

개발보조도구

For Computer Vision Projects

1 Day

조윤실

CONTENTS

- 01** SW 개발도구 소개
- 02** 코드관리 도구 – Git, Github
- 03** 문서관리 & 협업 도구 - Notion
- 04** 협업도구 –Slack
- 05** 서비스관리 도구 - Docker



- SW 개발에 필요한 도구를 살펴보고 개발 단계를 이해할 수 있다.
- 형상관리(버전관리)를 이해하고 Git & Github를 사용해 내 코드를 관리할 수 있다.
- 교육과정에서 학습한 내용과 특이사항을 Notion을 이용하여 기록하고 공유할 수 있다.
- Slack을 팀내 소통 보조도구로 사용한다.

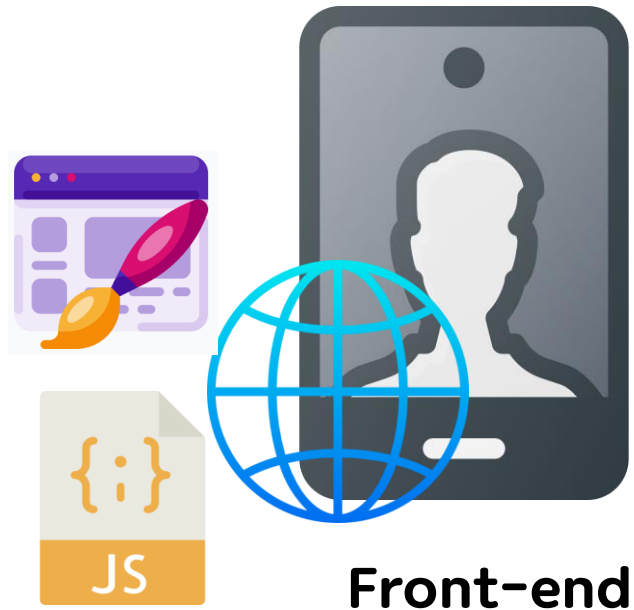
01

SW 개발도구

SW 개발 분야

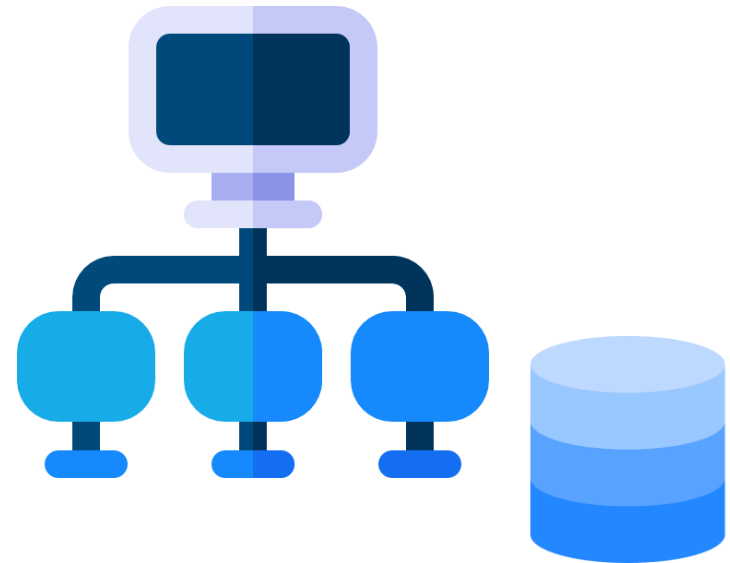
■ 개발자 분류

- 업무 분야별, 서비스영역별 기준이 다를 수 있으나 일반적으로 많이 구분되는 개발자 분류
- 참고 동영상: https://www.youtube.com/watch?v=G7crMao49_I



Front-end

Full-stack



Back-end

SW 개발도구

■ SW 개발도구

- SW 개발과정(Software Development Life Cycle) 전체에 걸쳐서 사용되는 도구(Tool)
- 비즈니스 프로세스를 수행 및 조사하고 소프트웨어의 개발 프로세스를 문서화하며 모든 프로세스를 최적화하는 데 사용됨 → **프로젝트 결과 생산성 향상 & 워크 플로우 유지관리 용이**

■ SW 개발과정 단계



SW 개발도구...

■ SW 개발단계별 도구

- <https://www.swbank.kr/helper/tool/toolMain.do>
- 각 회사별 목적별 필요에 따라 개발단계별 도구는 달라질 수 있다.

분류	도구명	라이선스	언어	활용지침	동영상	연계자료
요구사항관리	JFeature	CPL		PDF	▶	
	JRequisite	EPL		PDF		
	OSRMT	GPL2		PDF		
	ProrR	EPL		PDF	▶	
	Trello	기타		PDF		
설계/모델링	ArgoUML	BSD		PDF		
	DB Designer	GPL2		PDF	▶	
	ERMaster	APL2	JAVA	PDF	▶	
	Papyrus	EPL		PDF	▶	
	PlantUML	GPL3		PDF		
	StarUML	GPL		PDF	▶	
	Violet UML Editor	GPL		PDF		

전체 내용 확인

SW 개발도구...

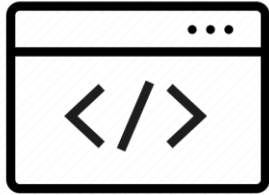
■ 생산성 도구

- 업무 분야별, 목적별, 사용하는 OS별 도구는 다를 수 있음 → 유료 제외하고 소개



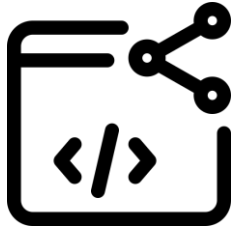
Web browser

Chrome
Firefox
Edge
IE
Safari



Code editor

Visual Studio Code
Pycharm
Sublime 3
Evernote



Code share

Git/Github
Google Drive
SVN



Documentation

Google Doc.
MS Office
Notion



Image editor

Pixlr
MS 기본명령어



Team Project

Notion
Slack

02

코드 관리 도구
Git, Github

- 01. 형상관리
- 02. Git & Github
- 03. [실습] Git 사용해보기
- 04 [실습] Github 사용해보기

문서관리



형상관리

■ 형상관리

- 형상 관리(Software Configuration Management)는 소프트웨어의 변경사항을 체계적으로 추적하고 통제하는 것으로, 형상 관리는 일반적인 단순 버전관리 기반의 소프트웨어 운용을 좀 더 포괄적인 학술 분야의 형태로 넓히는 구간을 이야기함 – by 위키백과
- 소프트웨어 개발 생명주기 전반에 걸쳐 생성되는 모든 산출물의 종합 및 변경 과정을 체계적으로 관리하고 유지하는 일련의 개발 관리 활동

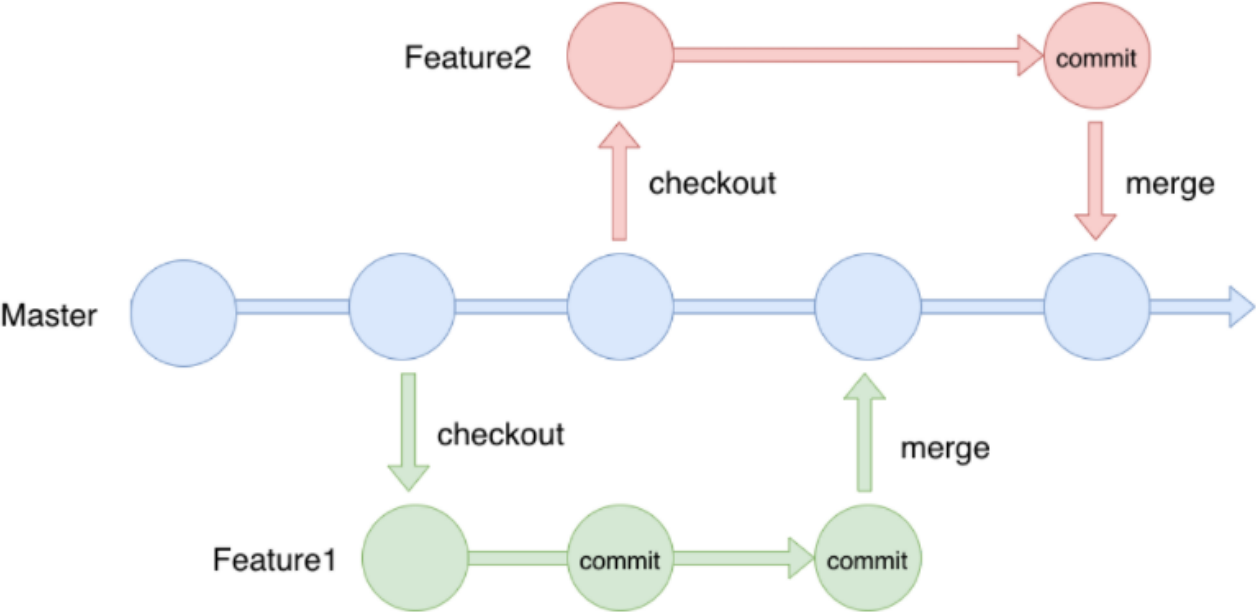
$$\begin{array}{ccccc} \text{변경 관리} & & + & & \text{버전 관리} & & = & & \text{형상 관리} \\ (\text{소스코드 변경 사항 관리}) & & & & (\text{변경사항을 '버전' 이란} & & & & (\text{프로젝트와 관련된} \\ & & & & \text{개념을 통해 관리}) & & & & \text{모든 변경사항 관리}) \end{array}$$

• 형상관리구조

- 형상 식별(Configuration **I**dentification), 형상 제어(Configuration **C**ontrol),
형상 감사(Configuration **A**udit), 형상 상태 보고(Configuration **S**tatus **A**ccounting)

형상관리 용어

- 형상관리 용어



[branch에서 merging하는 과정 사례]

용어	설명
Repository	저장소
Checkout	Repository에서 로컬로 프로젝트를 복사함
Commit	로컬의 코드를 Repository에 저장
Update	로컬에서 작업중인 코드를 Repository로 저장
Add	로컬에서 새로운 파일을 추가했을 때 Repository에 등록
Trunk	Root 프로젝트
Branch	Root 프로젝트로 부터 파생된 프로젝트. Branch를 생성하여 버전 관리가 가능
Merge	Branch에서 진행하던 작업을 Root 프로젝트와 합침

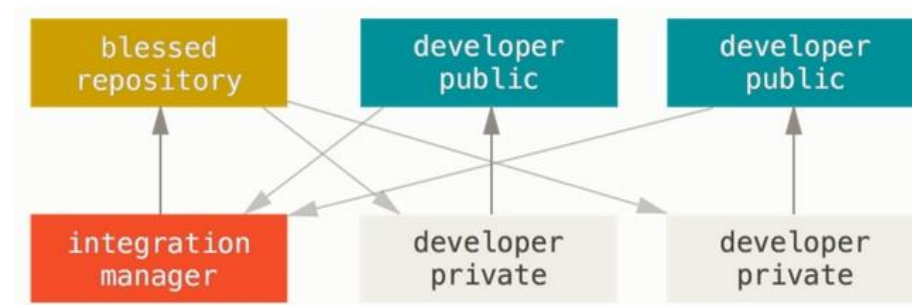
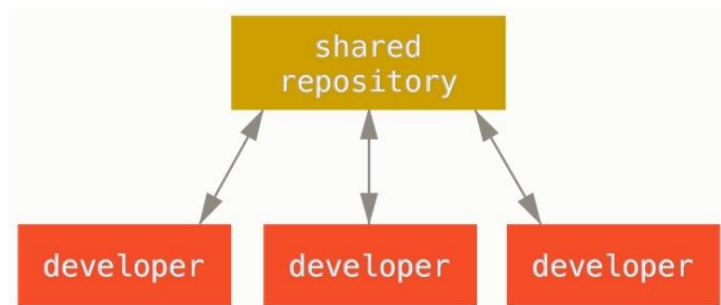
형상관리 도구

■ 형상관리 도구

용어	설명
CVS (Concurrent Version System)	90년에 출시된 무료 서버-클라이언트 형상관리 시스템. 파일 전체를 저장하는 것이 아니라 변경사항만을 저장함으로 용량을 적게 차지하지만 속도가 상대적으로 느림
SVN (Subversion)	형상관리/소스관리 툴의 일종. 중앙관리만을 지원. 다른 사용자의 커밋과 엇키지 않으며, 커밋 실패 시 롤백 기능을 지원. 안정성에 면에서는 CVS보다 낮다.
Git	리누스 토발즈가 2005년 리눅스 커널의 개발을 위해 만든 형상 관리 시스템. 분산형 버전관리 시스템 . Repository의 완전한 복사본을 로컬에 저장할 수 있음. 처리속도가 빠르지만 대용량 코드 관리에 부적절함
Github	Git을 웹호스팅을 통해 관리할 수 있게 해주는 무료 사이트
Perforce(P4D)	2014년 출시된 형상관리 툴, 코드와 바이너리 파일의 변환을 추적하기 위해 제작. 빠른 속도, 빠른 Merge가 가능하며 큰 리소스 관리에 좋음. 다만, 유료이고 파일명이 바뀌면 히스토리 추적이 곤란함

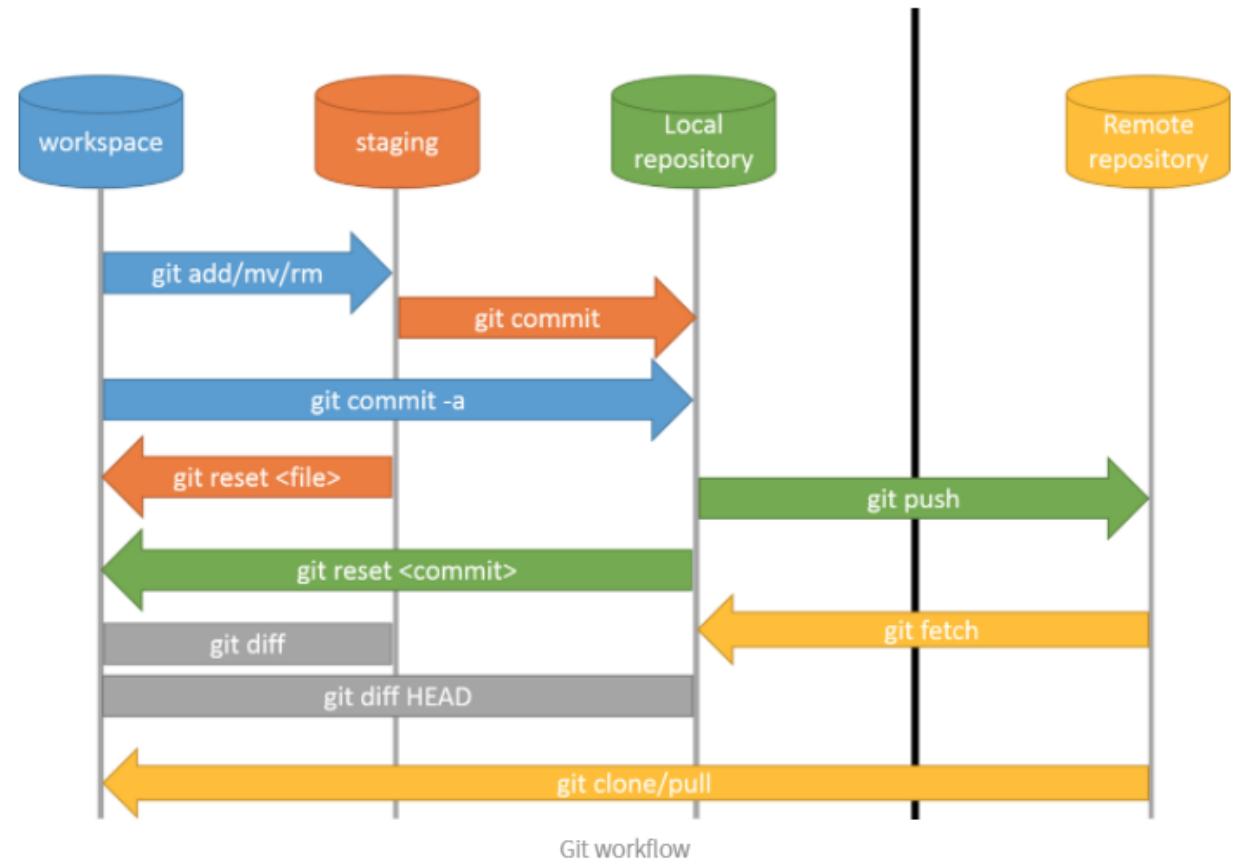
Git 특징 (Version Control System)

	SVN	Git
사용법	간편하고 어렵지 않다	다소 복잡하고 초보자가 이해하기 어렵다
기능	버전관리에 최적화된 간편한 기능	버전관리에 필요한 다양한 기능이 존재
프로세스	중앙 집중식	분산 관리식
소스충돌위험	매우 높음	권한 설정을 통해 충돌 위험 감소
저장소백업여부	저장소 백업이 용이하지 않음	Git 저장소만 있으면 리모트 복구 용이
다수작업관리	관리에 한계가 존재	프로그래머의 분산 작업에 매우 용이
작업내용복구	다소 불편	이전 리비전으로 복구가 매우 편리
브랜치생성	다소 불편	로컬에서 브랜치 생성 및 태그 생성이 편리



Github

- Github <https://github.com/>
 - 로컬에서 Git으로 관리하는 자료를 웹에서 다른 사람들과 공유하거나 백업하는 등 관리할 수 있게 해주는 무료 웹사이트
 - 다른 사람들과 공동으로 프로젝트를 진행할 때 협업할 수 있는 도구이면서 코드를 공유/복제(fork)하고 프로젝트에 참여(contribute)할 수 있도록 하는 소셜 네트워크 기능 제공함



Git 명령어

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

구분	명령어	설명
Git 설치	# sudo apt install git # git --version	→ 설치완료 → 설치된 버전 확인
저장소 생성	# git init	→ (디렉토리를 신규 생성하거나) 소스코드가 있는 디렉토리에서 해당 폴더를 git repository로 만들기 위한 명령
저장소 복제	# git clone <대상>	→ 특정 git repository를 가져옴
저장소에 추가	# git add <파일명> # git add *	→ 파일의 수정 사항들을 staged상태로 변경할 때 사용 → git repository에 새로 추가된 파일들을 staged 상태로 옮김
저장소에 확정	# git commit -m '커밋메시지'	→ Staged된 파일들을 저장소에 commit
브랜치 생성	# git branch <브랜치이름>	→ 브랜치 목록, 새 branch 로컬에 생성
체크아웃	# git checkout	→ 저장소에 있는 것으로 되돌림
변경사항 업로드	# git push	→ 원격 저장소에 업로드
갱신 및 병합	# git pull	→ 원격 저장소의 변경된 내용을 현재 디렉토리에 가져와(fetch) 병합(merge)
이력 보기	# git log # git status # git diff	→ Commit 내역 확인 → 현재 상태 확인 → 어떤 내용이 수정되었는지 확인
파일 삭제 & 이동	# git rm # git mv	→ 원하는 파일을 git repository에서 삭제 → 원하는 파일을 git repository 상에서 이동/파일명 변경

한글 참고: <https://medium.com/@joongwon/git-git-%EB%AA%85%EB%A0%B9%EC%96%B4-%EC%A0%95%EB%A6%AC-c25b421ecdbd>

[실습] Git 사용해보기



- Git 설치하기 (Windows/MacOs/Ubuntu)

- 참고 동영상: <https://www.youtube.com/watch?v=dBuyUHKvmEk>

1. Git 다운로드 <https://git-scm.com/>



Git-2.32.0.2-64-bit.exe

2. Git 설치

- ① Windows/MacOS → 프로그램 다운로드 & 설치

- ② Ubuntu

```
# sudo apt-get install git
```

```
# git config --global user.name "Yoonshil Jo"
```

```
# git config --global user.email "yswoola@gmail.com"
```

[실습] Git 사용해보기...



- 저장소 만들기과 Commit 하기

- 참고 동영상: <https://www.youtube.com/watch?v=W0aofTKVwJs>

1. 저장소 생성 : `git init`
2. 저장소에 파일 추가 : `git add <filename>`
3. 저장소에 변경 내용의 반영 : `git commit`
4. 저장소의 상태 확인 : `git status`
5. 커밋 로그 확인 : `git log`

[실습] Git 사용해보기...



- Branch 사용하기

- 참고 동영상: <https://www.youtube.com/watch?v=iiAlXe8H5y8>

1. branch 목록 보기 : `git branch`
2. 신규 branch 추가 : `git branch <브랜치명>`
3. 현재 branch 변경 : `git checkout <브랜치명>`
4. branch 내용 합치기 : `git merge <브랜치명>`
5. branch 삭제 : `git branch -d <브랜치명>`

[실습] Git 사용해보기...



- Merge, 충돌 해결

- 참고 동영상: <https://www.youtube.com/watch?v=vGRISK0IS-w>

1. 신규 branch 추가 : `git branch <브랜치명>`
2. 변경된 파일 저장 및 커밋 : `git add <filename> + git commit -m "커밋메시지"`
3. branch 변경 : `git checkout master`
4. 변경된 파일 저장 및 커밋 : `git add <filename> + git commit -m "커밋메시지"`
5. branch 내용 합치기 : `git merge <브랜치명>` → 충돌(Conflict)발생
6. 충돌된 내용 확인 및 파일 수정 후 커밋: `git add <filename> + git commit -m "커밋메시지"`
7. 변경된 내용 시각적으로 보여주기 : `gitk`

[실습] Github 사용해보기



- 원격 저장소(Github) 생성

- 참고 동영상: <https://www.youtube.com/watch?v=vGRISKOIS-w>

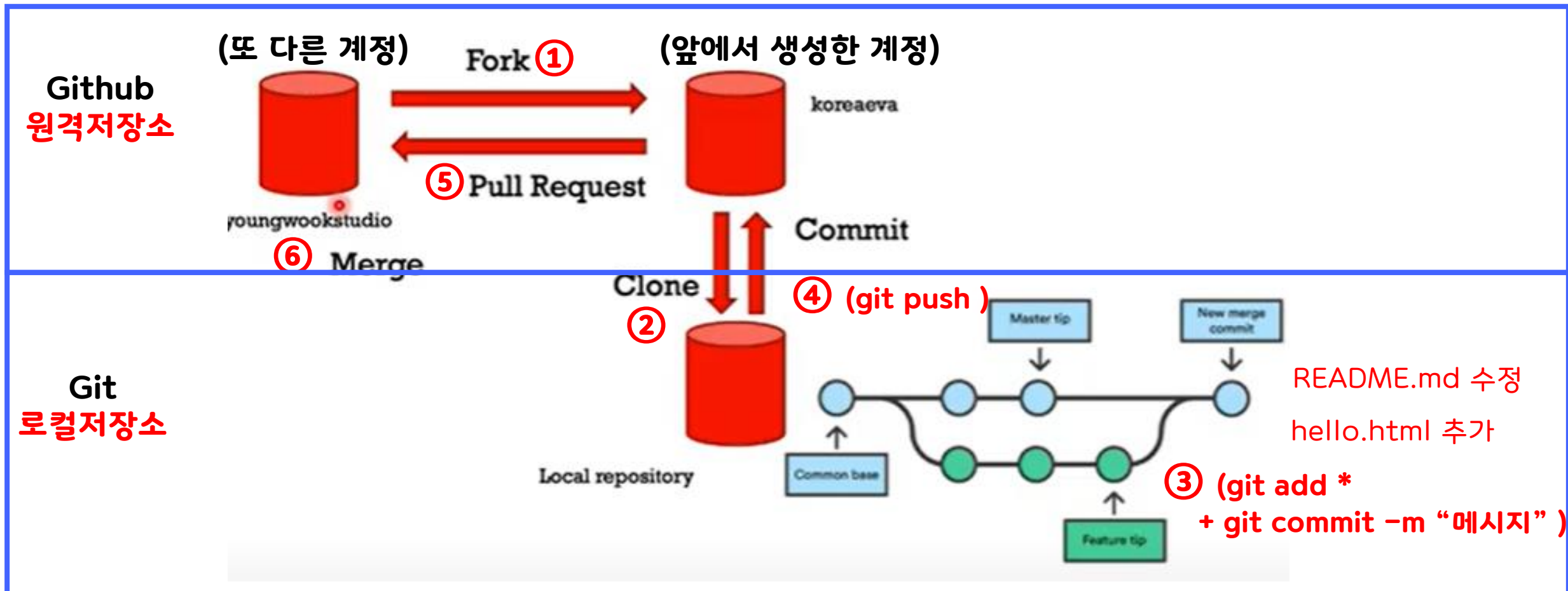
1. Github 사이트에 가입하기
2. Github 저장소(Repository) 생성하기
3. Github 저장소(Repository) URL 복사하기
4. (로컬)저장소 복사하기 : `git clone <Github URL>`
5. (로컬)README.md 변경하기 : `notepad README.md` → 내용 1줄 추가
6. (로컬)파일 신규 추가 : `notepad hello.md`
7. (로컬)변경된 파일 추가 및 커밋 : `git add *` + `git commit -m "커밋메시지"`
8. (로컬) 변경된 내용 Github로 푸시하기 : `git push`
9. Github 저장소(Repository) 페이지 reload(F5) 해서 push 된 내용 확인하기

[실습] Github 사용해보기...



- 원격 저장소(Github)와 PR(Pull Request)

- 참고 동영상: <https://www.youtube.com/watch?v=xlydJ53nnqY>



[미션] PC에 있는 파이썬 코드 Github에 올리기



1. 파이썬 과정에서 작성된 PC에 있는 자신의 파이썬 코드를 Git & Github를 이용하여 Github 사이트에 올려보세요.

단, repository 이름은 rapa_python (또는 자유롭게)

03

문서 정리 & 협업 도구

- 01. Notion 소개
- 02. Notion 특징
- 03. [실습] Notion 사용해보기

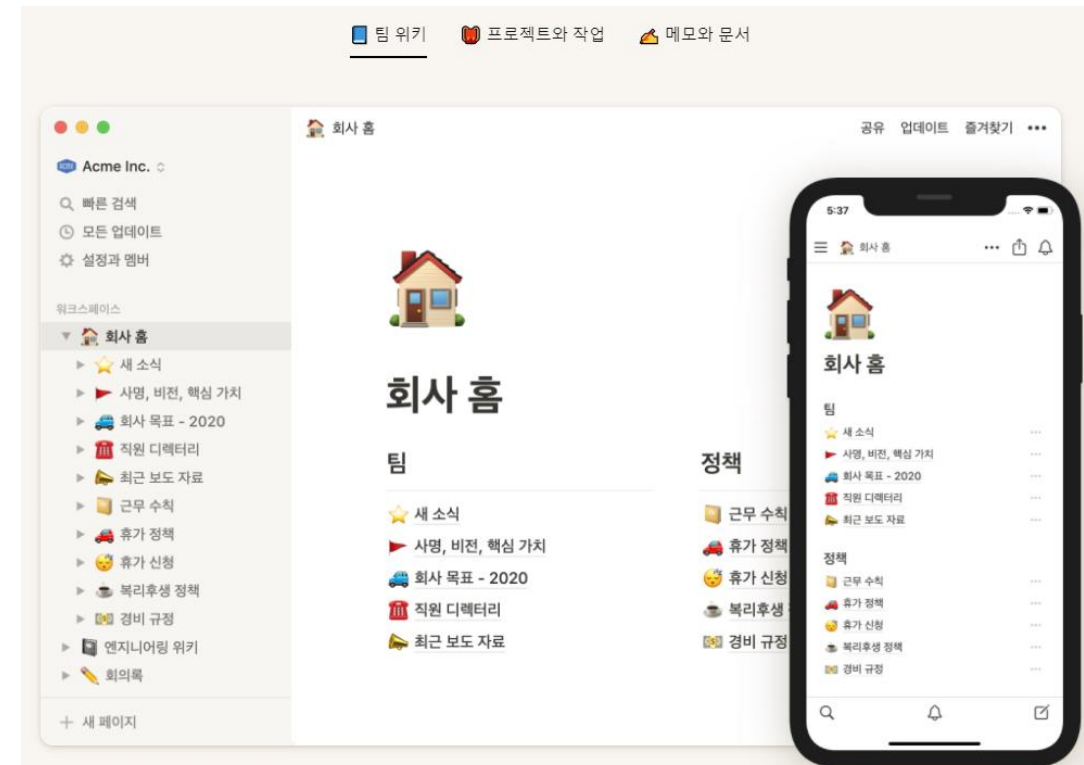
Notion 소개

■ Notion



<https://www.notion.so/>

- 노션은 문서 작성, 일정 관리, 업무, 프로젝트 등을 효율적으로 생성하고 관리할 수 있는 All-in-one 생산성 도구로 홈페이지 작성 및 협업 도구로 사용하기 좋은 도구임
- 샌프란시스코에 기반을 둔 스타트업인 Notion Labs Inc에서 2016년 에 출시
- 웹, MacOS, Windows, iOS, Android 사용가능
- 미국 다음으로 한국 사용자가 많음
- 국내 스타트업(Toss, Sandbox, Class101 등 에서 사용되고 있음 - by 위키피디아



Notion 특징

- Notion 요금제 <https://www.notion.so/ko-kr/pricing>



개인 요금제
US\$0

시작하기

일상의 모든 것을 정리할 수 있습니다.

개인 사용자에게 무료 제공

- ✓ 무제한 페이지와 블록
- ✓ 최대 5명의 게스트 초대
- ✓ 모든 기기에서 동기화
- ✓ API 베타



개인 프로 요금제
US\$4 매달

무료 체험

더 많은 작업을 수행하고자 하는 파워 유저들을 위한 요금제입니다.

개인 요금제의 모든 기능 +

- ✓ 무제한 파일 업로드
- ✓ 무제한 게스트
- ✓ 버전 기록



팀 요금제
US\$8 매달 멤버 1인당

무료 체험

모든 팀원이 한 워크스페이스에서 공동 작업할 수 있습니다.

개인 프로 요금제의 모든 기능 +

- ✓ 무제한 팀원 추가
- ✓ 공동 워크스페이스
- ✓ 고급 사용 권한
- ✓ 관리 도구



기업 요금제
영업팀에 문의 →

무료 체험

회사 운영에 필요한 제어와 지원을 제공합니다.

팀 요금제의 모든 기능 +

- ✓ SAML 싱글사인온(Single Sign-On)
- ✓ 사용자 프로비저닝(SCIM)
- ✓ 고급 보안
- ✓ 전담 관리자
- ✓ 맞춤형 계약

- 참고 동영상

연간 결제 ☐ 매달

<https://www.youtube.com/watch?v=UHHTFIE93ag>

[실습] Notion 사용해보기



1. 회원 가입하기 (구글계정)

<https://www.notion.so/ko-kr>

<https://www.youtube.com/watch?v=UHHTFIE93ag>

2. 페이지 만들기

- ① 기본 명령 사용하기
- ② 표 & 보드 사용하기
- ③ 캘린더 사용하기

3. 템플릿 사용하기

[미션] 간단 소개 Profile 페이지 만들기



1. 앞에서 실습한 Notion 기능을 이용하여 나 자신을 간단히 소개하는 Profile 페이지를 만들어 보세요.
2. Notion의 템플릿 > 개인 > 작업 목록을 참고하여 1~2주 동안의 자신의 작업 목록 페이지를 만들어 보세요.

※ 자기 소개 페이지 대신
학습한 내용에 대한 정리 페이지를 작성해도 좋습니다.

04

협업 도구 - Slack

- 01. Slack 소개
- 02. Slack 특징
- 03. [실습] Slack 사용해보기

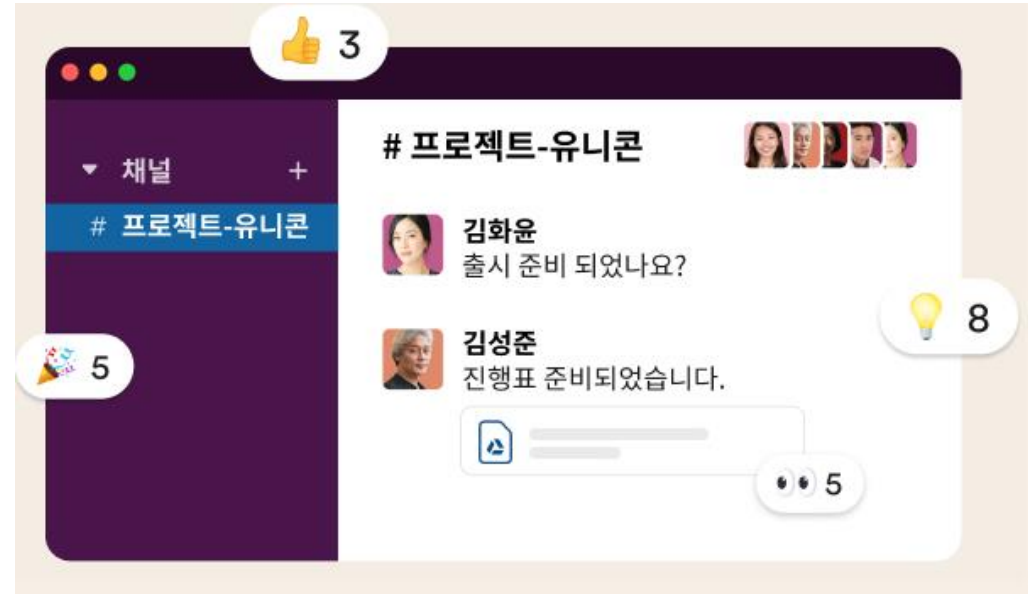
Slack 소개



- Slack이란?

<https://slack.com/intl/ko-kr/>

- 기업용 메시징 앱.
사내 커뮤니케이션 도구,
프로젝트 팀을 위한 메신저 등



- Slack 공식 소개 동영상 : <https://youtu.be/yHaLOoXA0vU>
- Slack help center : <https://slack.com/intl/ko-kr/help/categories/3600000049043>
- 참고 동영상 : <https://www.youtube.com/watch?v=j28tVbO5qZ8>

Slack 특징

■ Slack 특징

**사용하기 쉬운
프로젝트 메시지 도구**

슬랙 인터페이스 구성:

- 왼쪽 사이드바:**
 - workspace 이름: RAPA_컴퓨터비전...
 - Slack 찾아보기
 - 채널 목록:
 - # 1_파이썬과정 (선택됨)
 - # 2_리눅스과정
 - # 3_개발보조도구스터디
 - # 4_컴퓨터비전_opencv
 - # 일반
 - + 채널 추가
 - 다이렉트 메시지 목록:
 - 조윤실 나
 - yswoola99
 - + 팀원 추가
- 상단 바:** RAPA_컴퓨터비전과정 검색, 검색 아이콘, 도움말 아이콘, 사용자 프로필 (윤실)
- 채널 헤더:** # 1_파이썬과정, 채널 정보 (윤실 1)
- 채널 본문:**
 - + 채널피 추가
 - 메시지 #1: #1_파이썬과정 채널의 맨 첫 부분입니다. 오늘에 이 채널을 생성했습니다. RAPA 파이썬과정을 위한 채널입니다. [설명 편집](#)
 - + 사용자 추가
 - 오늘
 - 메시지 2: 조윤실 오후 11:05 #1_파이썬과정에 참여했습니다.
 - 메시지 3: 조윤실 오후 11:05 채널 설명 설정: RAPA 파이썬과정을 위한 채널입니다.
 - 나에게만 표시
 - 메시지 4: Slackbot 오후 11:06 이 채널은 "RAPA_컴퓨터비전과정"에 참여한 모든 새 멤버를 위해 "기본 채널"로 생성되었습니다. 기본 채널에서 이 채널을 제거하려면 언제든 [workspace admin settings](#)에서 제거할 수 있습니다.
 - 메시지 5: 조윤실 오후 11:22 채널 이름을 "파이썬과정"에서 "1_파이썬과정"(으)로 변경했습니다.
- 메시지 입력 영역:** #1_파이썬과정에게 메시지 보내기, 텍스트 형식 도구 (B, I, U, 등), 링크, 이미지, 파일 업로드, 전송 버튼

Slack 특징...

■ Slack 요금

더욱 생산적인 팀워크 구축하기

<p>무료</p> <p>Slack을 사용해 보는 가장 빠르고 쉬운 방법</p> <p>US\$0/월</p> <p>시작하기</p> <ul style="list-style-type: none">✓ 팀의 최근 메시지 10,000개 액세스✓ Google Drive, Office 365 등과 같은 다른 앱을 사용한 통합 10회✓ 팀원 간 1:1 음성 및 비디오 통화 <p>자세히 알아보기</p>	<p>Pro</p> <p>협업을 개선하고 싶은 소규모 팀에서 다양한 기능을 활용 가능</p> <p>US\$6.67/월*</p> <p>시작하기</p> <p>무료 플랜의 모든 장점 및 다음 기능</p> <ul style="list-style-type: none">✓ 조직 메시지 내역의 전체 상황 정보를 간편하게 확인✓ 무제한 통합을 이용해 한곳에서 정보 및 작업을 적시에 확인✓ 최대 15명의 팀원이 참여할 수 있는 그룹 음성 및 비디오 통화로 면대면 커뮤니케이션 지원✓ Slack 내에서 외부 조직 또는 게스트와 안전하게 협업 <p>자세히 알아보기</p>	<p>Business+</p> <p>비즈니스를 확장하고, 생산성을 향상시키고, 팀이 연결된 상태를 유지하도록 지원</p> <p>US\$12.50/월*</p> <p>시작하기</p> <p>Pro 플랜의 모든 장점 및 다음 기능</p> <ul style="list-style-type: none">✓ SAML 기반 SSO, 그리고 OneLogin, Okta 및 Ping Identity를 통한 Active directory 실시간 동기화로 고급 ID 관리 실현✓ 모든 메시지에 대해 데이터 내보내기를 지원하여 규정 준수 요구사항 충족✓ 가동 시간 99.99% 보장과 4시간 이내에 대응하는 연중무휴 지원으로 24시간 팀워크 및 지원 제공 <p>자세히 알아보기</p>	<p>Enterprise Grid</p> <p>요구 사항이 다양한 엔터프라이즈를 위한 Slack — 규제 사항을 준수하기 위한 유연성과 성장에 걸맞은 기능 제공</p> <p>영업팀 문의</p> <p>Business+의 모든 장점 및 다음 기능</p> <ul style="list-style-type: none">✓ HIPAA 지원 및 엔터프라이즈 키 관리 애드온과 같은 엔터프라이즈급 보안 및 규정 준수로 안심하고 사용 가능✓ 최대 500,000명의 사용자 지원으로 대규모 협업 및 조정 가능✓ 중앙 집중식 제어와 사용자 지정 가능한 정책으로 관리 간소화✓ 전담 어카운트 및 고객 성공팀을 통한 맞춤형 지원 제공 <p>자세히 알아보기</p>
--	--	--	---

[실습] Slack 사용해보기



1. 회원 가입하기 (구글계정)

<https://slack.com/intl/ko-kr/>

<https://www.youtube.com/watch?v=j28tVbO5qZ8>

2. Slack 사용하기

- ① 워크스페이스 만들기
- ② 채널 만들기
- ③ 채널에 사용자 추가하기
- ④ 채널에 메시지 작성하기
- ⑤ 채널에 메시지(파일첨부) 작성하기

05

서버/서비스 관리 도구 Docker

01. Docker란?

02. [실습] Docker 사용해보기

Docker

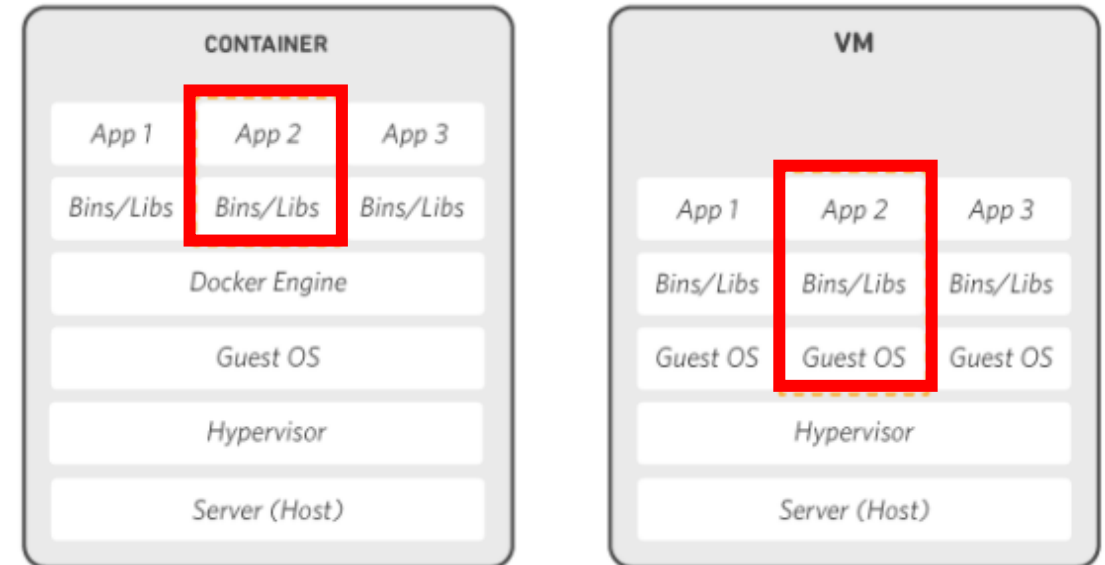
<https://www.docker.com/>



■ Docker란?

- Docker는 애플리케이션을 신속하게 구축, 테스트 및 배포할 수 있는 컨테이너 기반의 오픈소스 가상화 플랫폼
- Docker는 소프트웨어를 컨테이너라는 표준화된 유닛으로 패키징 함
- 컨테이너에는 라이브러리, 시스템 도구, 코드, 런타임 등 소프트웨어를 실행하는 데 필요한 모든 것이 포함되어 있음
- 환경에 구애 받지 않고 애플리케이션을 신속하게 배포 및 확장할 수 있음
- 참고 동영상:

https://www.youtube.com/watch?v=gQG_FX9hIMg



[실습] Docker 사용해보기




- Docker 설치하기 (for Windows 10 Home)

- 참고 동영상: <https://www.youtube.com/watch?v=J9pxPVcd-fY>

1. Docker 다운로드 <https://www.docker.com/products/docker-desktop>

2. Docker 설치

- ①  Docker Desktop Installer.exe → 다운로드된 파일 실행 & 설치
- ② 참고 동영상을 참고해서 진행하기

THE END
