

Chapter 02. 실전 응용편

개발환경 구축하기

개발환경?

나의 개발환경 구축하기

- 개발 언어
 - C
 - Python
 - Java
- 개발 환경 설정하기
 - 버전 선택/확인을 위한 각종 유틸리티
 - alternates, which, whereis
 - anaconda
 - docker
- 개발 도구 설정하기 (IDE)
 - jupyter notebook

개발언어 – C 개발 환경 (컴파일러 등 설치하기)

C언어 개발환경 구축하기

- C언어 및 관련 개발환경 구축하기
 - `sudo apt install build-essential`
 - `sudo apt install binutils`
- 확인하기
 - `gcc`
 - `g++`
 - `make`

```
user1@user1-VirtualBox:~$ apt show build-essential
Package: build-essential
Version: 12.4ubuntu1
Priority: optional
Build-Essential: yes
Section: devel
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Matthias Klose <doko@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 20.5 kB
Depends: libc6-dev | libc-dev, gcc (>= 4:7.2), g++ (>= 4:7.2), make, dpkg-dev (>= 1.17.11)
Supported: 5y
Download-Size: 4,758 B
APT-Manual-Installed: yes
APT-Sources: http://kr.archive.ubuntu.com/ubuntu bionic/main amd64 Packages
Description: build-essential 패키지의 정보 목록
데비안 패키지를 만들 계획이 없다면, 이 패키지는 필요치 않습니다. dpkg (>= 1.14.18)로 시작하는 이 패키지는 데비안 패키지 제작을 위해 필요합니다.
```

```
user1@user1-VirtualBox:~$ gcc --version
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

user1@user1-VirtualBox:~$ g++ --version
g++ (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

user1@user1-VirtualBox:~$ make --version
GNU Make 4.1
x86_64-pc-linux-gnu 빌드
Copyright (C) 1988-2014 Free Software Foundation, Inc.
라이선스 GPLv3+: GNU GPL 버전 3 또는 이후 <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
user1@user1-VirtualBox:~$
```

```
#include <stdio.h>
int main() {
    printf("hello, world\n");
}
```

```
$ gcc -o hello hello.c
```

개발언어 – Python 개발 환경 (Python 패키지 설치하기)

Python 개발 환경 구축하기

- Python 언어 및 관련 개발환경 구축하기
 - `sudo apt install python`
 - `sudo apt install python3`
- 확인하기
 - `python --version`
 - `python3 --version`
- Ubuntu 16.04 기본값
 - Python 2.7.12
 - Python3.5.2
- Ubuntu 18.04 기본값
 - Python 2.7.17
 - Python 3.6.9

```
user1@user1-VirtualBox:~$ python --version
Python 2.7.12
user1@user1-VirtualBox:~$ python3 --version
Python 3.5.2
```

```
user1@user1-VirtualBox:~$ sudo apt install python
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
패키지 python는 이미 최신 버전입니다 (2.7.15~rc1-1).
0개 업그레이드, 0개 새로 설치, 0개 제거 및 11개 업그레이드 안 함.
user1@user1-VirtualBox:~$ sudo apt install python3
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
패키지 python3는 이미 최신 버전입니다 (3.6.7-1~18.04).
0개 업그레이드, 0개 새로 설치, 0개 제거 및 11개 업그레이드 안 함.
user1@user1-VirtualBox:~$ python --version
Python 2.7.17
user1@user1-VirtualBox:~$ python3 --version
Python 3.6.9
```

개발언어 – Python 개발 환경 (Python 패키지 설치하기)

Anaconda를 이용한 개발 환경 구축하기

- 공식 도메인
 - <https://www.anaconda.com/products/individual>
- 다운로드
 - `curl -O https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh`
- 설치 스크립트 실행
 - `bash Anaconda3-2020.02-Linux-x86_64.sh`
또는
`bash Anaconda3-2020.02-Linux-x86_64.sh -b -p /data/anaconda`
- PATH 설정 및 재로딩(바로적용)
 - `/data/anaconda/bin/conda init`
 - `source ~/.bashrc`
 - `conda --version`
- 개발환경 만들기
 - `conda create --name my_env35 python=3.5`
 - `conda activate my_env35`



repo.anaconda.com/archive/

Anaconda installer archive

Filename	Size	Last Modified	MD5
Anaconda3-2020.02-Linux-ppc64le.sh	276.0M	2020-03-11 10:32:32	fef889d3939132d9caf7f56ac9174ff6
Anaconda3-2020.02-Linux-x86_64.sh	521.6M	2020-03-11 10:32:37	17600d1f12b2b047b62763221f29f2bc
Anaconda3-2020.02-MacOSX-x86_64.pkg	442.2M	2020-03-11 10:32:57	d1e7fe5d52e5b3c0cb38d9af262688e89
Anaconda3-2020.02-MacOSX-x86_64.sh	430.1M	2020-03-11 10:32:34	f022959e0bd45dee0c14b20e58ad316
Anaconda3-2020.02-Windows-x86.exe	423.2M	2020-03-11 10:32:58	64ae8d0e5095b9a878d4522db4ce751e
Anaconda3-2020.02-Windows-x86_64.exe	466.3M	2020-03-11 10:32:35	6b02c1c91049d29fc65be68f2443079a

```
user1@user1-VirtualBox:~$ bash Anaconda3-2019.03-Linux-x86_64.sh
Welcome to Anaconda3 2019.03

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
=====
Anaconda End User License Agreement
=====

Copyright 2015, Anaconda, Inc.

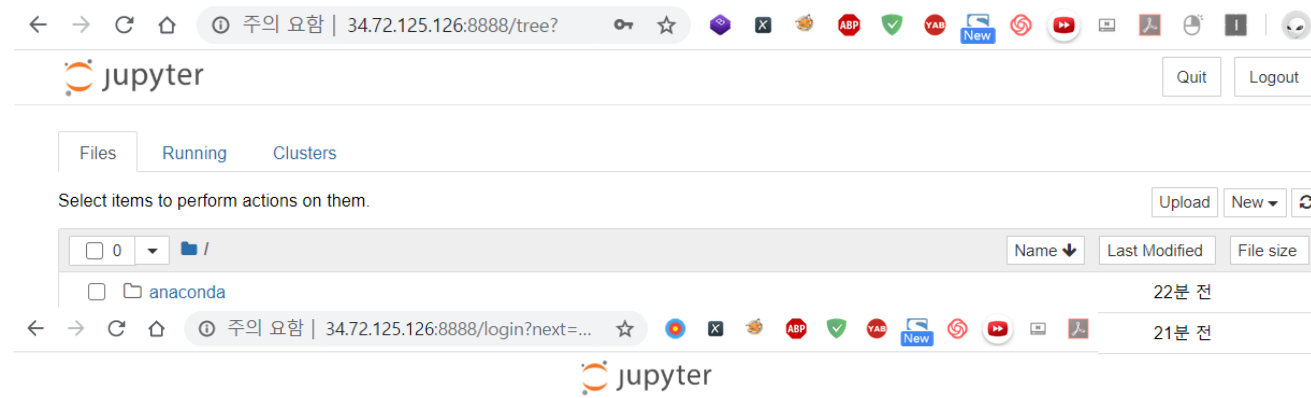
All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
```

개발언어 – Python 개발 환경 (Jupyter 노트북 설치하기)

Jupyter Notebook 을 이용한, 웹 기반의 IDE 로 개발 환경 구축하기

- 주피터 노트북 데몬 실행
 - `jupyter notebook --ip=0.0.0.0 --no-browser`
- 주피터 노트북 암호 생성
 - `jupyter notebook password`



- 주피터 노트북 설정 파일 만들기
 - `jupyter notebook --generate-config`
 - `jupyter notebook --config=~/.jupyter/jupyter_notebook_config.py --no-browser`
- 터미널이 닫혀도 백그라운드에서 실행되는 프로세스로 실행하기
 - `nohup jupyter notebook &`

개발언어 – Python 개발 환경 (서비스를 통한 자동 시작)

Jupyter Notebook 을 부팅 서비스로 만들어 자동 실행하기

- 서비스 만들기
 - `sudo vi /lib/systemd/system/jupyter.service`

```
[Unit]
Description=Jupyter Notebook Server
```

```
[Service]
Type=simple
PIDFile=/run/jupyter.pid
ExecStart=/data/anaconda/bin/jupyter-notebook
                --config=/home/<username>/.jupyter/jupyter_notebook_config.py
WorkingDirectory=/data/anaconda
User=<username>
Group=<username>
Restart=always
RestartSec=10
```

```
[Install]
WantedBy=multi-user.target
```

- 설정 변경 (~/.jupyter/jupyter_notebook_config.py)
 - `c.NotebookApp.ip = '*'`
 - `c.NotebookApp.notebook_dir = '/data/workspace/'`
 - `c.NotebookApp.open_browser = False`

- 서비스 등록
 - `sudo systemctl daemon-reload`
 - `sudo systemctl enable jupyter`
 - `sudo systemctl start jupyter`
 - `systemctl status jupyter`

개발언어 – Java 개발 환경 (Java 패키지 설치하기(JDK/JRE))

Java (OpenJDK) 설치하기

- 배포판 기본 버전 java 설치하기 (openjdk11)
 - Runtime 설치하기
 - `sudo apt install default-jre`
 - SDK 설치하기
 - `sudo apt install default-jdk`
- 내가 원하는 버전 설치하기 (openjdk12)
 - ??? (update-alternative 파트로...)

개발도구 버전관리 유틸리티 – which, whereis

실행되는 바이너리 위치 확인

- which gcc
- which python
- which java

- whereis gcc
- whereis python
- whereis java

개발도구 버전관리 유틸리티 – update-alternatives

다양한 버전의 유틸리티 관리 (java, python 등)

- 다중 Python 버전 관리 (이것보다 virtualenv 가 더 좋은 방법)
 - `update-alternatives --list python`
 - `update-alternatives --install /usr/bin/python python /usr/bin/python2.7 10`
 - `update-alternatives --install /usr/bin/python python /usr/bin/python3.4 1`
 - `update-alternatives --display python`
 - `update-alternatives --remove python /usr/bin/python2.7`
 - `update-alternatives --config python`
- 다중 Java 버전 관리??
 - `update-alternatives --list java`
 - `ls -al /usr/bin/java*`
 - `ls -al /etc/alternatives/java*`

개발도구 버전관리 유틸리티 – update-alternatives

다양한 버전의 java 관리하기

- JAVA 버전 찾아 다운로드
 - <http://jdk.java.net/archive/>
 - `curl -O https://download.java.net/java/GA/jdk12.0.2/e482c34c86bd4bf8b56c0b35558996b9/10/GPL/openjdk-12.0.2_linux-x64_bin.tar.gz`
- 현재 설치된 자바 확인
 - `ls -al /usr/bin/java*`
- 새로 자바 설치(압축풀기)
 - `sudo mkdir -p /usr/java`
 - `tar xvfz openjdk-12.0.2_linux-x86_bin.tar.gz /usr/java`
- 신규 버전 등록
 - `sudo update-alternatives --install "/usr/bin/java" "java" "/usr/java/jdk-12-0.2/bin/java" 1`
 - `sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/java/jdk-12-0.2/bin/javac" 1`
 - `sudo update-alternatives --config java`

개발환경 – docker


컨테이너 관리도구 docker 설치

- 우분투 공식 사이트에서 유지보수 되는, 패키지 관리자를 통한 도커 설치
 - `sudo apt install docker.io`
- 공식 사이트로부터 리포지토리를 추가 한 후 설치
 - `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"`
 - `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
 - `sudo apt-key fingerprint 0EBFCD88`
 - `sudo apt update`
 - `sudo apt install docker-ce docker-ce-cli`
- 공식 사이트로부터 자동 설치 스크립트를 다운로드 받아 설치
 - `curl -fsSL https://get.docker.com/ | sudo sh`

-f : fail-silently, -s : silent, -S : show-error, -L : location

개발환경 – docker

docker 의 설정 다루기 (이미지/컨테이너 디렉토리 관리/변경)

- sudo 없이 사용하기
 - `sudo usermod -aG docker $USER`
- 도커 이미지/컨테이너 디렉토리 변경
 - 현재 동작중인 도커 서비스 확인 및 중지
 - `sudo lsof | grep /var/lib/docker`
 - `sudo systemctl stop docker`
 - 디렉토리 변경 설정파일 추가 
 - `sudo vi /lib/systemd/system/docker.service`
 - `ExecStart=... --data-root=/data/docker_dir`
 - 기존 도커 디렉토리 용량 확인
 - `sudo du -sh /var/lib/docker`
 - 도커 재시작
 - `sudo systemctl start docker`
 - 변경된 디렉토리 용량 확인
 - `sudo du -sh /data/docker_dir`

참고: SysV init, Upstart 에서는...

- `vi /etc/default/docker`
 - `DOCKER_OPTS="-g /data/docker_dir"`
- Systemd 에서는 좌측 설정 참고.