

# 파이썬 플라스크로 배우는 웹프로그래밍 #1

## ABCD Foundation



스노우키위, 한 성 일

# 0. 프롤로그

왜?

• 디자이너와의 친분



• 많아짐



• 스타트업 관계자들만 관심을 보임



# 기존 스테디와 다른점

- 개발 프로젝트 과정 전체를 본다.



- 개발자 체험 (개발자와의 대화 능력 향상)



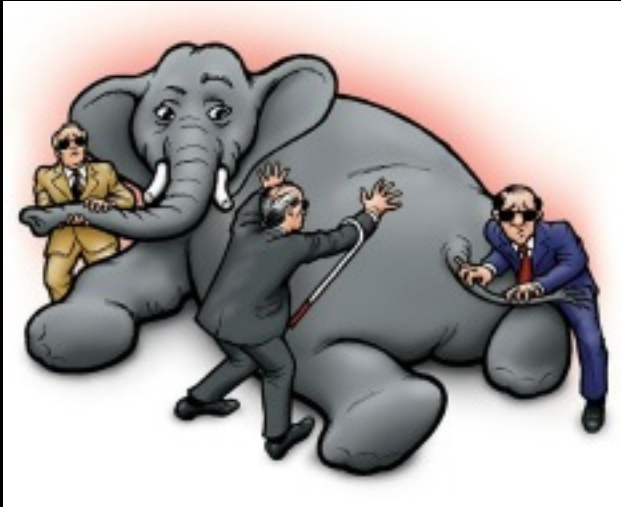
- 머리에 쥐가남





# 목표

- 프로젝트 전체를 보기



- 무조건 만들기 + 서비스 해보기



- 마이크로 프로필  
(커뮤니케이션 가능한..)



# 목표

- 혼자 하는 방법을 배운다.
- 코딩을 한다 못한다는 종이 한장 차이
- 못한다고 생각하면 절대 못한다.
- 생각보다 재미있다.
- 개발자도 더듬더듬 과정을 거친다.
  - You can do it!

# 스터디 멤버들

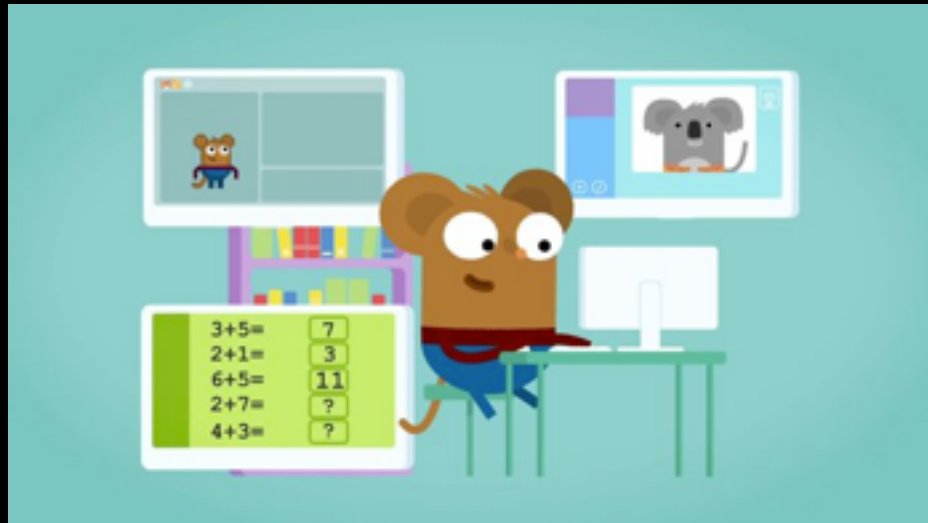
- 운영진 소개
- 자기소개

# 1. 프로그래밍(코딩)이란?

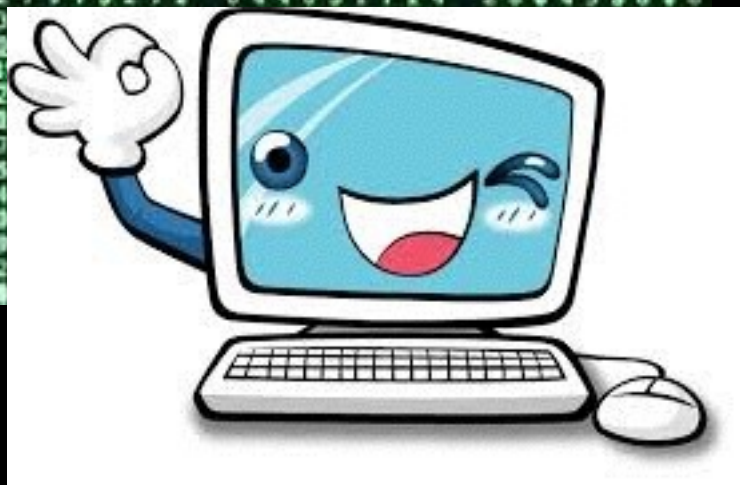
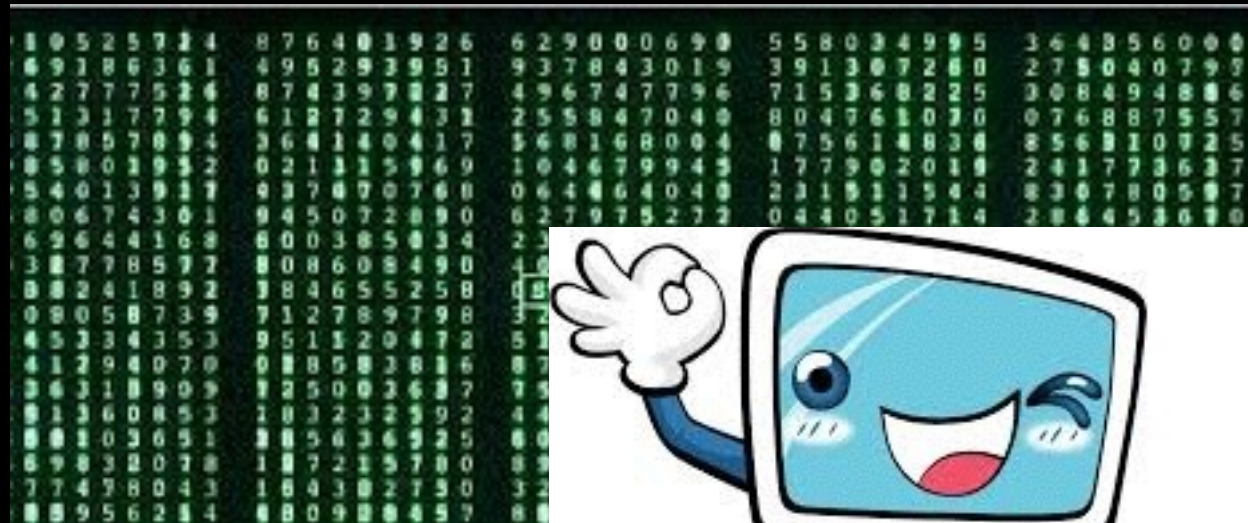


# 프로그래밍이란?

## 1. 컴퓨터에게 뭔가를 시키는 과정



## 2. 컴퓨터는 0 1 만 알아 먹음



## 3. 사람이 쓰는 언어를 컴퓨터가 알수 있도록 번역



# 프로그래밍을 배워야 하는 이유는?



<http://youtu.be/SESuctdE9vM>

## 2. 개념



# 파이썬(Python) 이란?



- 프로그래밍 언어 쉽고 다방면에 사용가능
- 인터프리터 언어

<https://mirror.enha.kr/wiki/Python>

- 귀도 반 로섬(Guido van Rossum)  
크리스마스  
TV 프로그램

# 웹서비스 구조도





# 개발환경

- 서버에서의 처리를 배움
- 로컬 PC (노트북)에 서버설치
- 파이썬
- 플라스크 설치

### 3. 파이썬 설치

개발툴

사용 툴

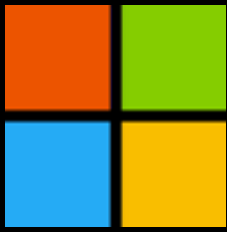


- Brackets  
에디터

<http://brackets.io/>

- 그외 (편리한)





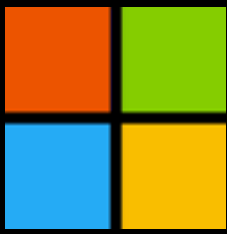
# 파이썬 + Flask 설치

## 1. 파이썬 다운로드

<https://www.python.org/> (파이썬 공식)

**꼭! Python 2.7.8을 받아야 합니다.**

The screenshot shows the Python.org website with the 'Downloads' menu open. The 'All releases' option is selected, and the 'Python 2.7.8' button is highlighted with a red dashed box. The text 'Download for Windows' is visible above the button. Below the button, it says 'Not the OS you are looking for? Python can be used on 21 different operating systems and environments. View the full list.' The footer text reads: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'



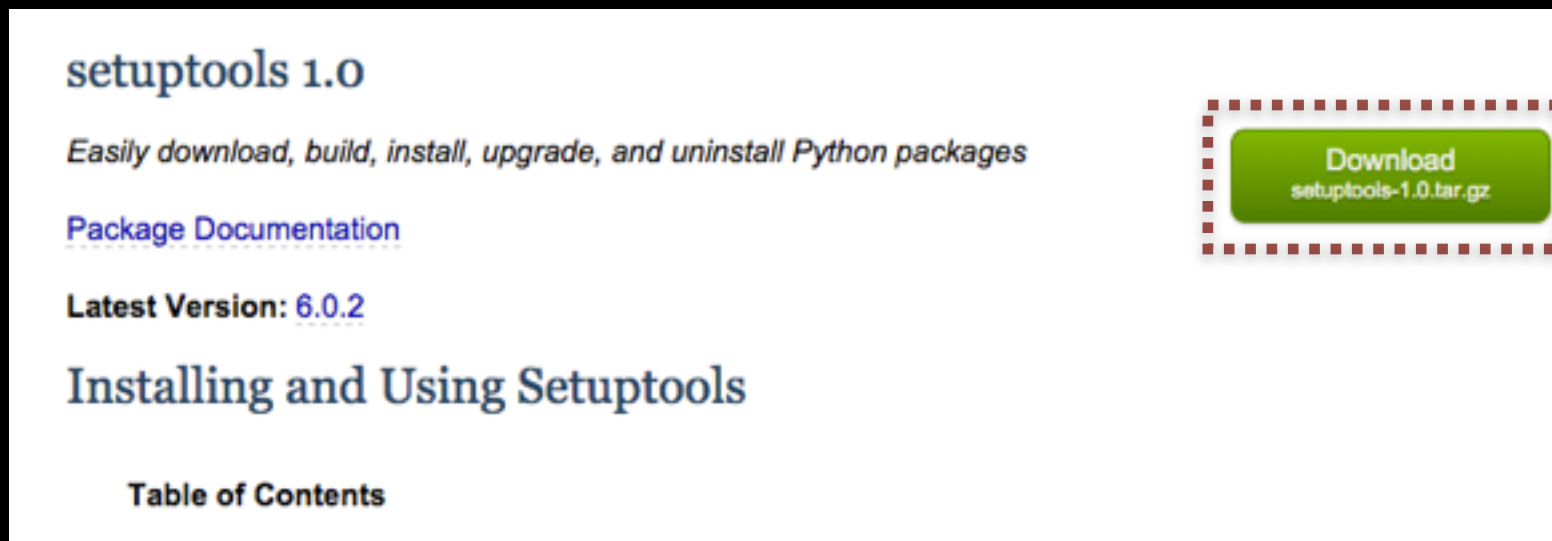
# 파이썬 설치

## 2. python-2.7.8.msi 설치

C:\Python27 경로에 설치됨

## 3-1. setuptools 설치 (파이썬 패키지 관리툴)

<https://pypi.python.org/pypi/setuptools/1.0#windows>

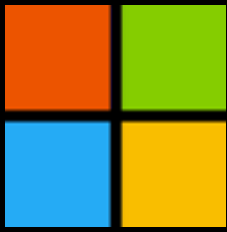


## 3-2. 다운로드 파일 압축해제

ez\_setup.py 더블클릭 (커맨드창이 뜨고 주루룩.. 사라집니다.)

easy_install.py	2013-08-10 오후...	Python File	1KB
ez_setup.py	2013-08-15 오전...	Python File	11KB
launcher.c	2013-08-10 오후...	C 파일	11KB





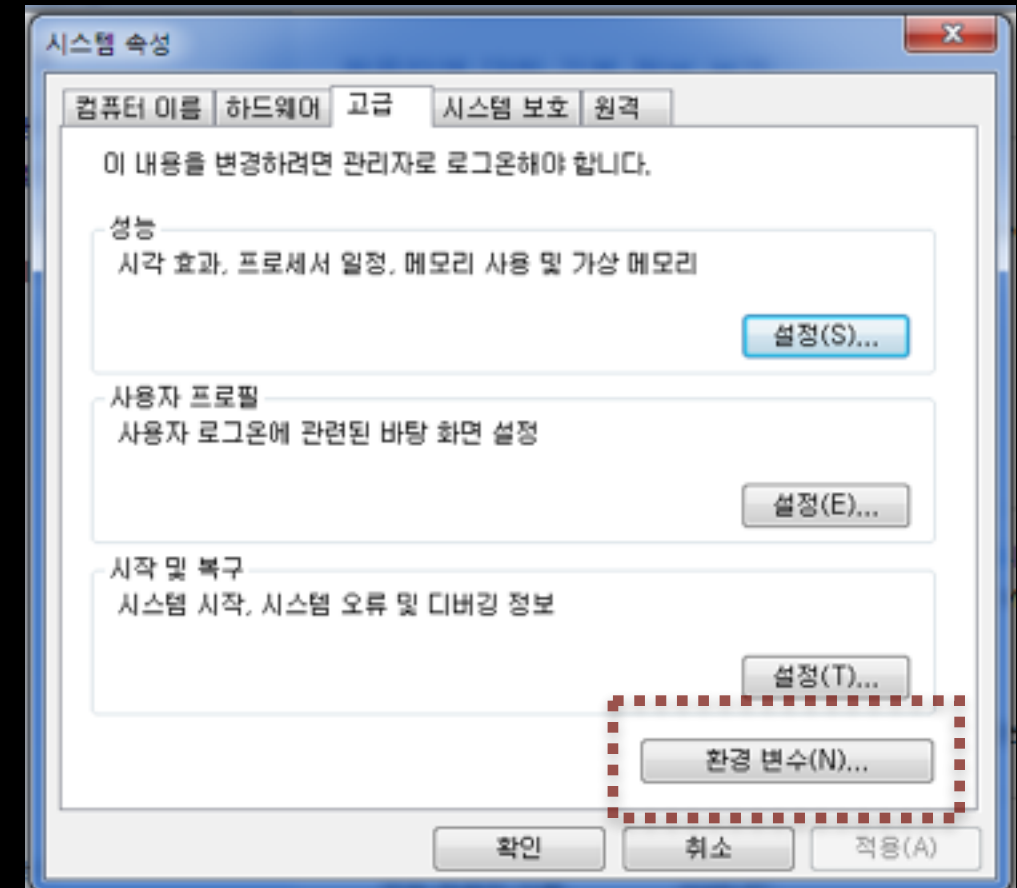
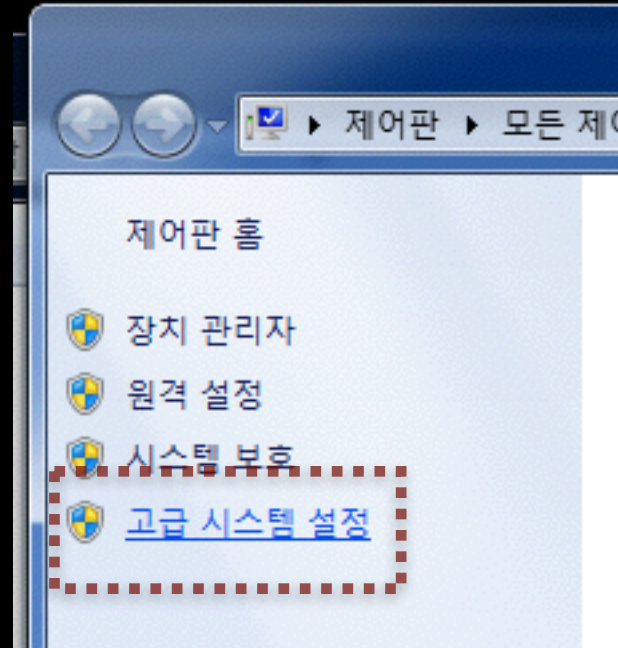
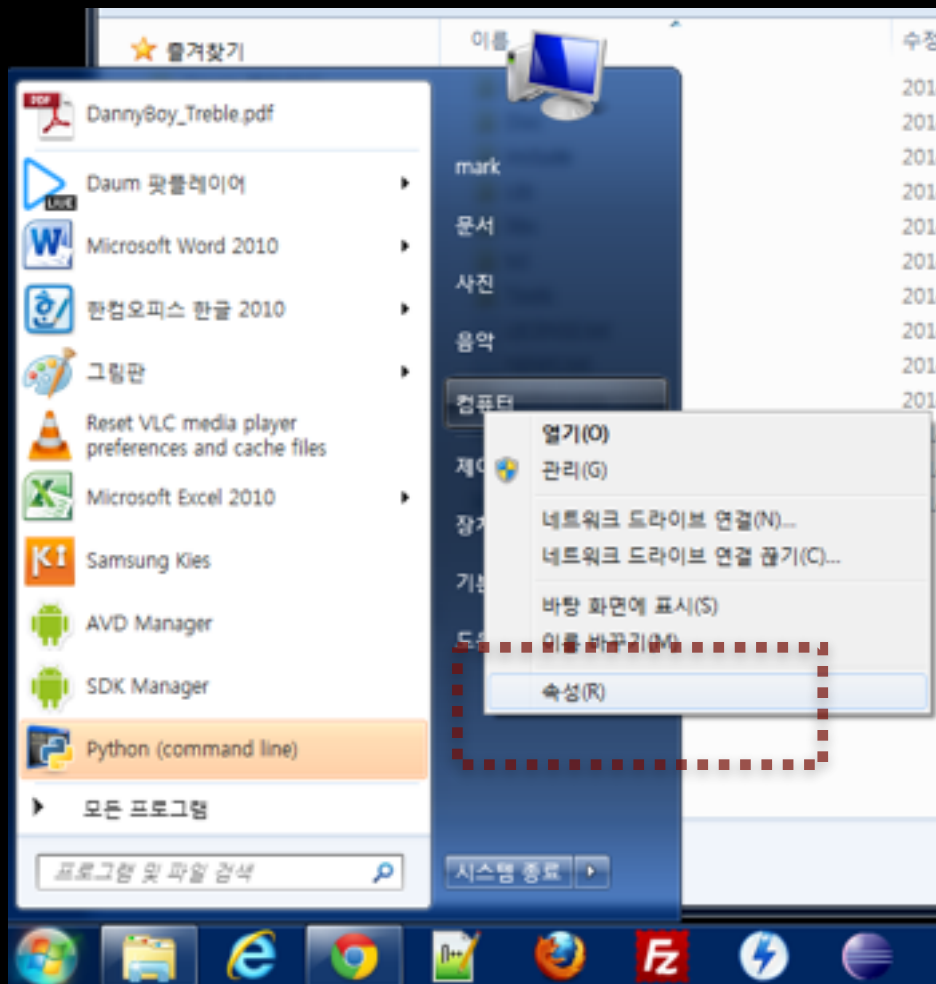
# 파이썬 설치

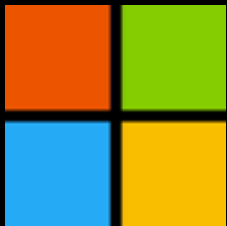
## 4-1. 환경변수 설정

컴퓨터 > 마우스우클릭 > 속성

고급 시스템 설정

환경변수

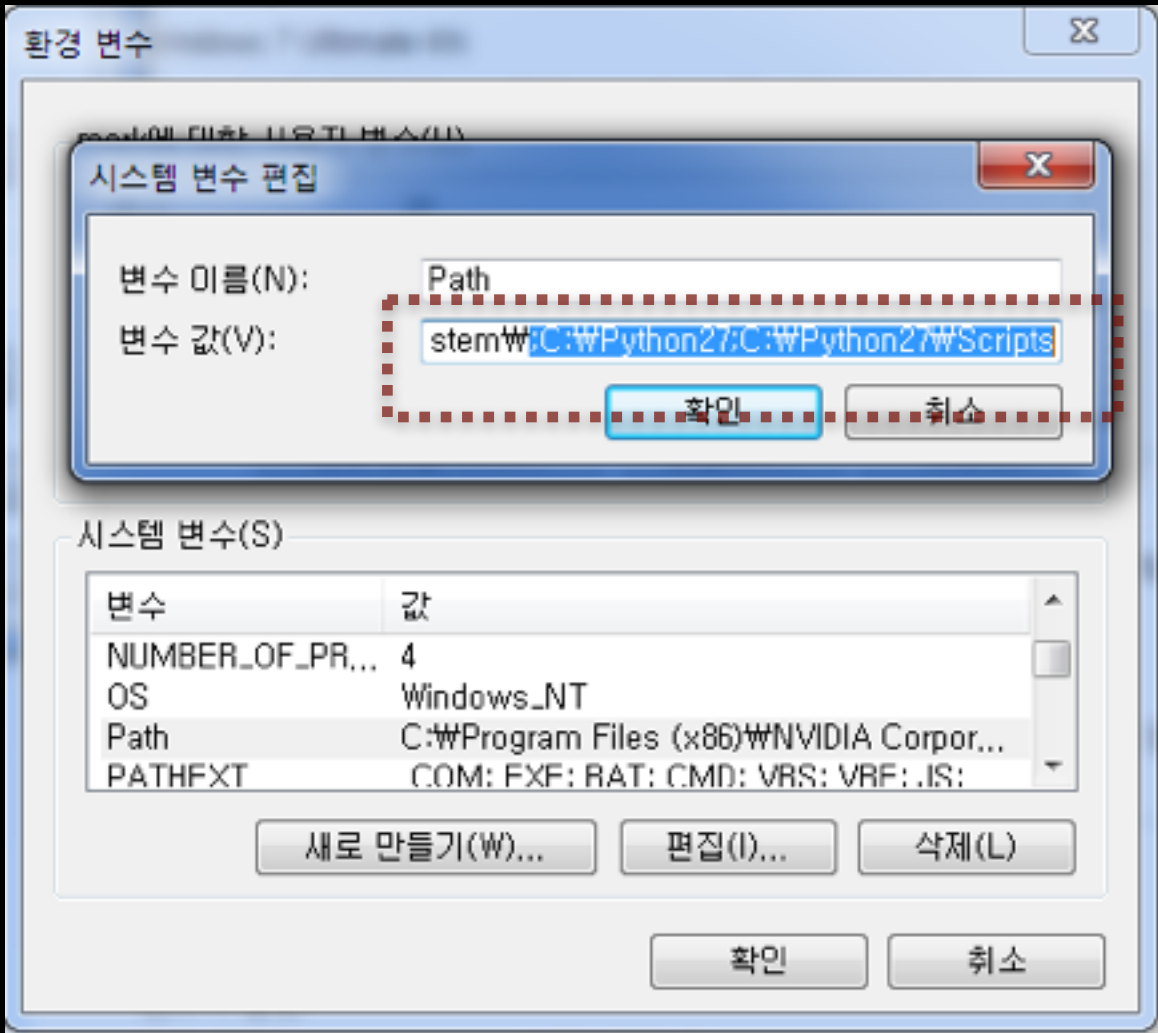




# 파이썬 설치

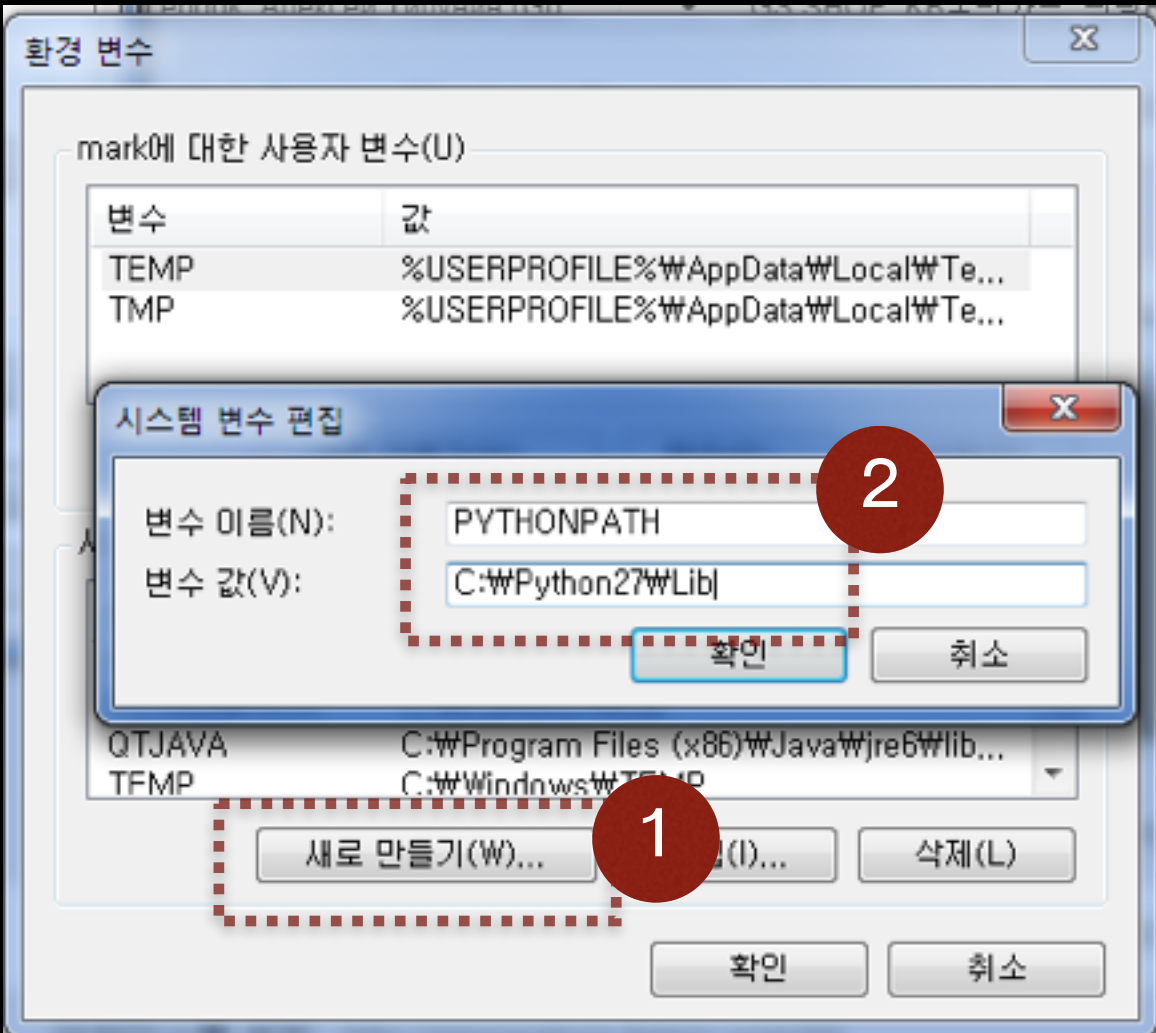
## 4-2. 환경변수 설정

기존 설정 뒷줄에  
;C:\Python27;C:\Python27/Scripts  
추가



## 4-3. 환경변수 설정

새로만들기 > 입력  
PYTHONPATH  
C:\Python27\Lib  
추가



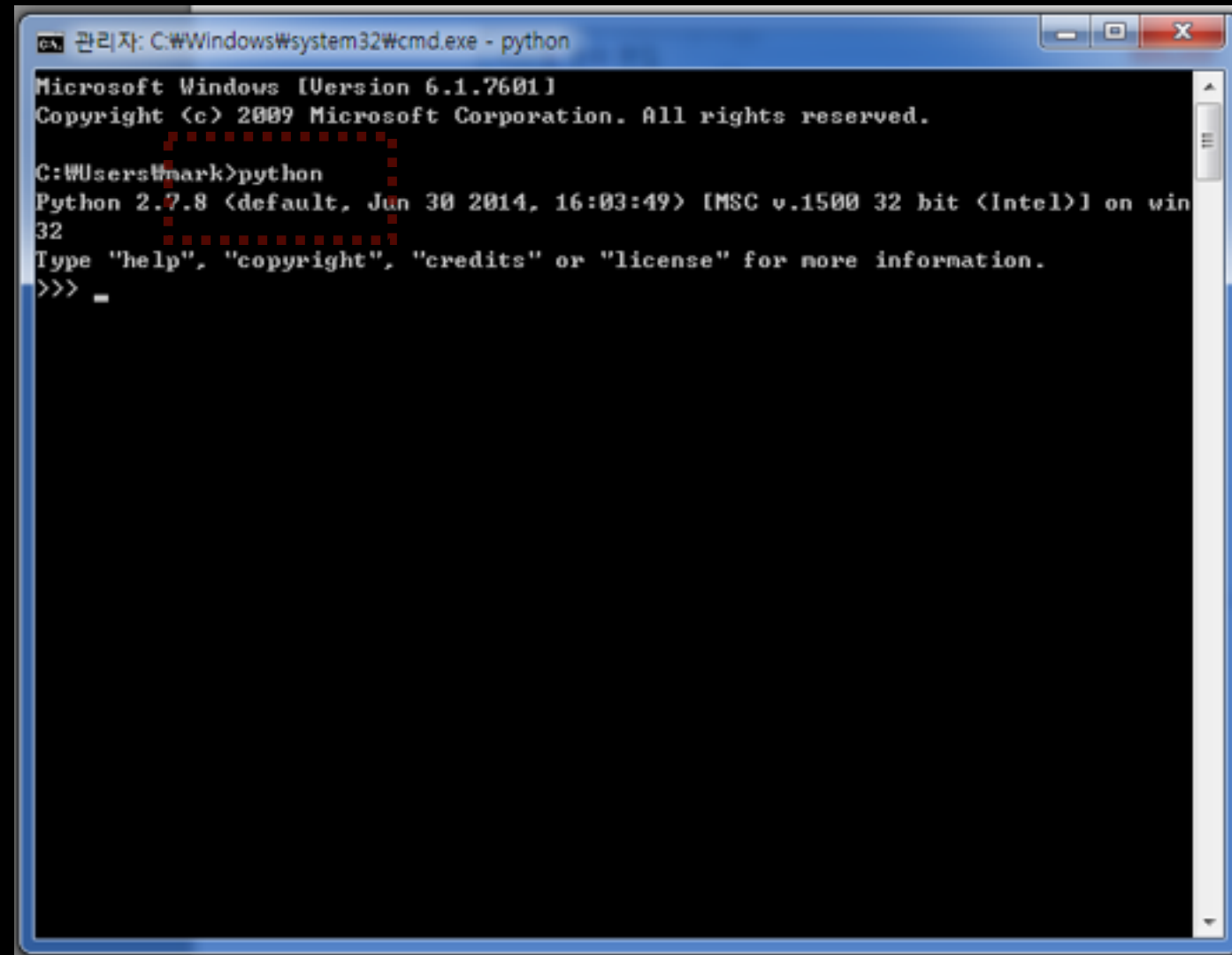
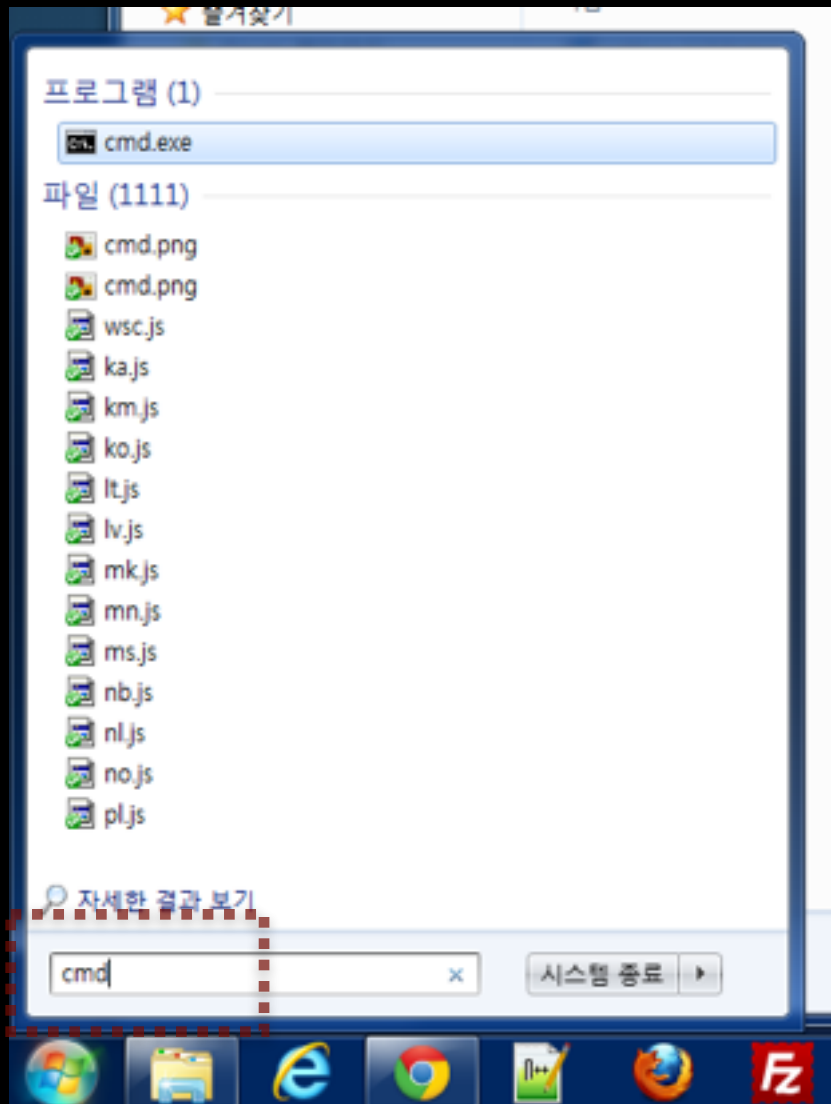


# 행

## 5. 파이썬 설치확인 (IDLE : pure Python)

명령어창 실행  
시작버튼 > 검색창에 cmd

python





# 행

## 6. PIP 설치 (Python 패키지 관리자)

`easy_install pip`

```
C:\workspace>easy_install pip
Searching for pip
Best match: pip 1.5.6
Processing pip-1.5.6-py2.7.egg
pip 1.5.6 is already the active version in easy-install.pth
Installing pip-script.py script to C:\Python27\Scripts
Installing pip.exe script to C:\Python27\Scripts
Installing pip.exe.manifest script to C:\Python27\Scripts
Installing pip2.7-script.py script to C:\Python27\Scripts
Installing pip2.7.exe script to C:\Python27\Scripts
Installing pip2.7.exe.manifest script to C:\Python27\Scripts
Installing pip2-script.py script to C:\Python27\Scripts
Installing pip2.exe script to C:\Python27\Scripts
Installing pip2.exe.manifest script to C:\Python27\Scripts

Using c:\python27\lib\site-packages\pip-1.5.6-py2.7.egg
Processing dependencies for pip
Finished processing dependencies for pip
```

\* pip 에 대해 더 알고 싶다면

<https://pypi.python.org/pypi/pip>



# 파이썬 + 플라스크 설치 (맥OS)

## 1. 터미널 실행







# 파이썬 + Flask 설치 (맥OS)

## 2. 파이썬 확인 (IDLE : pure Python)

\$python

```
jamieui-MacBook-Pro:~ jamie$ python
Python 2.7.5 (default, Mar 9 2014, 22:15:05)
[GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## 3. PIP 설치 (Python 패키지 관리자)

\$sudo easy\_install pip

```
jamieui-MacBook-Pro:~ jamie$ sudo easy_install pip
Password:
Searching for pip
Best match: pip 1.5.6
Processing pip-1.5.6-py2.7.egg
pip 1.5.6 is already the active version in easy-install.pth
Installing pip script to /usr/local/bin
Installing pip2.7 script to /usr/local/bin
Installing pip2 script to /usr/local/bin

Using /Library/Python/2.7/site-packages/pip-1.5.6-py2.7.egg
Processing dependencies for pip
Finished processing dependencies for pip
jamieui-MacBook-Pro:~ jamie$
```

\* pip 에 대해 더 알고 싶다면  
<https://pypi.python.org/pypi/pip>

## 4. 프로그래밍 기초 (Python)

\* 참조

<http://www.stavros.io/tutorials/python/>

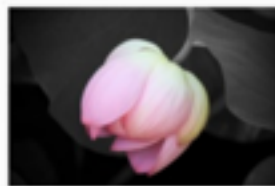
# 파이썬 기초와 문법

“돈을 넣으면 캔이 나온다.”  
- 함수

“내가 이름 지어주기 전에는 그는  
다만 메모리 주소에 지나지 않았다.”  
- 변수

## 김춘수 시인의 꽃

내가 그의 이름을 불러주기 전에는  
그는 다만 하나의 몸짓에 지나지 않았다.  
내가 그의 이름을 불러 주었을때  
그는 나에게로 와서 꽃이 되었다.  
내가 그의 이름을 불러 준 것 처럼  
나의 이 빗칼과 향기에 알맞는  
누가 나의 이름을 불러다오  
그에게로 가서 나도 그의 꽃이 되고싶다  
우리들을 모두 무엇이 되고싶다.  
나는 너에게, 너는 나에게  
잊혀지지 않는 하나의 의미가 되고 싶다.





# 파이썬 기초와 문법 (실습)

## 0. 기본

### 변수선언

```
myString = "a"
```

- \* 문법을 쓸때 꼭 탭으로 인덴트(탭 : 들여쓰기)
- \* #으로 시작하는건 주석 (주석은 메모 실행되지 않음)



# 파이썬 기초와 문법 (실습)

## 1. 변수 (Variable)

```
>>> myvar = 3
>>> myvar += 2
>>> myvar
5
>>> myvar -= 1
>>> myvar
4
>>> mystring = "Hello"
>>> mystring += " world."
>>> print mystring
Hello world.
>>> myvar, mystring = mystring, myvar
>>> print myvar
Hello world.
>>> print mystring
4
```





# 파이썬 기초와 문법

## 2. 제어문

```
animals = ["cat", "dog", "lion", "rabbit"]
>>> print animals
for item in animals:
    print item
```

```
if animals[1] == "cat":
    print "고양이다"
elif animals[1] == "dog":
    print "개다"
else:
    print "뭐지"
```

```
while animals[1] == "dog":
    print animals[1]
    pass
# 무한루프
```



# 파이썬 기초와 문법

## 3. 함수 (Functions)

```
def sum(a, b, c):  
    return a+b+c
```

```
>>> sum(1, 2, 3)
```

```
def add(a, b, c=" alive"):  
    return a + b + c
```

```
>>> add("I ", "am")
```

```
>>> def sum(a, b, c):  
...     return a+b+c  
...  
>>> sum(1, 2, 3)  
6  
>>>
```

## 5. 플라스크(Flask) 설치

\* 참조

<http://flask-docs-kr.readthedocs.org/ko/latest/installation.html>



# 플라스크 설치

## OS 별로 따로 보세요!!

### 1-1. 윈도우 (C:\flask-space)

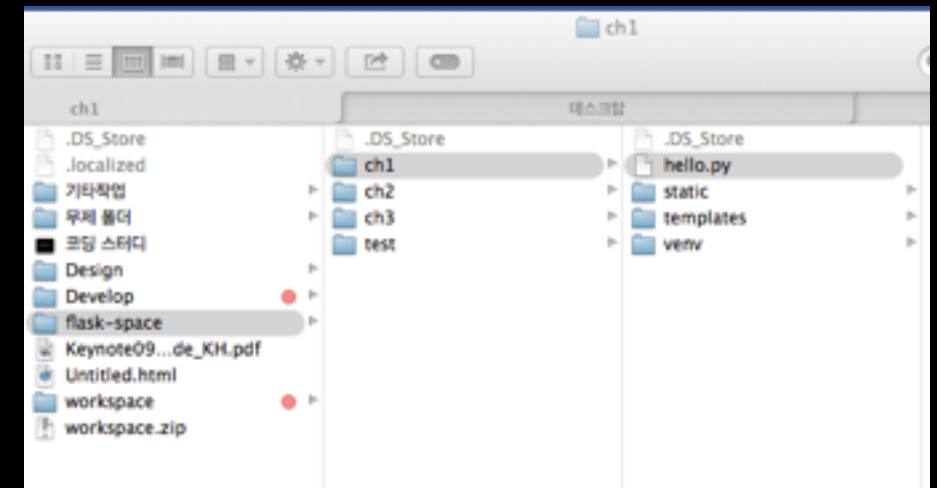
```
pip install virtualenv
mkdir C:\flask-space\ch1
cd C:\flask-space\ch1
virtualenv venv #virtualenv를 사용한다면
pip install flask
```

명령어 창에서 flask-space 로 가려면  
cd C:\flask-space\ch1

### 1-2. 맥 OS (도큐먼트 / flask-space)

```
sudo pip install virtualenv
sudo mkdir ~/Documents/flask-space/ch1
sudo chmod 777 ~/Documents/flask-space/ch1
cd ~/Documents/flask-space/ch1
sudo virtualenv venv #virtualenv를 사용한다면
sudo pip install flask
```

cd ~/Documents/flask-space/ch1





# 플라스크 설치 (실습)

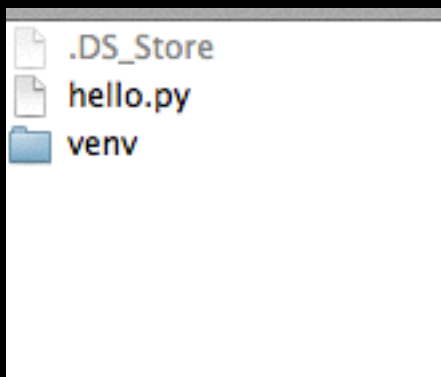
## 2. 기본 어플리케이션 생성

hello.py 로 저장 (flask-space/ch1)

```
from flask import Flask
app = Flask(__name__)
```

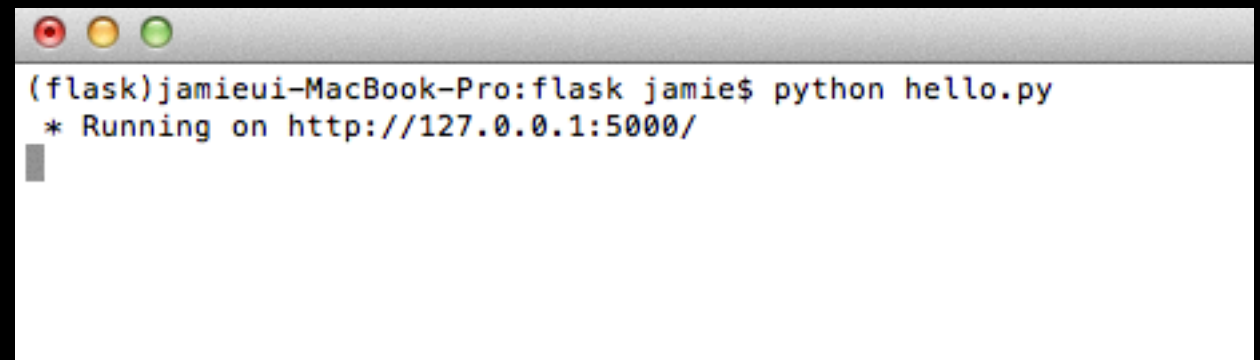
```
@app.route('/')
def hello_world():
    return 'Hello World!'
```

```
if __name__ == '__main__':
    app.debug = True
    app.run()
```



## 3. 내장서버 구동 & 테스트

hello.py 경로에서 아래 명령어로 내장서버 구동  
python hello.py



테스트는 브라우저에서 아래 경로 입력  
http://127.0.0.1:5000/



## 6. 플라스크(Flask) 실습

\* 참조

<http://flask-docs-kr.readthedocs.org/ko/latest/quickstart.html>

# 플라스크(Flask)는?

## 1. 기본개념

- Python이 웹서비스가 가능하도록 만들어주는 프레임워크
- 설정이 쉽다?
- 파일 하나로 기능을 제공하는 마이크로 프레임워크
- 필요한 기능이 있다면 그때 그때 기능 추가 가능 (pip)





# 플라스크 사용

## 라우팅(Routing)

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import sys
reload(sys)
sys.setdefaultencoding('utf8')
```

한글 깨짐 방지  
복사해서 사용하세요.

```
from flask import Flask
app = Flask(__name__)
```

플라스크 객체 실행

```
@app.route('/')
def hello_world():
    return 'Hello World!'
```

라우팅 설정

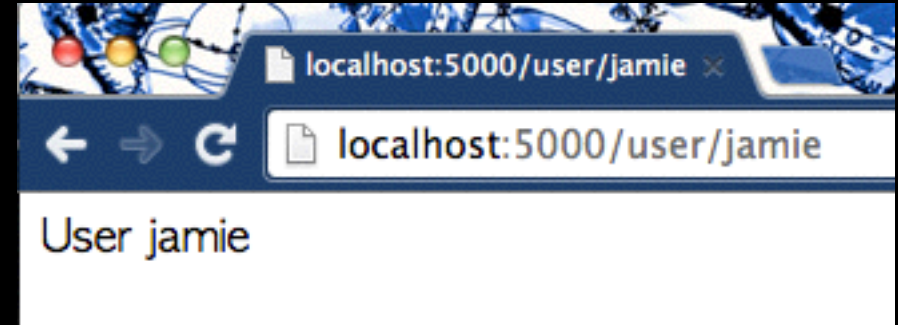
컨트롤러 설정

```
@app.route('/main')
def main():
    return 'main page'
```

```
if __name__ == '__main__':
    app.run()
```

내장서버 실행

# 플라스크 사용 파라미터(Parameter)



```
@app.route('/user/<username>')  
def show_user_profile(username):  
    return 'User %s' % username
```

username 으로 값을 받아 출력  
%s 는 문자

```
@app.route('/post/<int:post_id>')  
def show_post(post_id):  
    return 'Post %d' % post_id
```

<int:post\_id> 로 숫자값만 받도록 설정  
%d는 숫자(정수)

# 플라스크 사용 로깅(Logging)

```
@app.route('/logging')
def logging_test():
    test = 1
    app.logger.debug('디버깅 필요')
    app.logger.warning(str(test) + " 라인")
    app.logger.error('에러발생')
    return "로깅 끝"
```

콘솔에만 표시할때 사용  
개발자가 필요한 정보를 출력 할 수 있음

```
-----
DEBUG in hello1 [hello1.py:28]:
디버깅 필요
-----
WARNING in hello1 [hello1.py:29]:
1 번 줄 위험
-----
ERROR in hello1 [hello1.py:30]:
에러발생
-----
127.0.0.1 - - [10/Oct/2014 00:19:00] "GET /logging HTTP/1.1" 200 -
```



## 플라스크 사용 로그인 (Post, Session)

render\_template, request, session # import 에 추가 해야 함

```
@app.route('/login_form')
def login_form():
    return render_template('login_form.html')

@app.route('/login', methods=['POST'])
def login():
    if request.method == 'POST':
        if ( request.form['username'] == 'jamie'
            and request.form['password'] == '1234' ):
            session['logged_in'] = True
            session['username'] = request.form['username']
            return request.form['username'] + " 님 환영합니다."
        else:
            return '로그인 정보가 맞지 않습니다.'
    else:
        return '잘못된 접근'

app.secret_key = 'sample_secret_key'
```

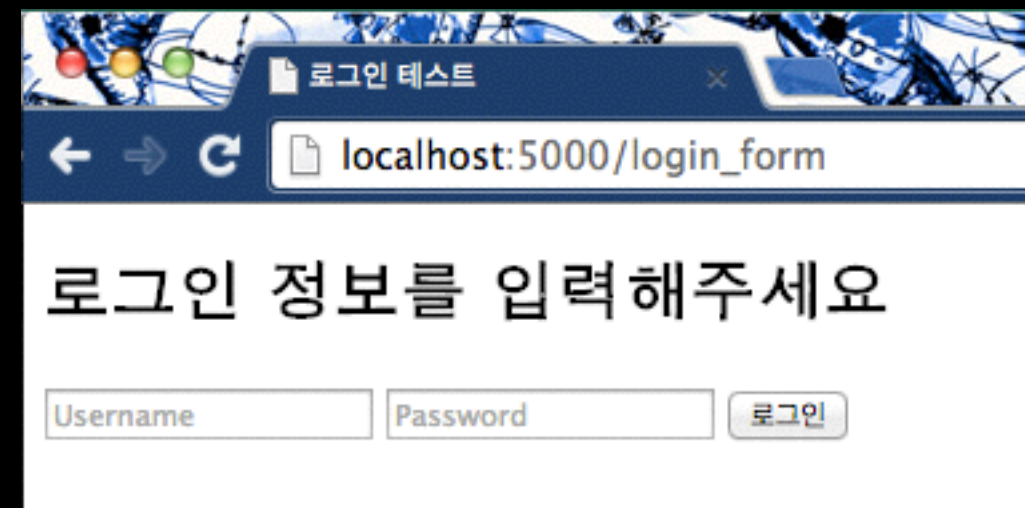
- session 과 request 가 사용되었음
  - app.secret\_key = 'abc'
- 세션키를 꼭추가해야함**

# 플라스크 사용

## 로그인

저장 위치 : templates/login\_form.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>로그인 테스트</title>
  </head>
  <body>
    <form method="post" action="/login">
      <h2>로그인 정보를 입력해주세요</h2>
      <input type="text" name="username" placeholder="Username">
      <input type="password" name="password"
placeholder="Password">
      <button type="submit">로그인</button>
    </form>
  </body>
</html>
```



# 플라스크 사용

## GET을 이용한 로그인 (request, session)

실제 서비스에선 절대 GET방식 로그인을 하면 안됨

```
@app.route('/get_test', methods=['GET'])
def get_test():
    if request.method == 'GET':
        if (request.args.get('username') == "jamie"
            and request.args.get('password') == "1234"):
            return request.args.get('username') + " 님 환영합니다."
        else:
            return '로그인 정보가 맞지 않습니다.'
    else:
        return '잘못된 접근'
```



# 플라스크 사용

## 로그아웃 (session)

`request, redirect, url_for, session` # import **에 추가해야 함**

```
@app.route('/logout')
def logout():
    session['logged_in'] = False
    session.pop('username', None)
    return redirect(url_for('index'))
```

# 플라스크 템플릿 (jinja2)

## 템플릿 테스트

```
@app.route('/template')
@app.route('/template/<tempid>')
def template_test(tempid=None):
    sports = ['야구', '축구', '농구']
    return render_template('template.html', tempid=tempid, sports=sports)
```

# 플라스크 템플릿 (jinja2)

## 템플릿 테스트

**저장 경로 :** templates/template.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>템플릿 테스트</title>
  </head>
  <body>
    <div>
      <h3> - 파라미터 : {{ tempid }} </h3>
      {% if not session.logged_in %}
        <h3> <a href="{{ url_for('login_form') }}">로그인</a> 해주세요. </h3>
      {% else %}
        <h3> <a href="{{ url_for('logout') }}">로그아웃</a> </h3>
      {% endif %}
      <h3> - 사람들이 좋아하는 운동 </h3>
      {% for i in sports %}
        <h3> {{ i }} </h3>
        {% if i == '축구' %}
          <h5> {{ i }}는 아닌데.. </h5>
        {% endif %}
      {% endfor %}
    </div>
  </body>
</html>
```

## 7. 예고편

# 부트스트랩 + Mysql

**Bootstrap - 프론트 앤드 제작툴**

<http://bootstrapk.com/BS3/>  
<https://wrapbootstrap.com/>

The Bootstrap logo, featuring the word "Bootstrap" in a white, sans-serif font, centered on a dark purple rectangular background.

**Mysql - 데이터베이스**

**데이터베이스 기본 설정과 기본 SQL**



# 부트스트랩

## 깔끔한 디자인의 화면..!!

### Please sign in

Sign in



**Q&A**



수고하셨습니다. :D

- ABCD Foundation  
<https://www.facebook.com/groups/562787713823026/>
- 한성일  
<https://www.facebook.com/jamie.han.16>  
[iheart79@gmail.com](mailto:iheart79@gmail.com)