

# Hello, Python!

멋쟁이사자처럼 9기 교육자료

멋사 8기 남민정



# CONTENTS

---

## Python 기초

: 기초적인 문법을 알아봅시다!

## • pip 가상환경이란?

: 가상환경을 왜 만들어야 되는거지?

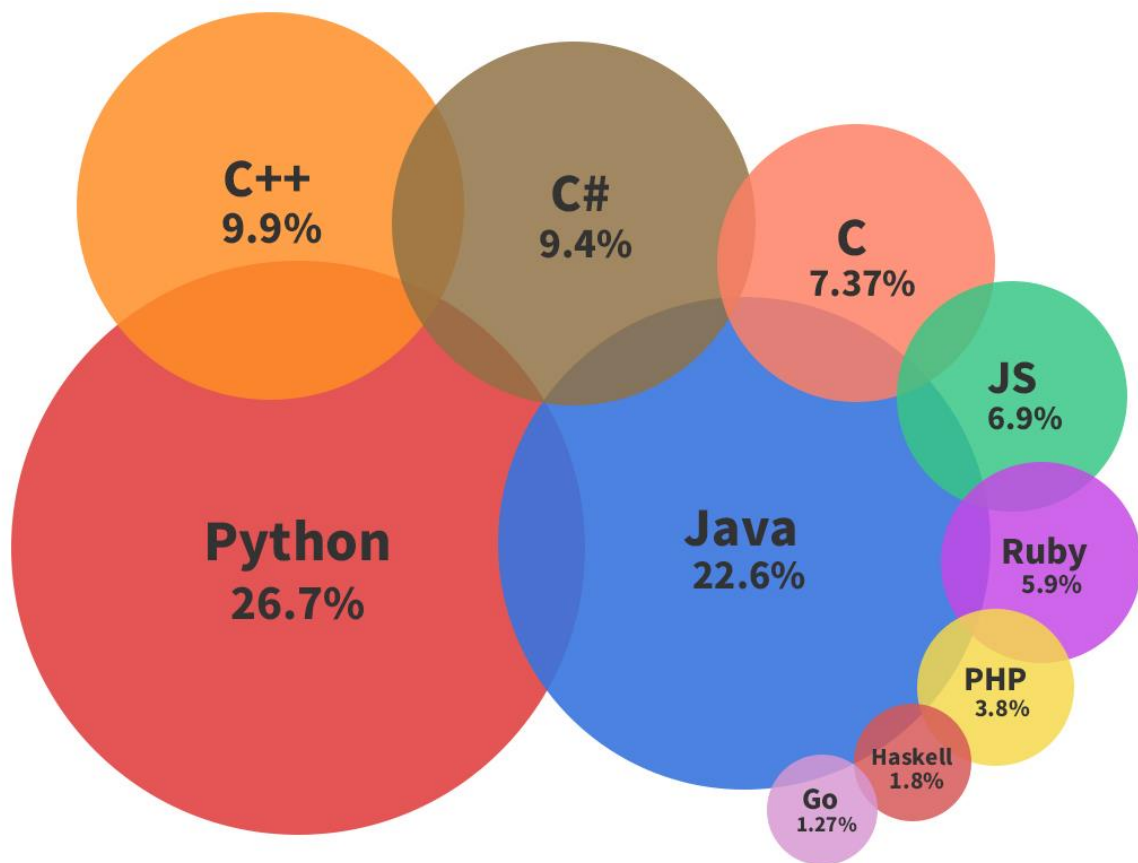
## • Django 개발환경 만들기!

: Django를 위한 준비 끝!

# Python 기초

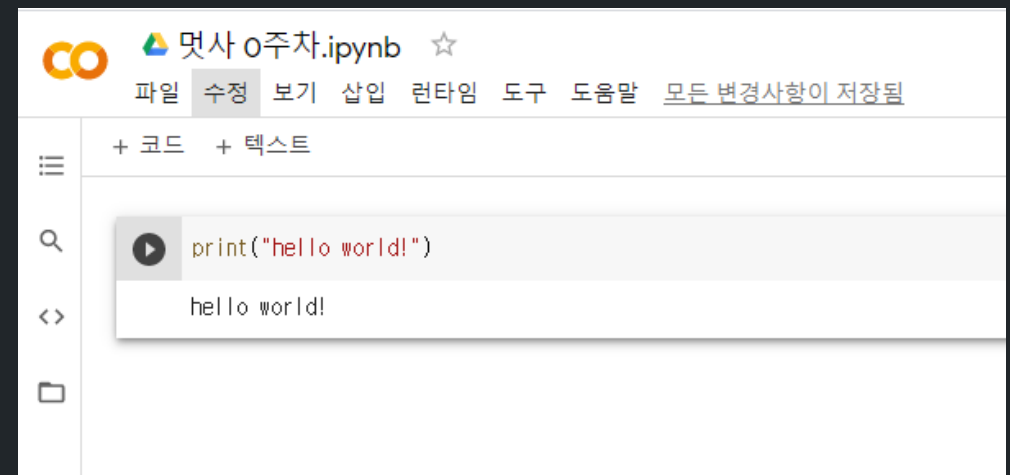
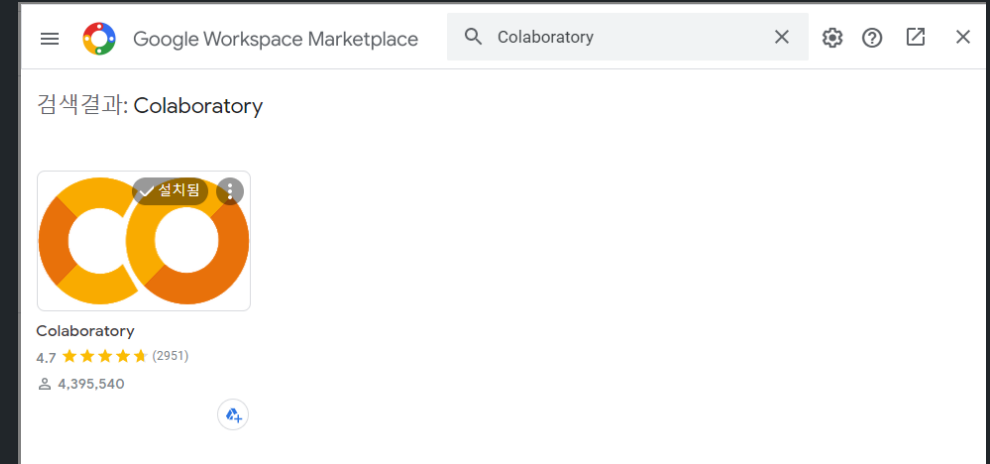
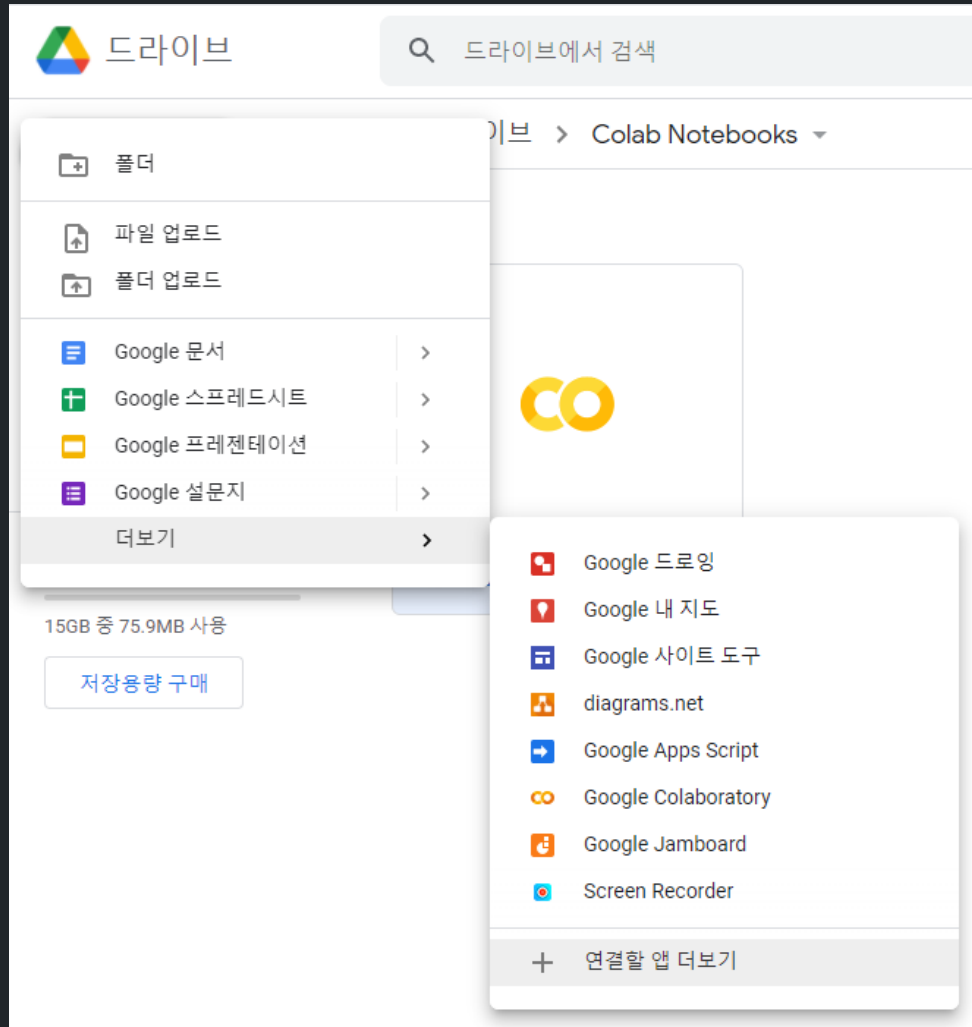
: 기초적인 문법을 알아보시다!

2016년 가장 인기 있는 프로그래밍 언어



# GitHub이란?

왜 모든 개발자들은 깃헙을 사용하나요?



실습 코드는 colab에 전부 올라와있습니다! > [링크](#)

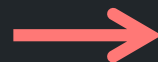
# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 01. 입출력

- ✓ Print() : 화면에 출력을 해주는 출력함수.
- ✓ Input(): 화면에 입력을 해주는 입력함수.

```
first.py x
first.py
1 print(1+1)
2 print("멋쟁이 사자처럼")
```



```
$ python first.py
2
멋쟁이 사자처럼
```

```
first.py > ...
1 student=input("이름 : ")
2 print(student)
```

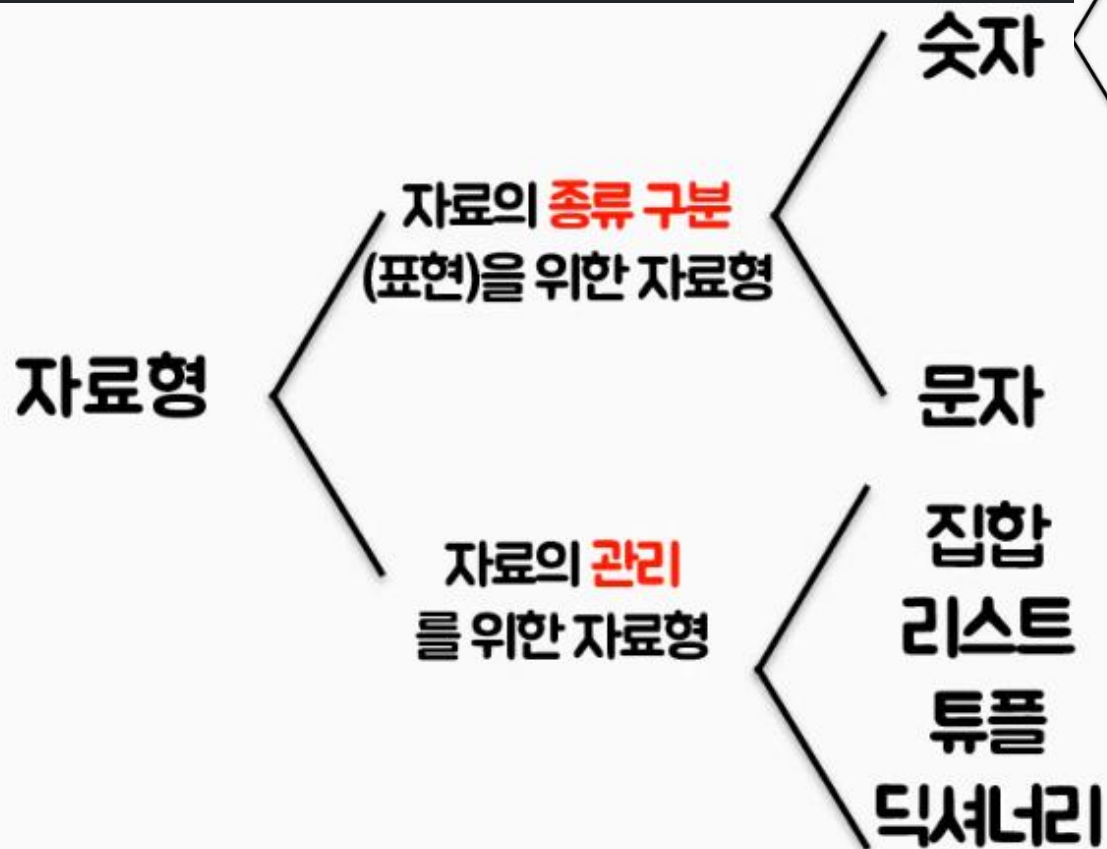


```
$ python first.py
이름 : 홍길동
홍길동
```

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 02. 자료형



→ int : 정수형

→ float : 실수형

```
$ python first.py
정수 : 10
실수 : 10
정수는 10 이고 실수는 10.0 입니다.
```

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 02. 자료형

### 리스트

변할 수도 있는 데이터들을 나란히 묶어주는 자료형

#### 리스트의 곱셈

`[1,2,3] * 3` : 리스트를 반복해서 출력

#### 리스트의 덧셈

`[1,2,3] + [4,5,6]` : 리스트를 합쳐서 출력

#### 리스트의 인덱싱

`list = [1,2,3], list[0]==1` : 원하는 인덱스 찾아서 출력

`list = [1,2,3] : list[0:2]==[1,2]`

`list[x;y]` : x번째 인덱스부터 y-1번째 인덱스까지 출력

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 02. 자료형

### 딕셔너리

대응대는 관계의 데이터를 나타내는 자료형,  
Key와 Value를 **한쌍으로** 갖는 자료형

```
dics = {'대한민국': '서울', '영국': '런던', '프랑스': '파리'}
```

### 딕셔너리 사용

dics['대한민국'] == '서울 '

### 내장함수

disc[key] = value → 하나의 key, value 쌍 추가  
disc.get(key) → key로 value 얻는 내장함수.



# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 02. 자료형

튜플

변할 수 없는 데이터들을 나란히 묶어주는 자료형

리스트 vs 튜플

리스트는 대괄호로 작성 / 튜플은 소괄호로 작성

`list[1] = 10` → `list = [1, 10, 3]` (변경 가능 자료형)

`tuple[1] = 10` → `tuple = error` (변경 불가 자료형)

튜플의 연산

리스트와 동일!

덧셈은 +로, 곱셈은 \*로 진행함

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 03. 제어문



무엇을 제어하는가?

코드의 흐름을 제어

WHY?

상황에 따라 다르게 컴퓨터에게 요구하기 위해



분기문 (=if문)

컴퓨터에게 선택의 여지와 조건 부여

If(조건):

↑ 조건이 참일 때 실행할 코드

★ 공백★ (tab1번 or 스페이스바 4번)

<예제>

Score가 85이상이면 PASS, 아니면 FAIL

```
score = int(input("점수를 입력해 주세요 : "))
```

- 범위를 조건으로

동아리가 멋사인가 아닌가에 따라 다른 결과 출력

```
activity = input("너 동아리 뭐해? : ")
```

- 일치 여부를 조건으로

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 03. 제어문



무엇을 제어하는가?

코드의 흐름을 제어

WHY?

상황에 따라 다르게 컴퓨터에게 요구하기 위해



분기문 (=if문)

컴퓨터에게 선택의 여지와 조건 부여

If(조건):

조건이 참일 때 실행할 코드

★ 공백★ (tab1번 or 스페이스바 4번)

<예제>

Score가 85이상이면 PASS, 아니면 FAIL

```
score = int(input("점수를 입력해 주세요 : "))

if(score>=85):
    print("PASS")
else:
    print("FAIL")
```

- 범위를 조건으로

동아리가 멋사인가 아닌가에 따라 다른 결과 출력

```
activity = input("너 동아리 뭐해? : ")

if(activity=="멋쟁이사자처럼"):
    print("어, 너도 멋사야?")
else:
    print("..그래..")
```

-일치 여부를 조건으로

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 03. 반복문



반복문

반복을 해주는 문법

반복제어변수

: 반복대상을 하나하나 갖고와 명령문을 실행

반복대상

: 반복할 대상 (range함수 많이 사용)

`range(x,y)` : x이상 y미만의 수 리스트로 변환

`range(x)` : 0부터 x미만의 수 리스트로 변환

Ex) 우리반 학생 성적 총합 구하기

: 우리반 학생 평균 성적을 반복해서 더하기

반복대상

반복행위

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 03. 반복문



반복문

반복을 해주는 문법

Ex) 우리반 학생 성적 총합 구하기

: 우리반 학생 평균 성적을 반복해서 더하기

반복대상

반복행위

for문

: 반복 대상에 반복 제어 변수가  
포함될 때까지 반복

for 반복제어변수 in 반복대상:  
반복 실행할 코드

```
sum=0
for number in [95,77,68,80,100]:
    sum += number
    print(sum)
```

While

: 조건이 참일 동안 반복

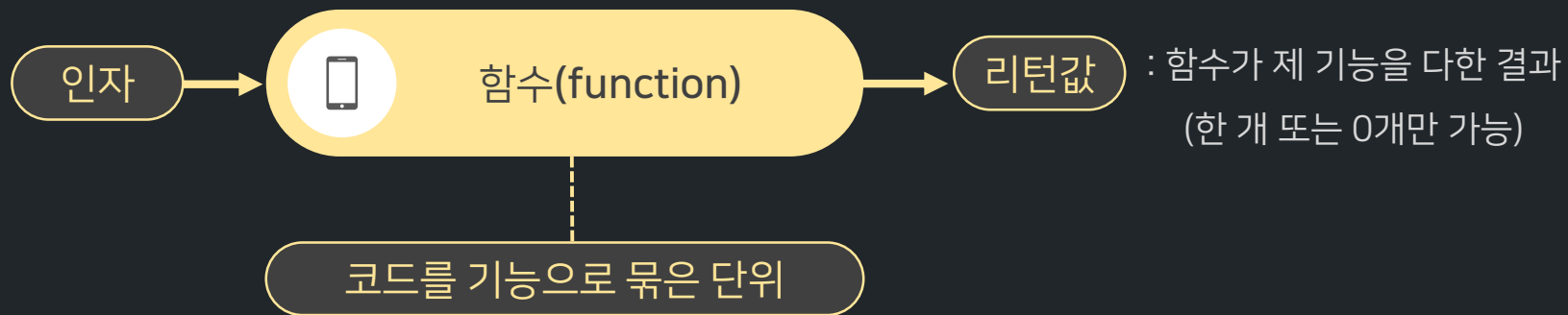
While(조건):  
조건이 참일 동안 수행할 명령문

```
num=10
while(num>0):
    print("반복문 수행중!")
    num--
```

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 04. 함수



WHY?

간단하게 줄이기 위해 + 오류를 찾기 쉽게

D.A.C

Divide & Conquer

복잡한 문제를 작게 쪼개서  
간단하게 만들고 정복한다

```
# 함수의 구성
def 함수이름(인자) :
    실행코드
    리턴값

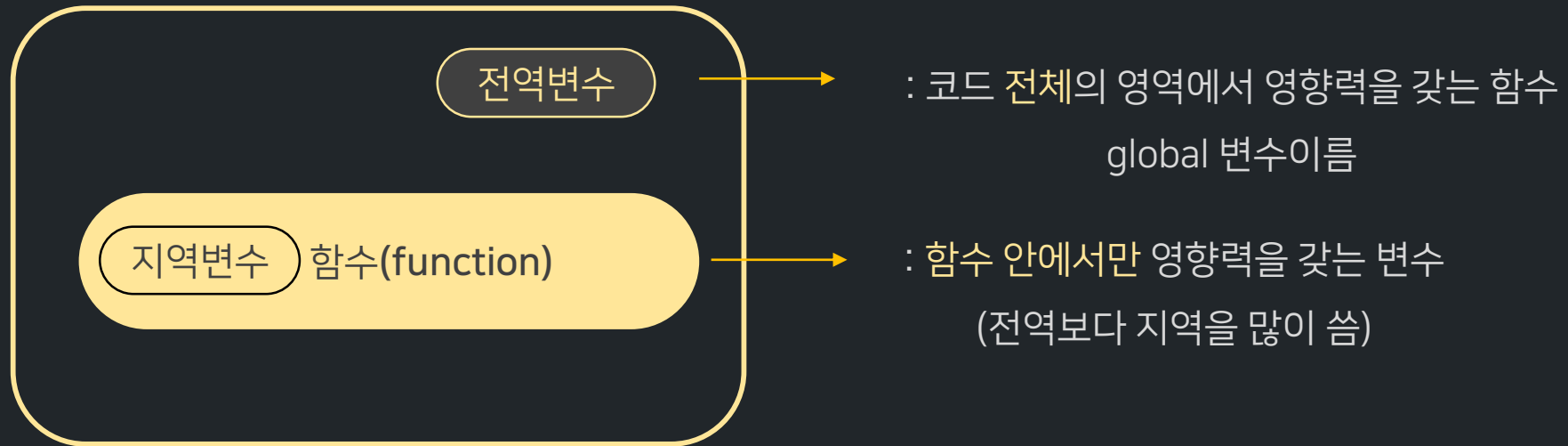
# 리턴값이 없는 함수
def printHello(name) :
    print(name)

# 리턴값이 있는 함수
def sumNumber(num1,num2) :
    return num1+num2
```

# 기초문법

왜 모든 개발자들은 깃헙을 사용하나요?

## 04. 함수



# Pip 가상환경 만들기

: 가상환경을 왜 만들어야 되는거지?

\* 해당 강의는 파이썬 3.6 이상의 버전이 설치되었음을 가정하고 진행합니다. 안되신 분들은 > [링크](#) 에서 설치해주세요!

## 가상 환경

### 프로젝트 A

python 3.4 실행 파일  
(인터프리터)

Lib/site-packages

패키지X 1.5

스크립트A.py

### 프로젝트 B

python 3.6 실행 파일  
(인터프리터)

Lib/site-packages

패키지X 2.0

스크립트B.py



# Pip 가상환경 만들기

: 가상환경을 왜 만들어야 되는거지?

\* 해당 강의는 파이썬 3.6 이상의 버전이 설치되었음을 가정하고 진행합니다.

## Pip 설치

맥> sudo easy\_install pip

윈도우> easy\_install pip

\* 환경변수에 포함되어있어야 합니다!

```
OK@DESKTOP-OK38BF2: ~$ easy_install pip
Searching for pip
Best match: pip 20.2.4
Adding pip 20.2.4 to easy-install.pth file
Installing pip-script.py script to c:\users\ok\appdata\local\programs\python\python38\Scripts
Installing pip.exe script to c:\users\ok\appdata\local\programs\python\python38\Scripts
Installing pip3-script.py script to c:\users\ok\appdata\local\programs\python\python38\Scripts
Installing pip3.exe script to c:\users\ok\appdata\local\programs\python\python38\Scripts
Installing pip3.8-script.py script to c:\users\ok\appdata\local\programs\python\python38\Scripts
Installing pip3.8.exe script to c:\users\ok\appdata\local\programs\python\python38\Scripts

Using c:\users\ok\appdata\local\programs\python\python38\lib\site-packages
Processing dependencies for pip
Finished processing dependencies for pip
```

# Django 개발환경 세팅하기


: Django를 위한 준비 끝!

## 가상환경 생성

python -m venv (가상환경 명)

```
ok@DESKTOP-DR38BF2 MINGW64 ~/Desktop/test
$ python -m venv myvenv
```

✓ TEST

>  myvenv

## 가상환경 실행

맥> source myvenv/bin/activate

윈도우> source myvenv/Scripts/activate

```
ok@DESKTOP-DR38BF2 MINGW64 ~/Desktop/test
$ source myvenv/Scripts/activate
(myvenv)
ok@DESKTOP-DR38BF2 MINGW64 ~/Desktop/test
$
```

*\* Django 개발을 할 때는 꼭 가상환경을 켜고 시작할 것!*

# Django 개발환경 세팅하기

: Django를 위한 준비 끝!

## 장고 설치

pip install django

또는

Pip install django == 특정버전

```
(myvenv)
ok@DESKTOP-DR38BF2 MINGW64 ~/Desktop/test
$ pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/b8/6f/9a4415cc4fe9228e26ea53cf2005961799b2abb8da0411e519fdb74754fa/Django-3.1.7-py3-none-any.whl (7.8MB)
    |████████████████████| 7.8MB 1.7MB/s
Collecting sqlparse>=0.2.2 (from django)
  Using cached https://files.pythonhosted.org/packages/14/05/6e8eb62ca685b10e34051a80d7ea94b7137369d8c0be5c3b9d9b6e3f5dae/sqlparse-0.4.1-py3-none-any.whl
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/70/94/784178ca5dd892a98f113cdd923372024dc04b8d40abe77ca76b5fb90ca6/pytz-2021.1-py2.py3-none-any.whl (510kB)
    |████████████████████| 512kB ...
Collecting asgiref<4,>=3.2.10 (from django)
  Downloading https://files.pythonhosted.org/packages/89/49/5531992efc62f9c6d08a7199dc31176c8c60f7b2548c6ef245f96f29d0d9/asgiref-3.3.1-py3-none-any.whl
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.3.1 django-3.1.7 pytz-2021.1 sqlparse-0.4.1
WARNING: You are using pip version 19.2.3, however version 21.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
(myvenv)
ok@DESKTOP-DR38BF2 MINGW64 ~/Desktop/test
$
```

\* Django 개발을 할 때는 꼭 가상환경을 켜고 시작할 것!

# 함께보면 좋은 자료

구글링은 나의 힘!

<https://drive.google.com/file/d/17J1e82GqNNA8Ye2ZeAHGFwcG53V1D0Jj/view?usp=sharing>

파이썬 기초 colab

<https://www.notion.so/adccc1ec978f4893929ea5379a6894ec?v=2fe4eb6dcf9e42518cca401e998538db>

멋사 라이브러리 파이썬