

GIT 세미나 ▼



통합검색

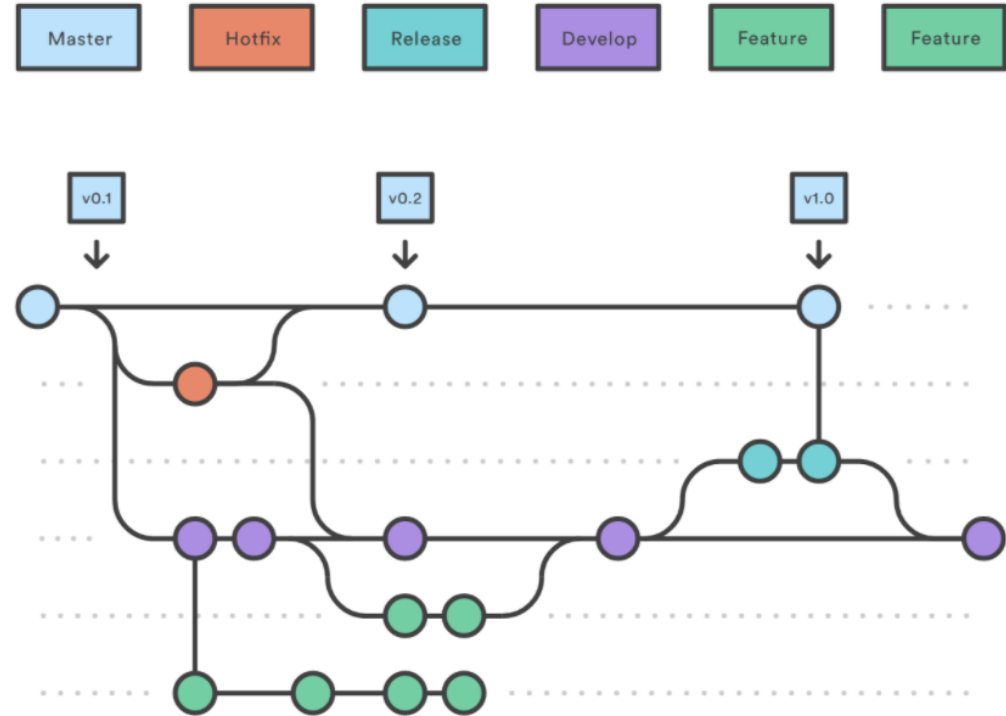
발제자 : 유영재

세미나 목표

1. Git의 원리를 이해함으로써 Git과 친숙해지기
2. 온라인 영상 혹은 구글링을 통해서도 학습이 가능한 정도로 만들기



<https://github.com/git/git>



<https://learngitbranching.js.org/?locale=ko>

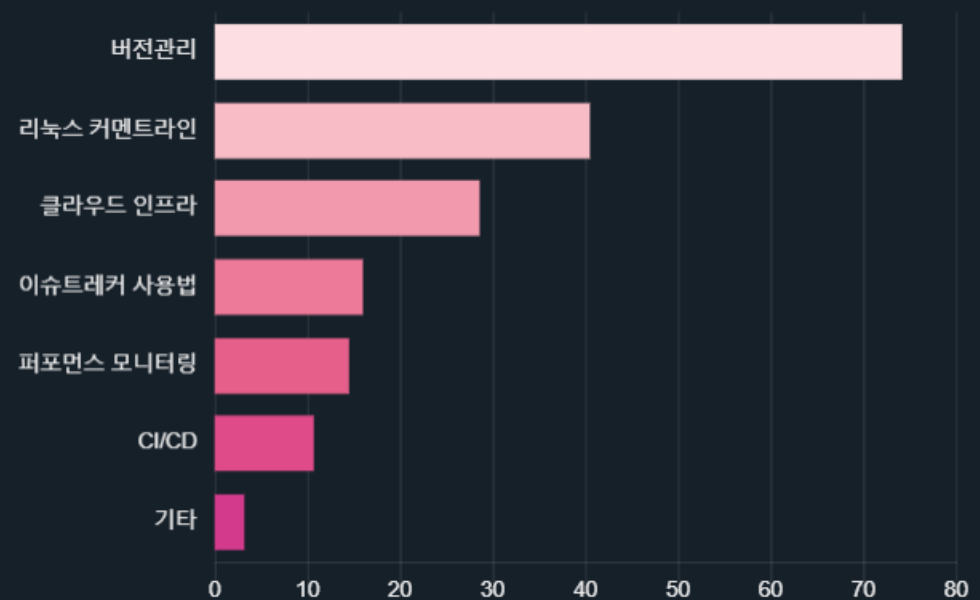


Q. Git 꼭 사용해야 하나?

최소한 알고 지원했으면 하는 것들

**버전 관리, CLI(Linux),
클라우드 서비스에 대한
지식이 TOP3로 꼽혔습니다.**

지원자가 최소한 알고 왔으면 하는 것 (단위:%)



개발자 채용 프로세스에 참여하고 있는 1582명이 이 질문에 답변했습니다.

Q. Git 꼭 사용해야 하나?

Q. 주로 사용하는 코드리뷰 툴/서비스는 무엇인가요?

코드 리뷰 툴은 대부분 Github를 사용합니다. 그러나 3%는 눈으로 모니터를 보는 등 툴 없이 코드 리뷰를 하고 있습니다.

- Github(58.2%)
- GitLab(16.7%)
- Bitbucket(11.8%)

응답자 전체

개발자 전체

솔루션/SI

콘텐츠

이커머스

금융

하드웨어

포털

게임

사용하는 코드리뷰 서비스(단위: %)

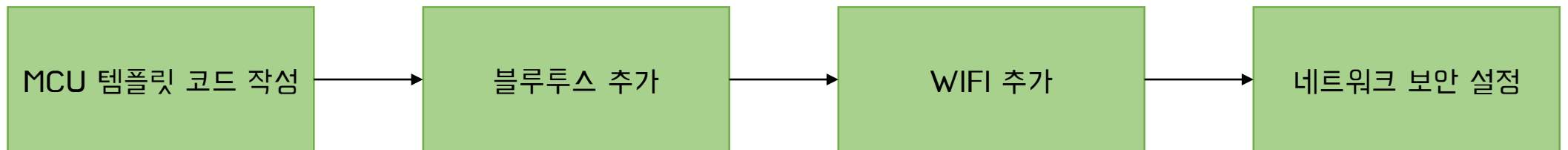
Q. Git 꼭 사용해야 하나?

1. 버전 관리

Q. 용량이 폴더 하나당 10G라면?

작업폴더 1	2020-05-20 오후 8:33	파일 폴더
작업폴더 1.1	2020-05-20 오후 8:33	파일 폴더
작업폴더 1.2	2020-05-20 오후 8:33	파일 폴더
작업폴더 2.0	2020-05-20 오후 8:33	파일 폴더
작업폴더 2.1	2020-05-20 오후 8:33	파일 폴더

Q. 오디오 장치를 개발하는데 보안기능까지 완료했다. 그런데 갑자기 블루투스에 LCD만 추가한 장치를 만들어 달라는 외주가 들어왔다. 기능별로 캡슐화를 잘했다고 생각하는데, WIFI, 보안 관련 코드가 너무 많을뿐더러 여러 소스파일에 추가되어 있다. 어떻게 삭제하지?

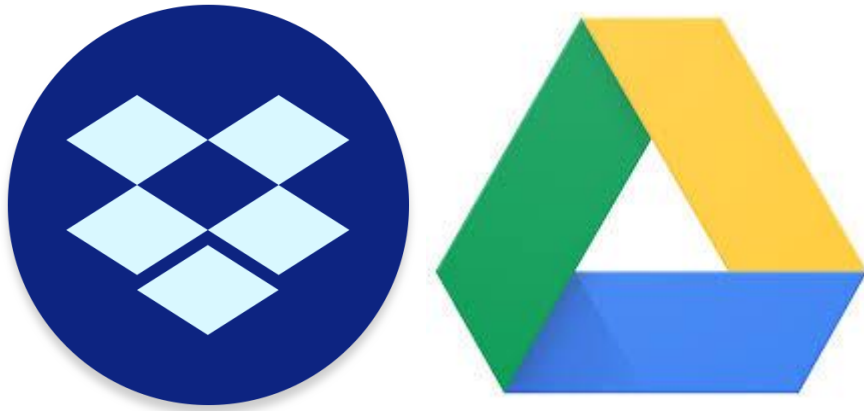


Q. Git 꼭 사용해야 하나?

2. 백업

컴퓨터 상황, 혹은 오류 상황이 언제 들이닥칠지 모르는 것이다. 버전관리를 통한 백업은 필수다

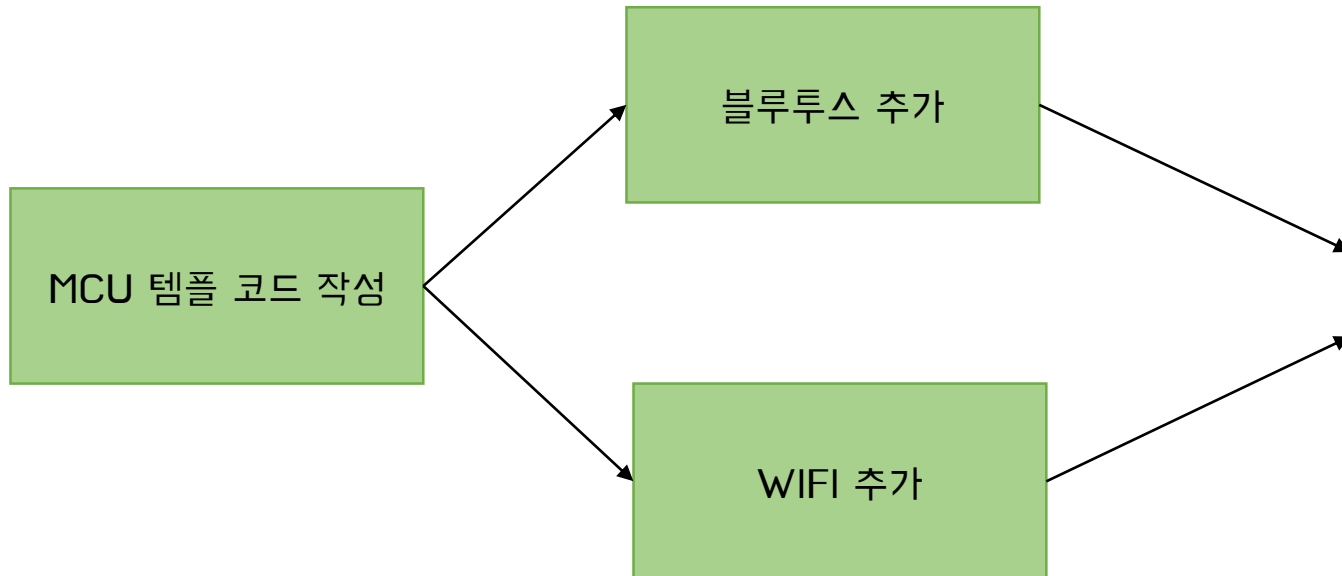
Q. 상용 클라우드를 사용해도 되지 않나?



1. 유지비
2. 파일 사이즈
3. 버전관리

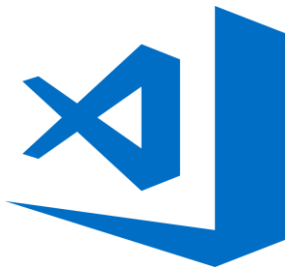
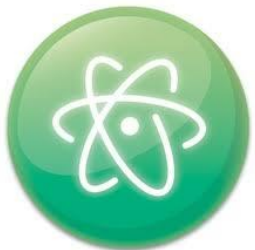
Q. Git 꼭 사용해야 하나?

3. 협업



1. 서로 개발한 파일을 합쳐야 하려면 어떻게 하지..?
2. 만일 같은 부분을 수정했다면?
3. 팀장인 내가 팀원이 수정한 부분에 대해서만 코멘트를 하고 싶다면?

클라이언트



서버



Q> 그러면 어떤 것을 써야하나?

1. Git 클라이언트

- 터미널 콘솔 vs GUI
- 초기에는 GUI, 나중에는 콘솔로 넘어오는 것을 추천 (+Linux)
- GUI에서 사용할 수 없는 명령도 있고, 속도 (GUI < 콘솔)

2. Git 서버

- 자신에게 맞는 서비스(오픈소스 프로젝트)를 제공하는 사이트
- 개인 서버를 두는 것도 하나의 대안

(참조: <https://github.com/ingbeeedd/raspberrypi-git-server>)

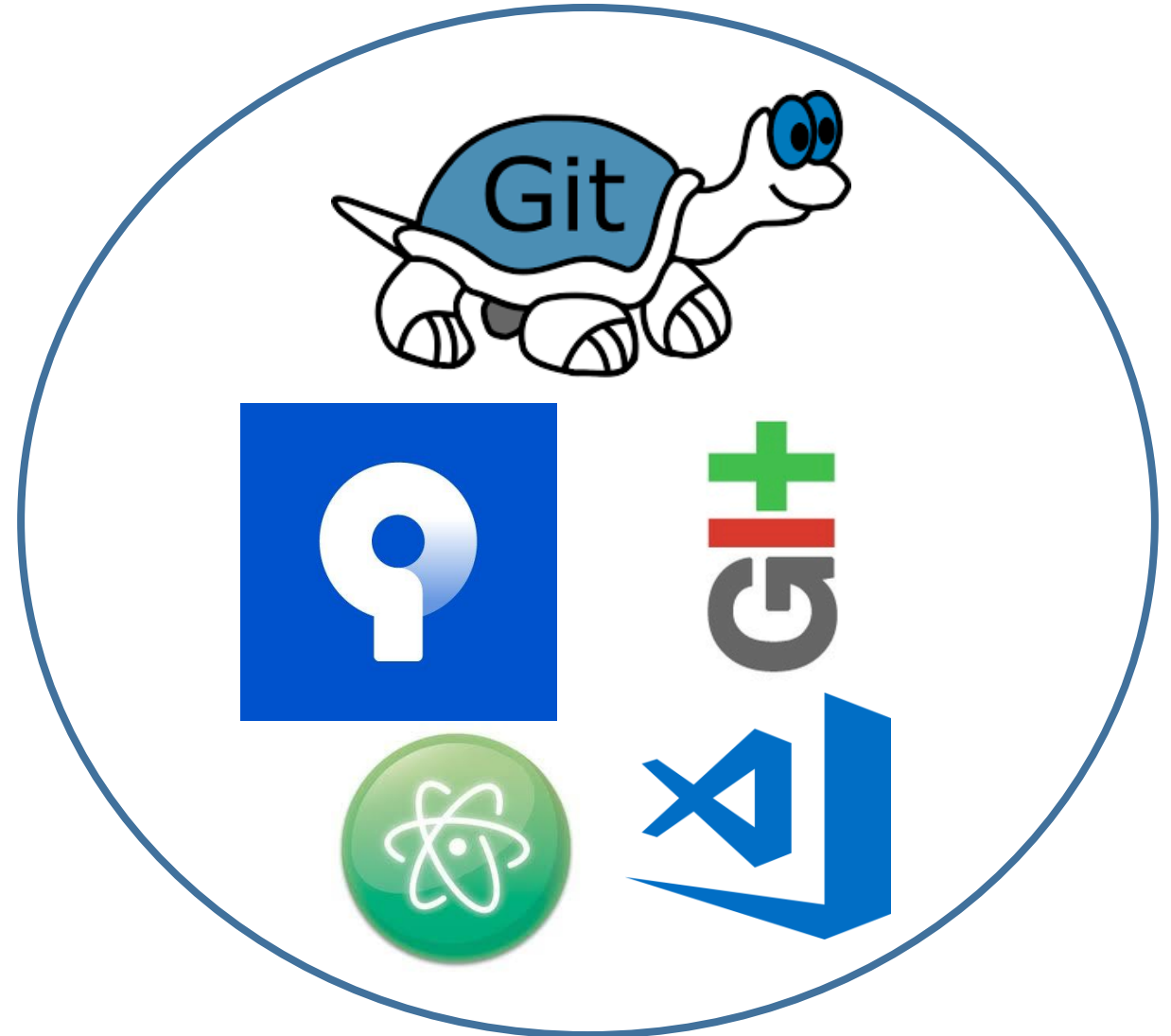
옵션!

Git 구성

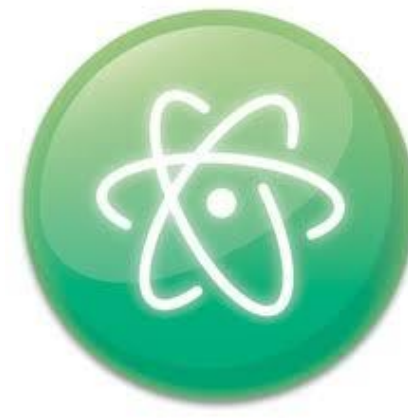
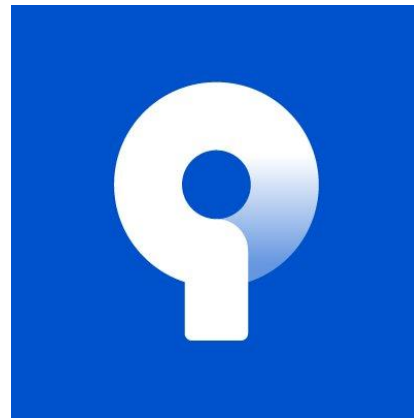


필수!

+



Git 구성 프로그램 설치



Git 초기화 후 숨겨진 디렉토리 탐색

```
$ mkdir git-test && cd git-test
```

```
$ git init
```

```
$ ls -al
```

```
$ cd .git
```



Git 원리 분석

<https://opentutorials.org/course/2708/15212>

Atom으로 커밋 만들어보기

과정 블로그 참조

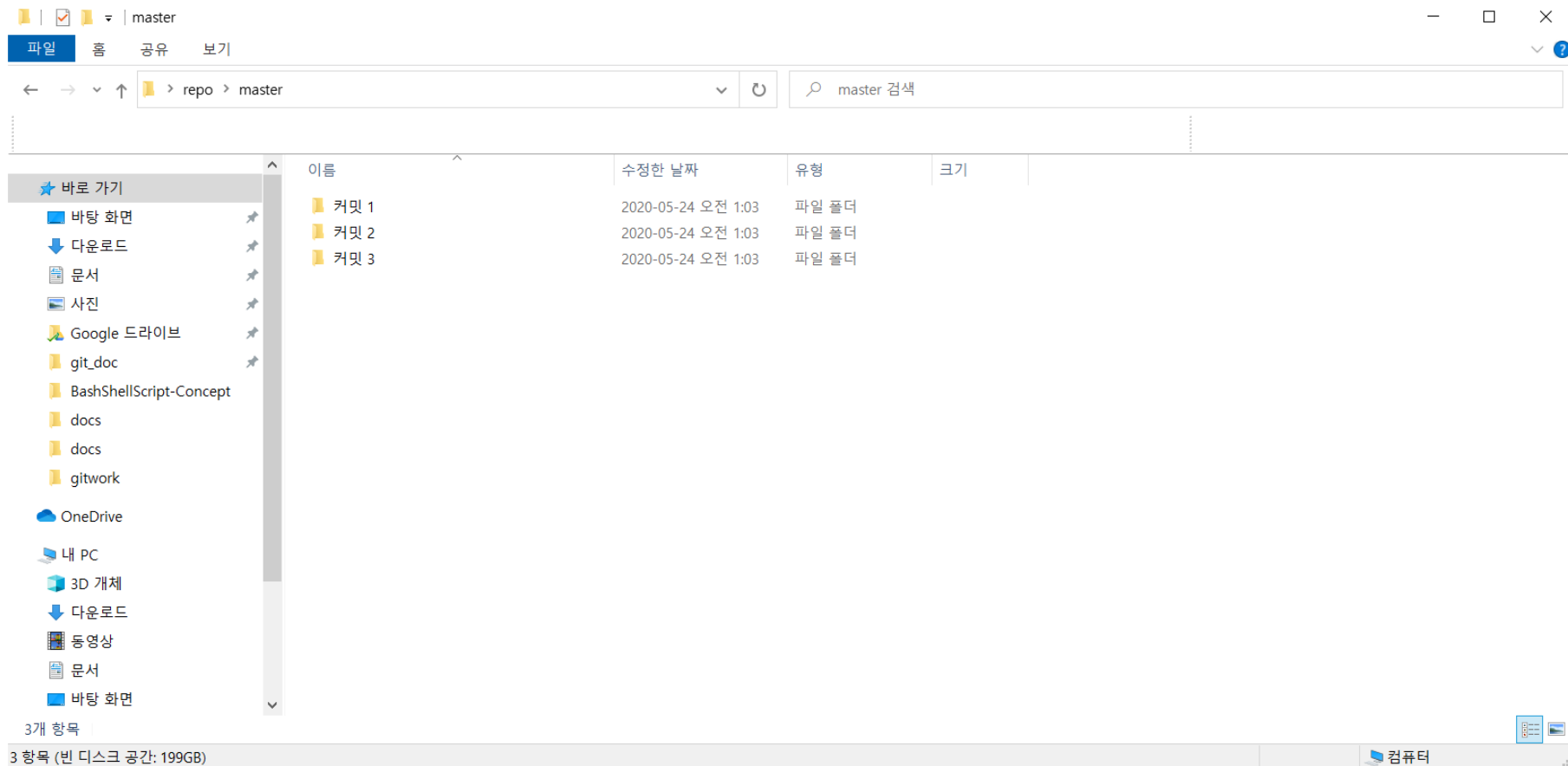


Bash로 커밋 이어서 만들어보기

과정 블로그 참조



실제 윈도우 폴더 작업으로 commit 이해하기



HEAD 포인터로 커밋 이해하기

HEAD 포인터는 현재 커밋의 위치를 가리키는 중요한 위치다. 그리고 파일은 HEAD가 가리키는 커밋(스냅샷)으로 상태를 유지하고 있다. HEAD를 옮겨보도록 한다
Git에서 커밋의 이름은 sha-256 알고리즘으로 만들어진 값을 말한다

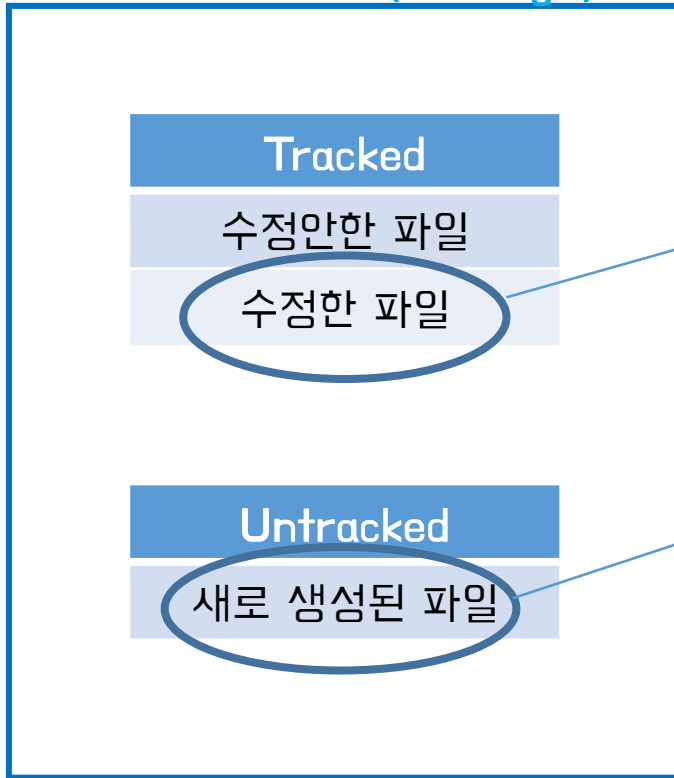
```
$ git checkout <커밋해쉬>
```

확인 후 master로 돌아오도록 한다

```
$ git checkout master
```

Git status 이해

작업 디렉토리(unstage)



Git이 추적, 커밋에 올라갈 수 있는 자격 충족!

Git이 추적 못하지만, 커밋에 올라갈 수 있는 자격 충족!

stage (커밋 대기 장소)

원하는 파일만 선택

\$ git add ???

Git status 이해

stage (커밋 대기 장소)

원하는 파일 완료

main.c
main.h
....

commit 메시지를 포함한 commit

commit은 일종의 스냅샷 기능!



Github 원격 서버와 연동

과정 블로그 참조

Github 프로젝트 협업하기 (1) - Fork, Pull Request

과정 블로그 참조

Github 프로젝트 협업하기 (2) - Collaborator

과정 블로그 참조

감사합니다

PPT 템플릿 자료는 <http://pptbizcam.co.kr/>에서 이용했습니다.