0900

    2nd description

diff --git a/index.html b/index.html
index 7cd50d9..1f92ce1 100644
--- a/index.html
+++ b/index.html
@@ -2,6 +2,7 @@

    <body>
        Hello Git!
        Hello Again
    </body>
</head>

commit 2ee0b80ef5ee7df1fa2c1d684953ea378b65c48b
Author: 이희진 <heejinlee@kopo.ac.kr>
Date: Thu Feb 27 12:41:54 2020 +0900

    git test 첫 번째 commit 설명

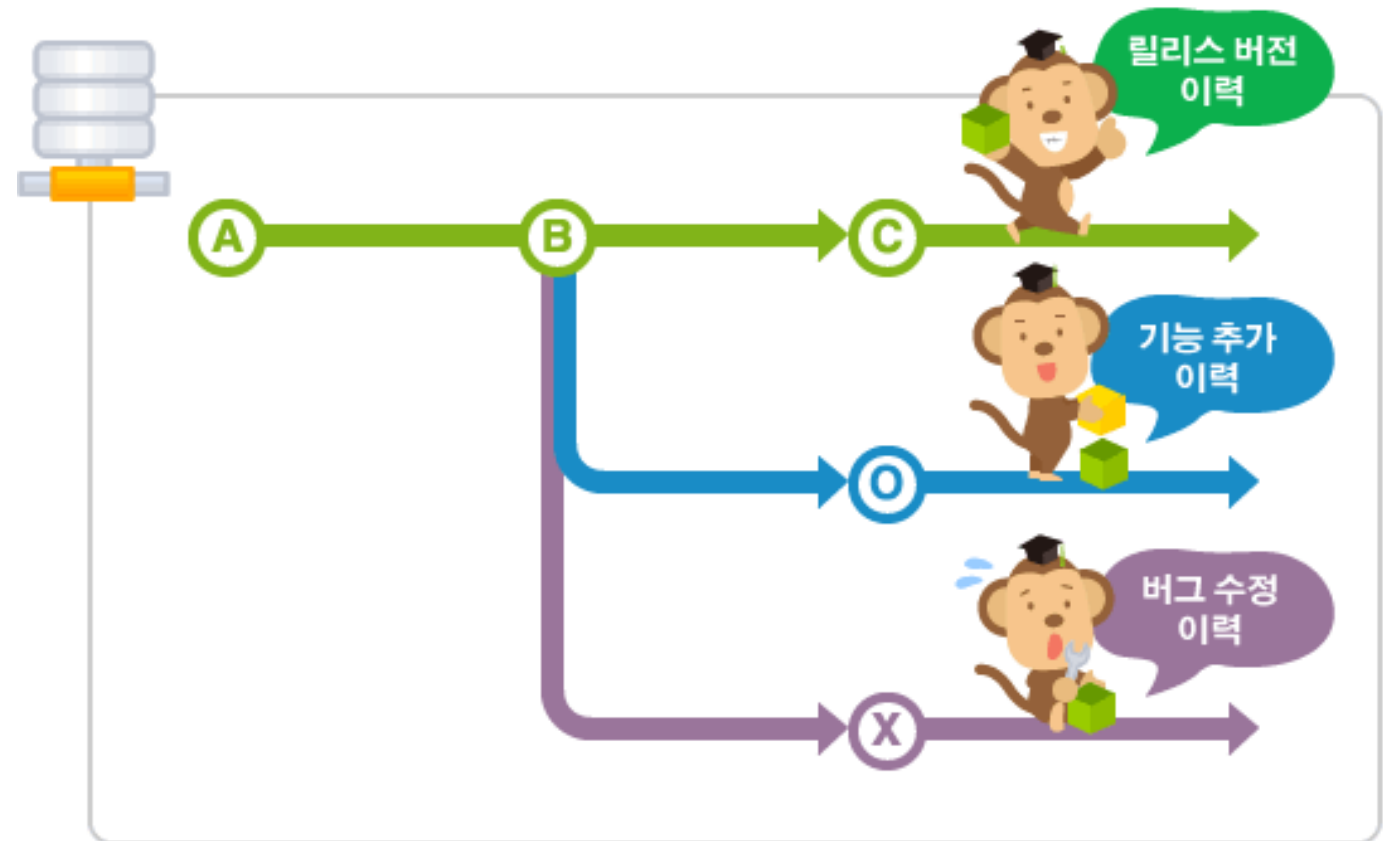
diff --git a/index.html b/index.html
new file mode 100644
index 0000000..7cd50d9
--- /dev/null
+++ b/index.html
@@ -0,0 +1,7 @@
+<head>
+
+    <body>
+        Hello Git!
+    </body>
+</head>
+
(END)
```

• Branch 정의

- ✓ 독립적인 작업을 진행하기 위한 다른 장소
- ✓ 서로 영향을 받지 않고, 여러 작업이 동시에 진행 가능
- ✓ 다른 브랜치와 병합(Merge)하여, 하나의 브랜치로 모을 수 있다.

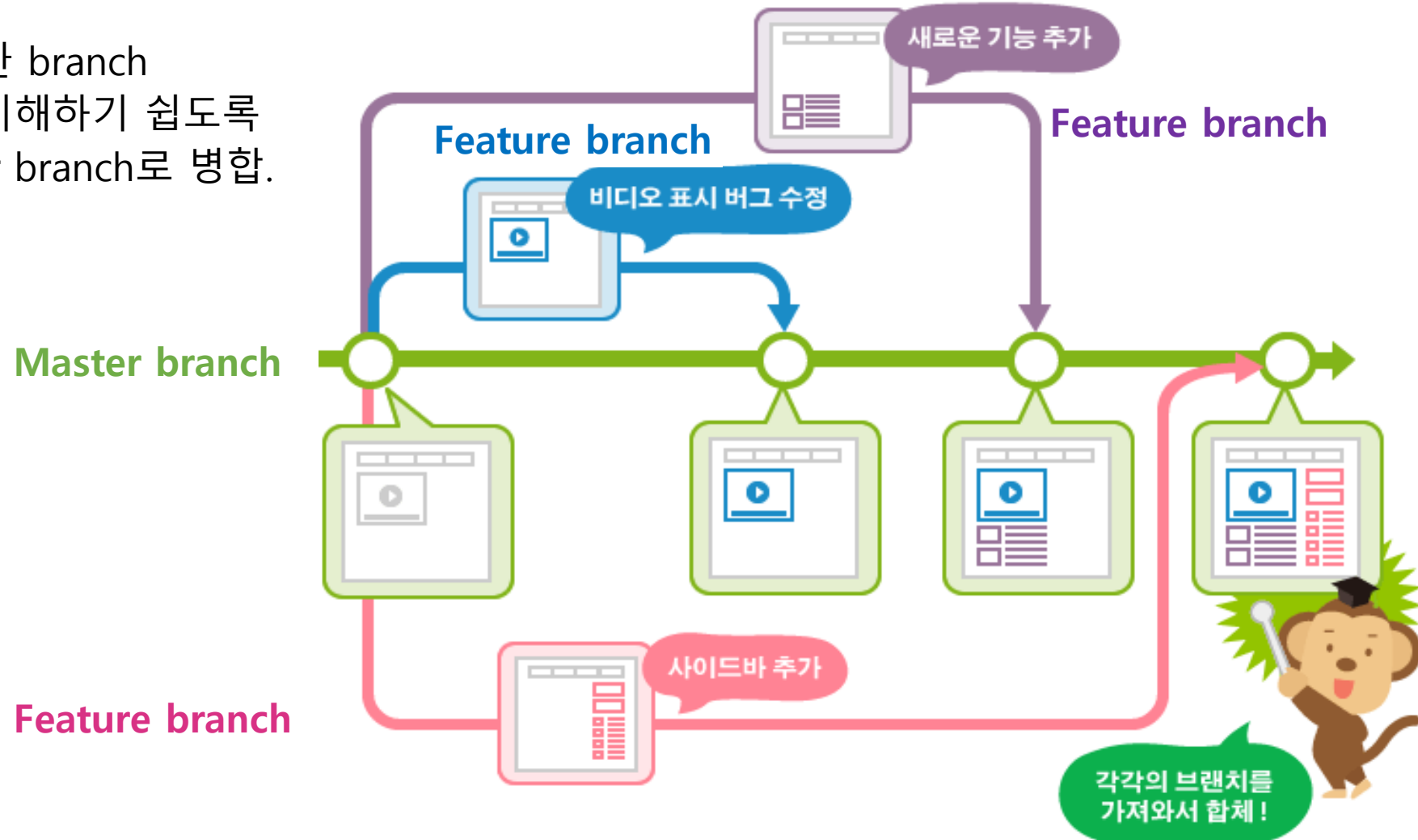
• Branch 필요한 경우 예시

- ✓ 안정적인 버전 외에 새로운 기능을 추가시
- ✓ 개발과 운영을 따로 할 때
- ✓ 고객사 별로 새로운 기능을 제공해야 할 때



• 기본 규칙

- ✓ Master branch - 안정화 branch
- ✓ Feature branch
 - 새로운 기능 구현을 위한 branch
 - 이름을 다른 사람들도 이해하기 쉽도록
- ✓ Feature branch에서 master branch로 병합.



1. 현재 branch 확인

```
>> git branch  
master // 현재 branch 이름
```

2. Branch 생성

```
>> git branch 새로운_branch_명  
>> git branch new_branch
```

3. Branch 변경

```
>> git checkout 변경하고자_하는_branch_명  
>> git checkout new_branch
```

4. Merge branch

<현재 branch(new_branch)에 test.txt commit> (1)

```
>> git branch
```

```
new_branch
```

```
>> git add test.txt
```

```
>> git commit -m "New file test.txt"
```

<반영할 branch(master)로 이동> (2)

```
>> git checkout 반영할_branch_명
```

```
>> git checkout master
```

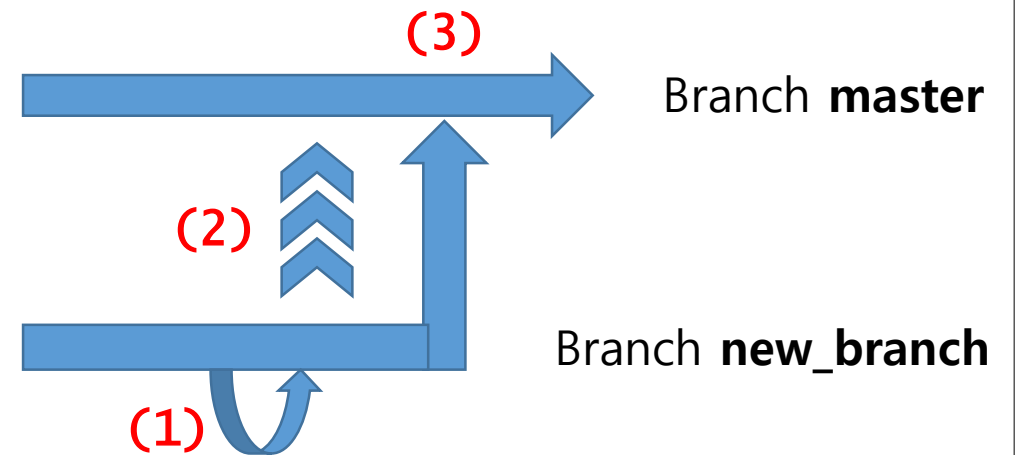
```
>> git branch
```

```
master
```

<merge - new_branch를 master에 merge> (3)

```
>> git merge 병합할_branch_명
```

```
>> git merge new_branch
```



정리

- Github에 가입하고, branch 생성, file commit 해보았다.
- Git을 내 컴퓨터에서 쓸 수 있다.
- ➔ 내 컴퓨터에서 변경한 file을 공용 저장소에 올려서 여러명이 다 같이 접근해서 사용하자. 내 컴퓨터와 외부 공용 저장소를 연결하자

Ⅲ. Remote 저장소 연동하기

Ⅲ.1 remote 저장소 추가 및 관리

Ⅲ.2 Git Clone

Ⅲ.3 Git Push

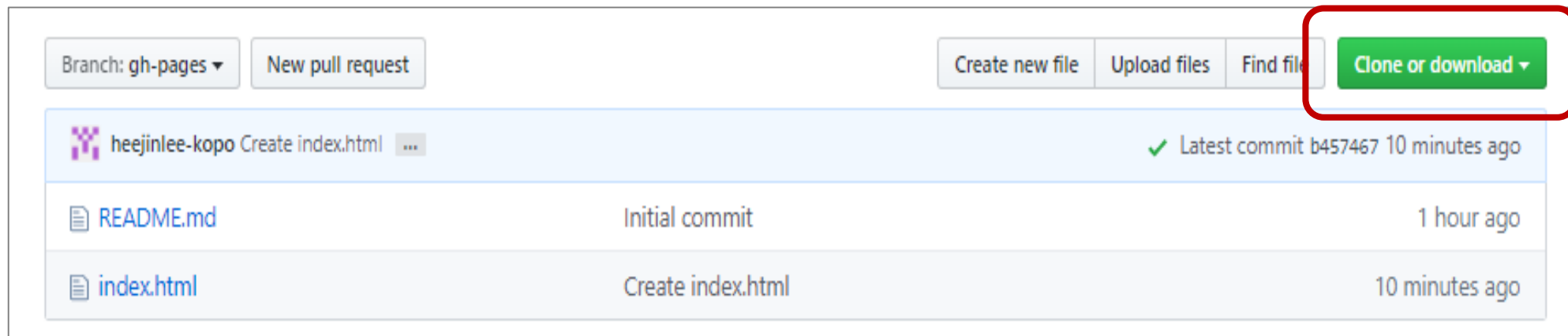
Ⅲ.4 Git Pull

- 로컬저장소와 Remote 저장소 (Github/외부 서버) 연결

```
>> git remote add 저장소이름 원격저장소_주소 //원격저장소 저장
>> git remote add origin http://github.com/xxx/yyy.git

>> git remote origin //원격저장소 목록 - 이름

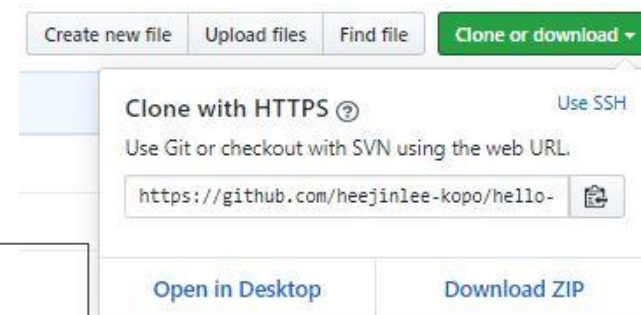
>> git remote -v //원격저장소 목록 - 이름, 주소
origin http://github.com/xxx/yyy.git
```



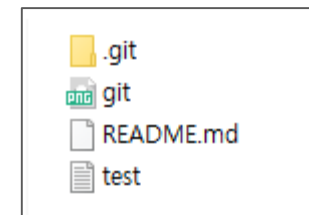
- **Git Clone - Remote 저장소의 branch를 그대로 로컬저장소로 가져오기**

- ✓ 외부 repository 주소 복사
 - `https://github.com/heejinlee-kopo/hello-world.git`
- ✓ 해당 디렉토리에서 `git clone` 명령

```
>> git clone 외부_repository_주소  
>> git clone https://github.com/user\_id/repository\_name.git
```



- ✓ 해당 디렉토리에 폴더 생성 및 `.git` 폴더 생성(숨겨진 폴더)



- Q1. clone을 할 때 Remote 저장소 주소를 사용하면 어떤 branch가 clone되는가?
- Q2. 다른 branch를 clone할 방법이 있는가?

- Push

- ✓ Remote 저장소에 정보 저장하는 것
- ✓ Local 저장소에 commit 한 것은 remote 저장소에 자동 저장되지 않으므로, 명시적으로 작업 필요

- Remote 저장소에 반영

- ✓ Default branch에 반영

```
>> git push
```

- ✓ Branch 지정하여 반영

```
>> git push remote저장소주소 반영할_브랜치명  
>> git push https://github.com/user_id/repository_name.git branch1
```

Remote 저장소에 변경사항 업데이트 받기 – Git Pull

remote 저장소 연동

• Pull

- ✓ 다른 사람들이 remote 저장소에 반영한(push) 내용을 로컬 저장소에 업데이트 하는 것
- ✓ 내가 작업하는 동안 다른 사람이 작업한 것을 업데이트
- ✓ Fetch + Merge
 - fetch – remote 저장소에서 데이터 가져오는 것
 - merge – 다른 부분을 합치는 것

```
>> git pull
```

```
>> git fetch
```

```
>> git merge
```



Q & A

The background features abstract geometric shapes in shades of orange and blue. A large blue triangle is positioned in the upper left, while several orange shapes, including a large triangle and a trapezoid, are layered behind it. A horizontal grey bar spans the middle of the image, serving as a backdrop for the text.

Thank you