##########################################################################################################

**Git 교과서서**

==========================================================================================================

이호진 / 2023.08 / 길벗

005.13 이95ㄱ

##########################################################################################################

저자 깃허브: <https://github.com/jinygit>

더북: <https://thebook.io/080212>

학습 사이트: <https://git.jiny.dev>

# **1. 깃과 버전 관리**

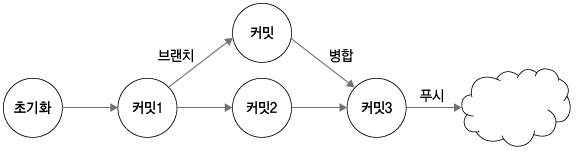
분산형 버전 관리 시스템(DVCS, Distributed Version Control System)

저장소가 여러 개 있고 여러 저장소에 각 버전별 소스를 개별 보관한다. 분산 저장소는 P2P 방식으로 공유하며, 각 개발자에게 공유 가능한 저장소 사본을 제공. 서버는 각 저장소 자료를 동기화하고 중개하는 역할만 수행. 따라서 메인 서버에 문제가 생기더라도 지속적으로 개발이 가능

대표적인 것이 Git과 Mercurial

머큐리얼은 파이썬 언어로 개발된 무료 CVS. <https://mercurial-scm.org>

## **1.4 깃의 동작 한눈에 보기**



초기화: 폴더를 깃 저장소로 변경

커밋: 변경된 코드의 이력을 기록

브랜치: 분리 격리된 코드 이력을 기록

병합: 기존 이력과 분리된 이력을 통합

푸시: 로컬 저장소의 이력을 서버로 전송 및 동기화(또는 공유)

# **2. 깃과 소스트리 설치 및 환경 설정**

## **2.2 소스트리 설치**

* atlassian.com 사이트에 계정 생성 먼저 해야 함
* <https://www.sourcetreeapp.com/> 사이트 접속하여 최신 버전 다운로드 후 실행
* Bitbucket 아이콘 클릭
* 앞서 생성한 계정으로 로그인
* 등록 완료 메시지 표시됨. 다음을 눌러 계속 진행
* Mercurial 체크 박스에 체크를 해제하고 다음 진행
* 첫번째 줄은 생성된 github의 main 이름과 동일하게 입력. 두번째 줄은 아틀라시안 계정이 기본값으로 들어가 있으므로 그대로 사용. 다음 진행
* SSH 키를 불러오느냐는 물음에는 우선 아니오 선택
* 원격 저장소 화면이 뜨게 되고, 첫 설치이므로 비트버킷에 생성된 원격 저장소 목록이 없어 아무것도 보이지 않음. 정상임

# **3. 깃 개념 잡기**

## **3.1 깃 로컬 저장소(repository) 생성**

먼저, 로컬 저장소 생성을 먼저 한다.

로컬 저장소이니 당연히 자신의 PC에 폴더를 만들고 그 폴더를 로컬 저장소로 사용하게 만든다는 말.

생성하는 기본적인 방법은 git 설치 후 제공되는 git bash를 통해 명령어 입력을 하여 생성하는 것이다.

소스트리를 설치했으니 당연히 소스트리를 이용하여 저장소를 생성하는 방법도 있다.

깃 저장소로 사용하기 위해 D:\915.git\_repo 폴더를 이미 생성해 두었다.

git을 설치한 상황이니, 해당 폴더에 마우스 오른쪽 클릭 후 'Open Git Bash here'를 선택한다.

깃 배시 터미널이 열린다.

이제 해당 폴더를 저장소로 초기화 한다. 이 작업이 수행되면 해당 폴더에 git 동작을 위한 숨겨진 영역들이 생성되는 등의 초기화 작업이 진행된다.

**$ git init .**

**Initialized empty Git repository in D:/915.git\_repo/.git/**

실행하면 위와 같은 결과가 나온다. 이 폴더에는 .git 이라는 숨겨진 폴더가 생성된다. 이 .git 폴더는 매우 중요하므로 만약 저장소를 통째로 복사하려 한다면 반드시 .git 폴더도 같이 복사해야 한다. 그렇지 않으면 그 저장소의 모든 이력 내용이 없어진다.

위와 같은 작업을 소스트리에서도 할 수 있다.

소스트리 실행 후,

* 'Local' 아이콘 클릭
* 'Create' 클릭
* 첫번째 행에는 생성할 경로 및 폴더명 입력
* 두번째 행에는 자동으로 폴더명 입력되므로 무시
* 세번째 행에 'Git' 선택

하고 [생성] 버튼 누르면 된다..

간혹, "대상 디렉토리에 문제가 발생했습니다" 라는 알림 팝업창이 뜨게 되고 계속 진행할 거냐고 묻는데 '예' 선택하고 진행 시킨다.

우리는 이미 깃 배시로 저장소를 생성했으므로 이를 소스트리에 추가하기만 하면된다.

소스트리 실행 후,

* 'Local' 아이콘 클릭
* 'Add' 클릭
* 첫번째 행에는 [탐색] 버튼 눌러 생성한 저장소 폴더 지정
* 두번째 행에는 자동으로 폴더명 입력되므로 무시
* 세번째 행에 '[루트]' 선택. (처음 생성이므로 루트 밖에 없음)

하고 [추가]버튼 누르면 된다.

간혹, "대상 디렉토리에 문제가 발생했습니다" 라는 알림 팝업창이 뜨게 되고 계속 진행할 거냐고 묻는데 '예' 선택하고 진행 시킨다.

# **5. 서버**

깃허브에 원격 저장소(remote repository) 생성을 해보자.

당연히 github에 회원가입을 하여 본인의 계정을 만들어야 한다. 이 문서에서는 이미 만들었다 가정하며, 만든 깃허브 주소는 <https://github.com/KimChoongHeon> 이다.

앞서 생성한 로컬 저장소 **915.git\_repo** 과 동일한 이름의 원격 저장소를 생성한다. 깃허브에 로그인하여 본인의

[github.com/KimChoongHeon](http://github.com/KimChoongHeon) 에 접속한다.

원격 저장소를 생성할 때에는 공개용 저장소로 생성할 것이다. 공개용 저장소 생성은 무제한으로 생성이 가능하나, 비공개용은 제약이 따르며 제약을 없애기 위해서는 비용을 github.com에 지불해야 한다.

[github.com/KimChoongHeon](http://github.com/KimChoongHeon) 에 접속한 상태에서 보이는 대시보드의 우측 상단에 + 표시가 보인다. 그 옆에 아래표시를 클릭하면 매뉴가 dropbox 형태로 보이는데 'New Repository'를 선택한다.

Repository name 란에 '**915.git\_repo**' 를 입력한다.

필요에 따라 Description에 정보 입력하거나, public 또는 private으로 사용할지 선택하거나, README 파일 추가할지, .gitignore 추가할지, license 정책 설정 등을 하고 'Create Repository' 클릭한다.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

## **이미 만들어 놓은 원격 저장소와 이미 만들어 놓은 로컬 저장소 연결**

이제 개인 PC의 로컬 저장소 **915.git\_repo** 폴더와 github에 생성된 원격 저장소 **915.git\_repo** 폴더를 연결해 보자.

소스트리를 실행하고 이미 생성한 **915.git\_repo** 로컬 저장소를 연다.

소스트리 맨 상단에서 '저장소 -> 원격 저장소 추가' 를 선택한다.

저장소 설정 팝업창이 뜨는데 '추가' 버튼을 누르고

* 원격 이름은 디폴트 원격을 체크하여 origin으로 만든다.
* URL/경로에 원격저장소 주소(<https://github.com/KimChoongHeon/915.git_repo>)를 입력한다.
* Remote Account는 Bitbucket 을 선택한다.

확인 버튼 눌러 추가를 완료하고 난 후, 다시 확인 버튼을 눌러 설정을 종료한다.

소스트리 왼쪽 사이드바의 '원격' 하위에 origin 이라는 하위 아이템이 생성된 것을 볼 수 있다.

이제 로컬 저장소(**D:\915.git\_repo**) 폴더에 임의의 파일 하나를 복사한다. 파일명은 test.txt 이다.

소스트리 왼쪽 사이드바 상단에 있는 파일 상태를 클릭한다.

복사한 파일이 스테이지에 올라가지 않은 파일 부분에 있는 것을 볼 수 있다.

test.txt 파일을 클릭하고 파일이 있는 행 맨 우측에 '+'를 클릭하거나 test.txt 파일을 드래그하여 스테이지에 올라간 파일 쪽으로 이동시킨다.

소스트리 맨 하단에 커밋 메시지를 입력한다. (변경된 내용을 설명하는 글 쓰면 됨)

이어서 [커밋] 버튼을 눌러 로컬 저장소에 커밋을 완료한다.

소스트리의 상단에 있는 'Push'를 클릭하여 푸시 대화 상자가 나타나면

* 푸시할 브랜치를 확인한다(일반적으로 기본 브랜치는 main 또는 master 를 체크)
* [Push] 버튼 누른다.
* 맨 처음으로 push 하는 경우에는 본인 github 계정으로 로그인하라고 뜨는데 로그인 하면 된다.

이제 github.com의 본인 계정에 가 보면 파일들이 업로드 되어 있는 것을 볼 수 있다.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

## **이미 만들어 놓은 원격 저장소에 새로운 로컬 저장소를 생성하고 연결**

이번에는 새로운 로컬 저장소를 생성하고 난 후 이미 만들어 놓은 **02.data\_n\_info** 라는 원격저장소에 연결하는 방법을 정리한다.

원격 저장소 이름과 동일하게 **D:\02.data\_n\_info** 폴더를 생성한다.

(깃 배시에서 생성해도 되고 탐색기에서 생성해도 됨)

깃 배시에서 생성한 폴더를 깃으로 초기화 한다.

**$ git init**

저장소의 소개 페이지 하나 생성.

**$ echo "# 02.data\_n\_info" >> README.md**

만든 README.md 파일을 커밋한다.

**$ git add README.md**

**$ git commit -m "make readme file and first commit"**

이제 원격 저장소와 연결작업을 해야 한다.

명령어 포맷은 git remote add <원격저장소별칭> <원격저장소 URL>

<원격저장소별칭> 은 본인이 원하는 별칭을 입력하면 된다. 보통은 origin 을 입력

<원격저장소 URL> 은 말 그대로 원격저장소 주소 입력. 마지막에 .git 있어야 한다.

**$ git remote add origin https://github.com/KimChoongHeon/02.data\_n\_info.git**

원격 저장소를 add 하고 나면 원격 브랜치는 자동으로 생성된다.

이제 아까 commit 했던 README.md 파일을 push해 보자.

**$ git push origin master**

github.com의 **02.data\_n\_info** 원격 저장소를 확인해 보면, README.md 파일이 있는 것을 볼 수 있다.

이어서 소스트리를 구동시키고, New Tab을 생성한 후에 소스트리 상단에 있는 Local 클릭하고 Add를 눌러

[탐색] 버튼을 클릭하고 **D:\02.data\_n\_info** 폴더를 지정한다.

[추가] 버튼을 눌러 추가한다.