

오픈소스 소프트웨어 실습

Open-Source Software Lab

#11

실습 담당 조교 연락처

◆실습 조교 : 김민곤

◆연구실 : 학연산클러스터 601호

◆이메일 : phenix235@hanyang.ac.kr

메일양식: [오픈소스]학번_수업일(20190000)_이름

Git



Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

Git



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



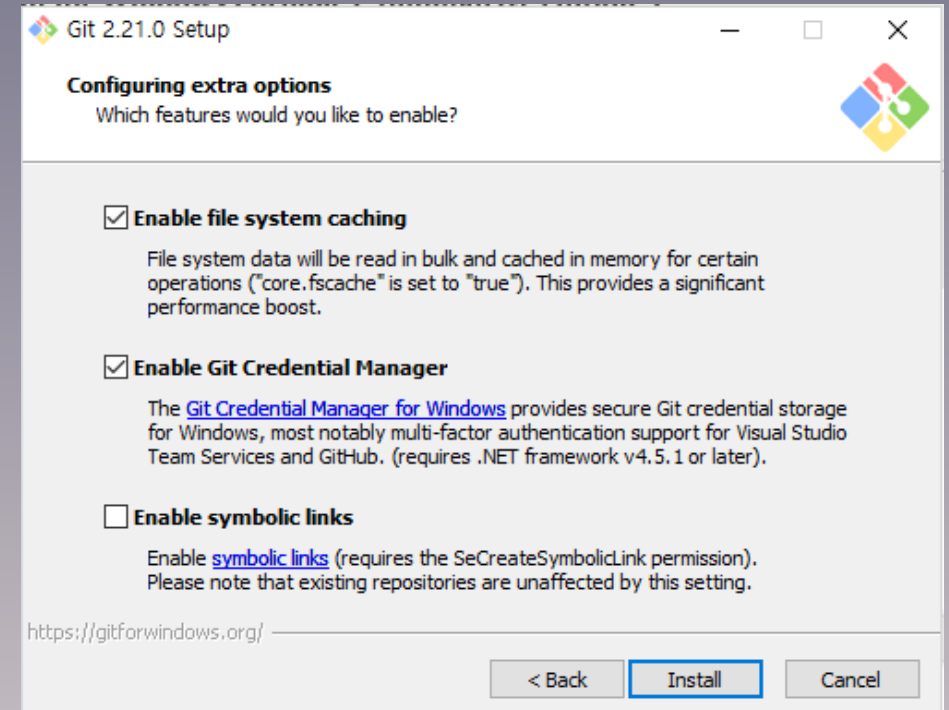
Community

Get involved! Bug reporting, mailing list, chat, development and more.

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

Git 설치

◆ Windows



◆ Mac



Git 설치

◆ Source code

◆ 가장 최근 버전이 필요할 경우 – curl, openssl, expat, libiconv와 같은 라이브러리 등이 필요

```
$ sudo yum install curl-devel expat-devel gettext-devel openssl-devel perl-devel zlib-devel
```

```
$ sudo apt-get install libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libssl-dev
```

◆ 라이브러리 설치 후 아래 링크에서 최신 배포 버전 다운

Kernel.org (<https://www.kernel.org/pub/software/scm/git>)

GitHub (<https://www.github.com/git/git/release>)

```
$ tar -zxf git-2.0.0.tar.gz
```

```
$ cd git-2.0.0
```

Git 설치

◆ Source code

◆ 컴파일 및 설치

```
$ tar -zxf git-2.0.0.tar.gz
```

```
$ cd git-2.0.0
```

```
$ make configure
```

```
$ ./configure --prefix=/usr
```

```
$ make all doc info
```

```
$ sudo make install install-doc install-html install-info
```

◆ 다양한 형식(doc, html, info)의 문서들을 이용하기 위한 패키지 (docbook2X package)

```
✓ $ sudo yum install asciidoc xmlto docbook2X
```

```
✓ $ sudo apt-get install asciidoc xmlto docbook2X
```

Git 사용자 정보 설정

◆ 이름 및 이메일 설정

◆ 설치된 git bash 혹은 git CMD 실행

```
$ git config --global user.name "Mingon Kim"
```

```
$ git config --global user.email mingonkim@example.com
```

◆ --global 옵션은 딱 한번만 하면 되며 해당 시스템에서 해당 사용자가 사용할 때에는 위 정보를 사용

◆ 만약 프로젝트마다 다른 이름과 이메일 주소를 사용하고 싶으면 --global 옵션을 빼고 명령을 실행

Git 사용자 정보 설정

◆ git config

◆ Git의 사용환경 설정 명령

1. /etc/gitconfig 파일 : 시스템의 모든 사용자와 모든 저장소에 적용되는 설정
\$ git config --system 옵션으로 이 파일을 읽고 쓸 수 있다
2. ~/.gitconfig, ~/.config/git/config 파일 : 특정 사용자에게만 적용되는 설정
\$ git config --global 옵션으로 이 파일을 읽고 쓸 수 있다
3. .git/config 파일 : Git 디렉토리에 있고 특정 저장소(혹은 현재 작업 중인 프로젝트)에만 적용

◆ 각 설정은 역순으로 우선시 된다

◆ \$ git config --f <file> 로 시스템 설정 파일의 경로를 바꿀 수 있다 (단, 관리자 권한이 필요하다)

Git 사용자 정보 설정

◆ 편집기

- ◆ 보통 vi 나 vim 을 기본으로 사용
- ◆ 다른 편집기를 사용할 수 있으며 설정을 바꾸면 된다
(ex : Emacs , 편집기마다 설정 방법이 다를 수 있음)

```
$ git config --global core.editor emacs
```

Git 사용자 정보 설정

◆ 설정 확인

- ◆ \$ git config --list 명령을 실행하여 설정한 모든것을 확인

```
C:\Users\Administrator>git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
credential.helper=manager
user.name=mg kim
user.email=phenix235qor@naver.com
core.editor=emacs
```

- ◆ git config <key> 명령으로 특정 key에 대한 어떤 값을 사용하는지 확인 가능

```
$ git config user.name
```

Mingon Kim

Git 저장소 생성

```
C:\Users\Administrator>git
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`)

<code>clone</code>	Clone a repository into a new directory
<code>init</code>	Create an empty Git repository or reinitialize an existing one

work on the current change (see also: `git help everyday`)

<code>add</code>	Add file contents to the index
<code>mv</code>	Move or rename a file, a directory, or a symlink
<code>reset</code>	Reset current HEAD to the specified state
<code>rm</code>	Remove files from the working tree and from the index

examine the history and state (see also: `git help revisions`)

<code>bisect</code>	Use binary search to find the commit that introduced a bug
<code>grep</code>	Print lines matching a pattern
<code>log</code>	Show commit logs
<code>show</code>	Show various types of objects
<code>status</code>	Show the working tree status

grow, mark and tweak your common history

<code>branch</code>	List, create, or delete branches
<code>checkout</code>	Switch branches or restore working tree files
<code>commit</code>	Record changes to the repository
<code>diff</code>	Show changes between commits, commit and working tree, etc
<code>merge</code>	Join two or more development histories together
<code>rebase</code>	Reapply commits on top of another base tip
<code>tag</code>	Create, list, delete or verify a tag object signed with GPG

collaborate (see also: `git help workflows`)

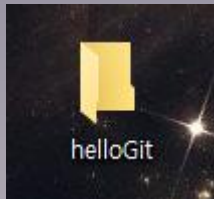
<code>fetch</code>	Download objects and refs from another repository
<code>pull</code>	Fetch from and integrate with another repository or a local branch
<code>push</code>	Update remote refs along with associated objects

'`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.

Git 저장소 생성

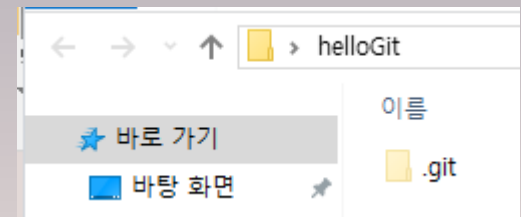
- ◆ clone : 기존에 있는 프로젝트를 복사하여 새로운 디렉토리에 생성
- ◆ init : 현재 디렉토리를 버전 관리할 프로젝트로 생성

- ✓ 바탕화면에 버전 관리할 프로젝트(폴더) 생성



- ✓ 프로젝트를 생성할 폴더로 경로 이동
- ✓ 경로 이동 후 git init 명령으로 프로젝트 생성 및 확인

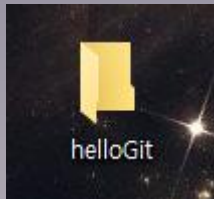
```
C:\Users\Administrator>cd Desktop/  
C:\Users\Administrator\Desktop>cd helloGit/  
C:\Users\Administrator\Desktop\helloGit>git init  
Initialized empty Git repository in C:/Users/Administrator/Desktop/helloGit/.git/
```



Git 저장소 생성

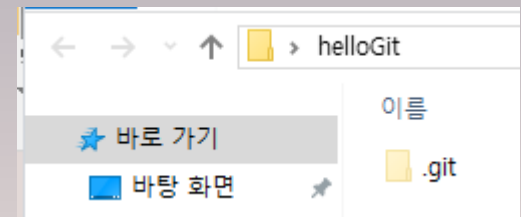
- ◆ clone : 기존에 있는 프로젝트를 복사하여 새로운 디렉토리에 생성
- ◆ init : 현재 디렉토리를 버전 관리할 프로젝트로 생성

- ✓ 바탕화면에 버전 관리할 프로젝트(폴더) 생성



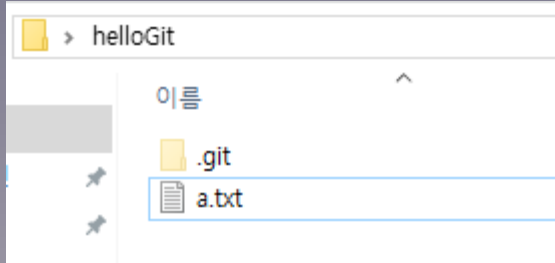
- ✓ 프로젝트를 생성할 폴더로 경로 이동
- ✓ 경로 이동 후 git init 명령으로 프로젝트 생성 및 확인

```
C:\Users\Administrator>cd Desktop/  
C:\Users\Administrator\Desktop>cd helloGit/  
C:\Users\Administrator\Desktop\helloGit>git init  
Initialized empty Git repository in C:/Users/Administrator/Desktop/helloGit/.git/
```



Git 저장소 생성

- ✓ 프로젝트 생성된 폴더에 내용이 없는 텍스트 파일 추가



- ✓ git status로 현재 상태 확인

```
C:\Users\Administrator\Desktop\helloGit>git status
On branch master

No commits yet

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

        a.txt

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

- ◆ branch : 분기한 프로젝트 버전 (master branch는 아직 분기를 하지 않은 상태)
- ◆ commit : 변경이력을 git에 저장해 놓은 것으로 프로젝트를 변경함에 따라 바뀜
- ◆ add : 변경 사항들 중에 어떤 것을 commit할 건지 정하여 추가하는 작업

Git 저장소 생성

- ✓ 추가한 텍스트 파일을 commit list에 등록

```
C:\Users\Administrator\Desktop\helloGit>git add a.txt

C:\Users\Administrator\Desktop\helloGit>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   a.txt
```

- ✓ git commit 명령을 통해 list에 등록된 파일을 프로젝트에 업데이트

```
C:\Users\Administrator\Desktop\helloGit>git commit
Aborting commit due to empty commit message.
```

관리자: Git CMD (Deprecated) - "C:\Program Files\Git\cmd\git.exe" commit

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   a.txt
#
```


Git 저장소 생성

- ✓ 입력 모드로 변환하여 create a.txt 입력하고 저장

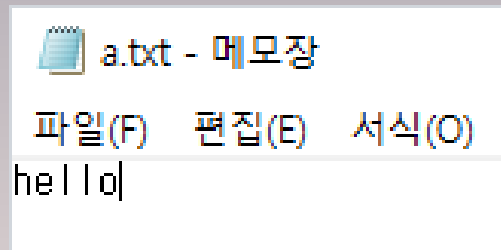
```
C:\Users\Administrator\Desktop\helloGit>git commit
[master (root-commit) 2bdf134] [create a.txt]
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 a.txt
```

- ✓ git log 명령을 통해 commit이 완료된 변경 이력을 확인

```
C:\Users\Administrator\Desktop\helloGit>git log
commit 2bdf13446795922117e06fb2cf77c0f745eb17cc (HEAD -> master)
Author: mg kim <phenix235qor@naver.com>
Date: Thu May 16 10:30:42 2019 +0900

    [create a.txt]
```

- ✓ 텍스트 파일에 변화를 주고 다시 commit 실행



```
C:\Users\Administrator\Desktop\helloGit>git add a.txt
C:\Users\Administrator\Desktop\helloGit>git commit
```

Git 저장소 생성

- ✓ 입력 모드로 변환하여 change a.txt 입력하고 저장
- ✓ git log 명령을 통해 commit이 완료된 변경 이력을 확인

```
C:\Users\Administrator\Desktop\helloGit>git commit
[master c95aab9] change a.txt
1 file changed, 1 insertion(+)

C:\Users\Administrator\Desktop\helloGit>git log
commit c95aab99fdb176b62260295db838c1dd9b7098bd (HEAD -> master)
Author: mg kim <phenix235gor@naver.com>
Date: Thu May 16 10:35:49 2019 +0900

    change a.txt

commit 2bdf13446795922117e06fb2cf77c0f745eb17cc
Author: mg kim <phenix235gor@naver.com>
Date: Thu May 16 10:30:42 2019 +0900

    [create a.txt]
```



끝