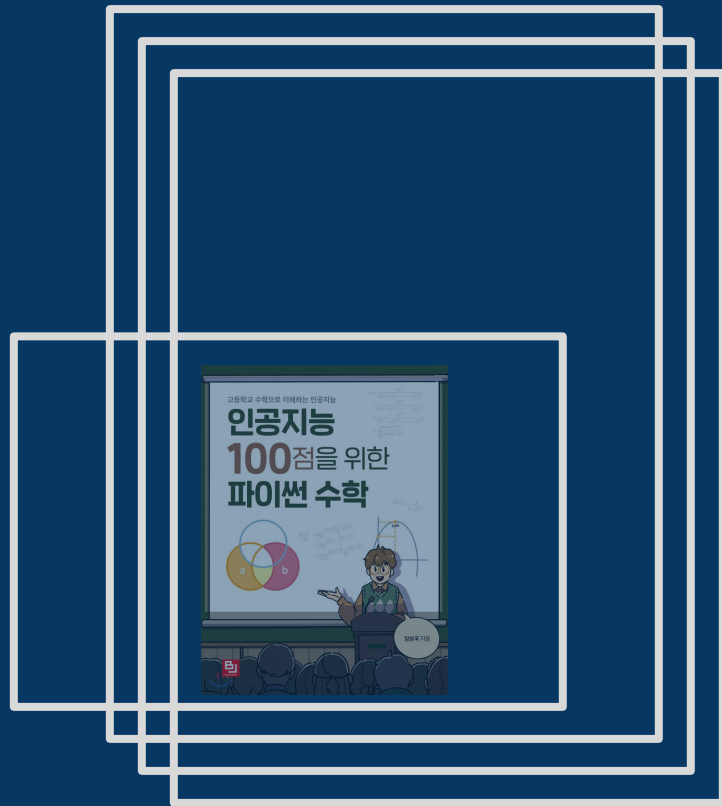


# 00. 개발 환경 설정

인공지능 100점을 위한 파이썬 수학



# Contents

1. 파이썬 설치
2. 가상개발환경
3. 패키지 설치
4. 주피터 노트북
5. CoLab
6. Visual Studio Code

# Contents

7. 깃허브 계정 생성

8. 노션 (notion)

9. 슬랙 (slack)

10. 개인 블로그 (깃허브 블로그)

# 1. python 설치

# 01. 파이썬 설치

## ● 버전

3.7, 윈도우, 64비트(x86-64)

# 01. 파이썬 설치

## ○ 위치

**python.org**

**C:\python\python37**

# 01. 파이썬 설치

## ○ 위치

파일탐색기 - '상단 디렉토리 위치창' 에서

**cmd**

(명령 프롬프트)실행

## 2. 가상개발환경



## 02. 가상개발환경

### ● 가상개발환경

**pip install virtualenv**

**pip list**

## 02. 가상개발환경

### ● 가상개발환경

[만들기]

이동 (c:\dev\ML01)

```
virtualenv -p c:\python\python37\python.exe venv
```

```
virtualenv venv
```

## 02. 가상개발환경

### ● 가상개발환경

[실행]

```
.\venv\Script\activate
```

[종료]

```
deactivate
```

# 3. 패키지 설치

## 03. 패키지 설치

### ○ 패키지

**scipy, matplotlib, tensorflow**

## 03. 패키지 설치

### 🔵 pip 로 설치

## 가상개발환경을 만든 뒤 설치

CODE

```
(venv) c:\dev\zb_ml>pip list  
(venv) c:\dev\zb_ml>pip install scipy  
...
```

## 4. 주피터 노트북

## 04. 주피터 노트북

### ○ 실행

**[설치] pip install jupyter**

**[실행] jupyter notebook**



# 5. CoLab

## 05. Google Colaboratory

### ● 실행

구글에서 무료로 제공하는  
머신러닝을 위한  
온라인 주피터 노트북  
(GPU 사용가능)

# 6. Visual Studio Code

# 06. Visual Studio Code

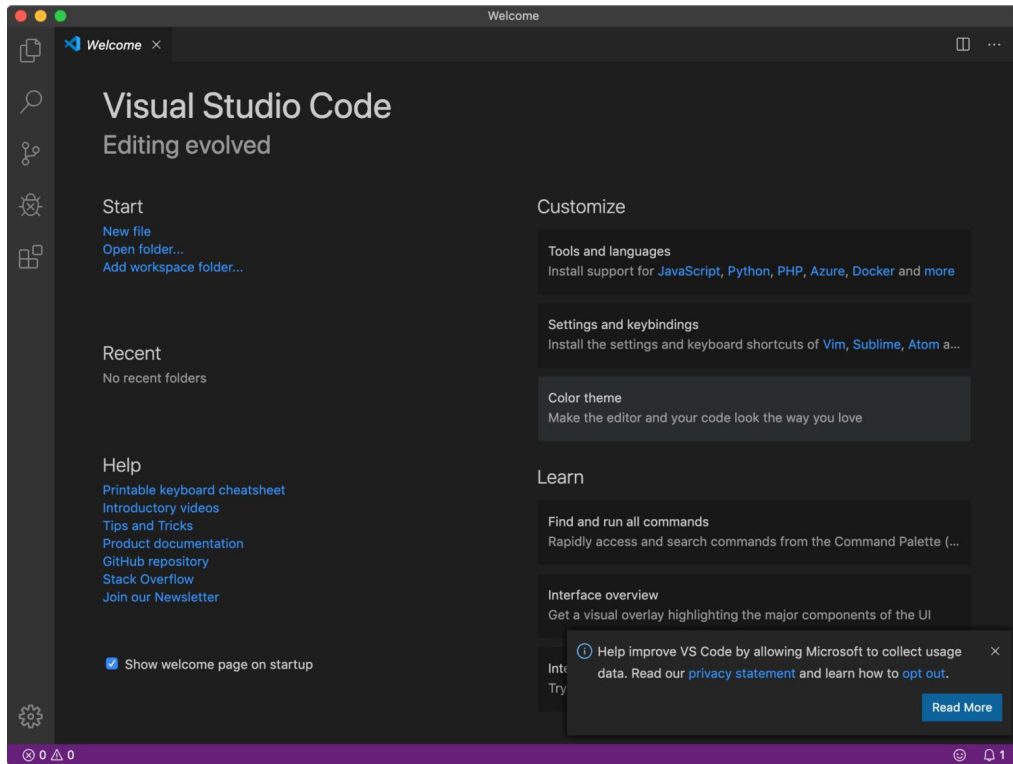
## 🔵 실행

비주얼 스튜디오 코드(Visual Studio Code)는  
마이크로소프트Microsoft에서 오픈소스로  
개발하고 있는 소스 코드 에디터

맥OSmacOS, 리눅스Linux, 윈도우Windows 등  
대부분 운영체제를 모두 지원

마이크로소프트의 통합 개발 환경(IDE) 비주얼  
스튜디오(Visual Studio)와 이름이 비슷하지만  
오픈 소스로 별도 개발

IDE보다는 코드 에디터에 해당, MIT 라이선스로  
오픈소스로 공개되어 있어, 무료 사용



## 06. Visual Studio Code

### ● 실행



# 7. GIT & GITHUB

## 07. GIT & GITHUB

### ○ 실행

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

<https://github.com/>

# 07. GIT & GITHUB

## Git 기본명령어

누구나 쉽게 이해할 수 있는 Git 입문

- <https://backlog.com/git-tutorial/kr/>

초심자를 위한 Github 협업

- <https://milooy.wordpress.com/2017/06/21/working-together-with-github-tutorial/>

Git cheat sheet

- <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>



## 07. GIT & GITHUB

### 🕒 Git 단계별

1. 새 프로젝트 생성 후 프로젝트 폴더에 **git** 설정
2. 개발진행 & **commit**
3. 초기버전 완성
4. 새 기능 추가 팀에겐 **feature branch** 를, 버그 수정팀에겐 **bugfix branch** 를 주어 개발 진행
5. 각 **branch** 개발이 끝나면 **master branch** 로 **merge** (통합)

# 07. GIT & GITHUB

## 🕒 git 기초 명령

<<git-scm.com 다운, 설치>>

```
git init
```

```
git config --global user.name "myName"
```

```
git config --global user.email "myEmail@email.com"
```

```
git status
```

```
git add --all .
```

```
git commit -m "설명"
```

<<github 계정 생성>>

github.com 에서 repository 생성 (myrepository)

<<github 저장>>

```
git remote add origin https://github.com/mygitid/myreposit.git
```

```
git push -u origin master
```

## 07. GIT & GITHUB

### 🕒 실습1

git 을 설치한다.

gitHub 에 계정을 등록한다.

# 07. GIT & GITHUB

## 🕒 실습2

### 최초 실행시 깃/깃허브 설정

```
$ git config --global user.name "eventia"  
$ git config --global user.email "myEmail@gmail.com"
```

## 07. GIT & GITHUB

### 🕒 실습3

GitHub 에서 new repository 실행

## 07. GIT & GITHUB

### 🕒 실습4

- ✓ PC 에 임시디렉토리 trygit 을 만든다.
- ✓ 마우스 우클릭으로 “Git Bash Here” 을 실행
- ✓ 그 위치에 편집기를 열어 텍스트 파일 myword.txt 를 만든다.
- ✓ myword.txt 를 편집기로 열어 적당한 가사를 넣는다.

## 07. GIT & GITHUB

### 🕒 실습5

- ✓ command(명령 프롬프트) 창에서

```
$ git init
```

- ✓ 추가할 파일을 git add 명령으로 추가한다.

```
$ git add myword.txt
```

- ✓ 추가가 잘 되었다면 커밋한다.

```
$ git commit -m "Add MyFirst Commit"
```

## 07. GIT & GITHUB

### 🕒 실습6

- ✓ 로컬저장소(PC의 디렉토리/폴더)와 깃허브의 repo 를 연결한다.

\$ git remote add origin 깃허브-레포지토리이름.git

- ✓ 확인을 위해 다음을 입력한다.

\$ git remote -v

- ✓ GitHub 로 변경사항을 업로드한다.

\$ git push -u origin master



## 07. GIT & GITHUB

### 🕒 협업을 위한 설정 - 단일 저장소

- ✓ GitHub 에서
  - + 아이콘을 클릭, New Organization 을 만든다.
- ✓ repo 를 생성한다.
- ✓ Collaborator 을 초대한다.
  - 초대받은 사람은 알림이 뜨고, 초대를 수락하면 권한이 주어진다.
  - `git clone [github주소.git]` 입력

## 07. GIT & GITHUB

### 🕒 협업을 위한 설정 - 단일 저장소

- ✓ 만든 이를 M 이라하고 초대받은 이를 S 라 하자. M 과 S는 브랜치를 만든다.
  - `git branch dev`
  - `git checkout dev`
  - `git push -u origin dev`
- ✓ S 가 `git pull` 을 하고, dev 로 브랜치를 이동한다.
  - `git pull`
  - `git checkout dev`
- ✓ branch 를 지울때는 `git branch -d <브랜치이름>`

## 07. GIT & GITHUB

### 🕒 협업을 위한 설정 - 단일 저장소

- ✓ M 은 feature/main 브랜치로, S 는 feature/form 브랜치로 개발
- ✓ (M 또는 S가) Pull Request 로 리뷰를 받고 합치기 요청
- ✓ (상대편 S 또는 M이) Merge pull Request 로 합치기 실행
- ✓ Merge 에서 Master 로
- ✓ (필요하면) releases 로 압축파일로 만듦



## 07. GIT & GITHUB

### 🕒 git 명령 순서 - git 처음 설치한 다음 1번만 실행

<<git 최초 설치 후>>   파일탐색기로 내 프로젝트가 있는 디렉토리로 간 후 마우스 우측버튼 클릭,  
"Git Bash Here" 선택

```
git config --global user.name "eventia"  
git config --global user.email "myEmail@gmail.com"
```

# 07. GIT & GITHUB

## 🕒 git 명령 순서 - 프로젝트를 처음 시작할 때

프로젝트 디렉토리에서 마우스 우클릭 “Git Bash Here”

```
git init
```

```
git status
```

```
git add --all .
```

```
git commit -m "설명"
```

## 07. GIT & GITHUB

### 🕒 git 명령 순서 - 프로젝트를 처음 시작할 때

Github 에 올릴때

github.com 에서 repository 생성 (myrepository)

<<github 저장 - 최초>>

```
git remote add origin https://github.com/mygitid/myreposit.git  
git push -u origin master
```

# 07. GIT & GITHUB

## 🕒 git 명령 순서 - 프로젝트를 처음 시작할 때

Github 에 올릴때

<<github 저장 - 매번>>  
**git push**





# 07. GIT & GITHUB

## 🕒 git 명령 순서

<<Project ML02>>

```
git init
```

<<매번 - 수정할 때마다>>

```
git add --all .
```

```
git commit -m "설명"
```

<<github 계정 생성>>

github.com 에서 repository 생성 (myrepository)

<<github 저장 - 최초>>

```
git remote add origin https://github.com/mygitid/myreposit.git
```

```
git push -u origin master
```

<<github 저장 - 매번>>

```
git push
```

# 8. NOTION

## 08. NOTION

### ● 실행

학생이라면 노션 계정을 만들자

# 08. NOTION

## ○ 실행

### For individuals



#### Personal

For organizing every corner of your life.

**\$0**

Get started

Free for individuals

- ✓ Unlimited pages & blocks
- ✓ Share with 5 guests
- ✓ Sync across devices

Pay annually ☐ Pay monthly



#### Personal Pro

For power users who want to do even more.

**\$4** per month

Try it free

Everything in Personal, plus

- ✓ Unlimited file uploads
- ✓ Unlimited guests
- ✓ Version history
- ✓ API access COMING SOON

### For teams & businesses



#### Team

For teams to write, plan, and work together in one spot.

**\$8** per member per month

Try it free

Everything in Pro, plus

- ✓ Unlimited team members
- ✓ Collaborative workspace
- ✓ Advanced permissions
- ✓ Admin tools



#### Enterprise

Controls and support to run your company.

 [Contact sales →](#)

Try it free

Everything in Team, plus

- ✓ SAML SSO
- ✓ Advanced security
- ✓ Dedicated manager
- ✓ Custom contract

# 9. SLACK

## 09. SLACK

### ● 실행

개발자들이 자주 사용하는 협업 툴

**BS\_ML**

**bsmlhq.slack.com**

# 10. BLOG



## 10. 개인 블로그 만들기

### ● 실행

원하는 곳

(네이버, 티스토리, **github**)