Flask

일단, Flask 를 쓰기 위해선 라즈베리파이에 Flask 가 깔려있어야한다.

항상, 파이썬 웹개발자들은 프로젝트 만들 때 virtualenv 부터 세팅한다.

? 왜 아나콘다로 하지 않을까? 웹개발자들의 세상에선 데이터 분석가들이 애용하는 아나콘다 대신 virtualenv 를 훨씬 많이 쓰기 때문

파이엔 virtualenv 가 깔려있지 않을 것이므로, 설치부터.

$ sudo apt install python3-pip

$ sudo pip3 install virtualenv

그리고 프로젝트 디렉토리 하나 만든 후, 안에 들어가서

$ virtualenv -p python3 .

맨 끝에 쩜을 붙여줘야한다. 해당 프로젝트 디렉토리와 동일한 이름의 가상환경을 만들 것이다.

가상환경 동작

$ source bin/activate

이제, 가상환경이 갖추어진 상태에서 flask 를 설치하자.

$ pip install flask

보통, 패키지 설치 한 번 해주고 가상환경을 껐다 켜주는게 좋다.

$ deactivate

$ source bin/activate

현재 가상환경의 패키지리스트를 보고싶다면,

$ pip freeze

그럼 현재 flask 의 버전이 확인될 것이다.

이 가상환경에, src 라는 디렉토리 하나 만들고, 이 디렉토리를 루트로 삼고 작업하자.

src 에 hello.py 하나 만들고, 다음 코드 입력

\* 코드를 표기할 땐 """ 를 앞뒤로 감싸겠습니다.

"""

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

def hello():

return "<h1>Hello flask</h1>"

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='0.0.0.0', port=8080)

"""

그리고 다음 명령을 실행

$ export FLASK\_APP=hello

$ flask run

이러면 현재 어느 서버에 붙었는지 나온다. 127.0.0.1:5000 이 디폴트일것이다.

코드를 분석하면,

먼저 Flask 클래스를 가져왔다.

다음으로 클래스의 인스턴스를 만들었다. \_\_name\_\_은 앱의 모듈 또는 패키지 이름이다. 이를 통해 Flask 가 템플릿 및 스태틱 파일 등의 리소스를 찾을 위치를 안다.

route() 는 어떤 url 로 빠지면 해당 함수가 실행되는지이다.

이 함수는 사용자의 브라우저에 표시하려는 메세지를 리턴한다. HTML 태그는 브라우저에서 렌더링될것이다.

그 다음 app.run(host='0.0.0.0', port=8080) 은, 외부에서 8080 에 접속해야 flask 서버에 올린 우리의 hello world 를 볼 수 있다는 뜻이다. 즉, 서버를 공개할 때 사용하는 구문이다. 0.0.0.0 은 모든 외부 컴퓨터를 의미한다.

다음, 페이지를 두 개의 라우터로 나눠보자. 코드 작성하고 flask 서버를 한번 껐다 켜줘야한다

"""

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def root():

return '<h1>This is root page</h1><a href="/next">Go next</a>'

@app.route('/next')

def next():

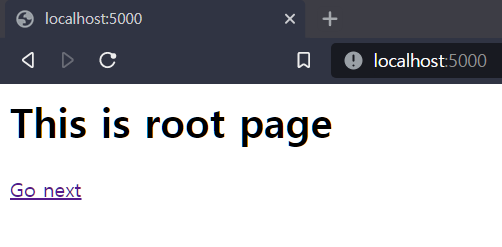
return '<h1>This is Next page</h1><a href="/">Go Root</a>'

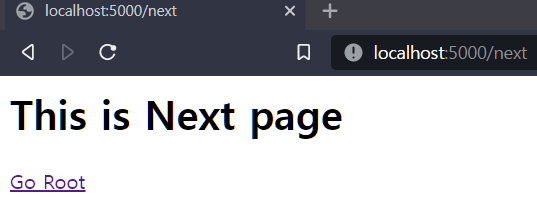
if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='0.0.0.0', port=8080)

"""

결과:





만약, url 에 특정 파라미터에 따라서 보여주는 결과를 다르게 하고 싶다면?

"""

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/user/<id>', methods=['GET'])

def show\_user(id):

print(id)

if id == "abc":

return 'User id is %s, name is %s' % (id, 'Lee')

elif id == "xyz":

return 'User id is %s, name is %s' % (id, 'Kim')

else:

return 'No User id : %s' % id

@app.route('/post/<int:post\_id>')

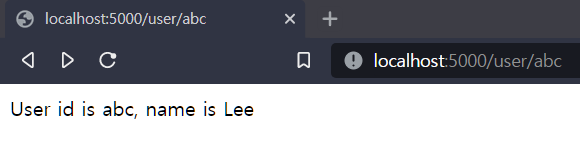
def show\_post(post\_id):

return 'Post id : %d' % post\_id

app.run(host='0.0.0.0', port=8080)

"""

결과:



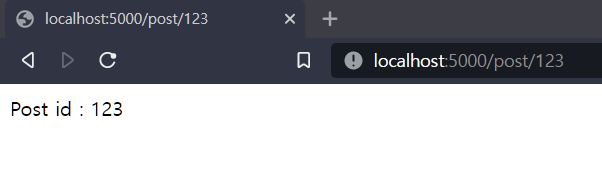
/user/abc 를 넣었을 때 다음과 같은 화면이 뜬다. xyz 일 경우에도 해당 화면이 뜸

만약, abc, xyz 이외일 경우엔 no user id 안내가 뜸

post 일 경우를 보자. 이름만 봐도 추후 post 통신을 위해 사용할것같다.

<int:post\_id> 는 타입을 int 로 고정하겠다는뜻

만약, /post/1234 를 쓰면,



그리고, route 에 등록하지 않고 바로 단순파일서비스도 가능하다. static 이름으로 디렉토리를 생성하고, 그 안에 샘플 html 을 담아보자.

"""

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/<path:path>')

def static\_file(path):

return app.send\_static\_file(path)

@app.route('/')

def root():

return "<h1>this is main page</h1>"

@app.route('/aaa.html')

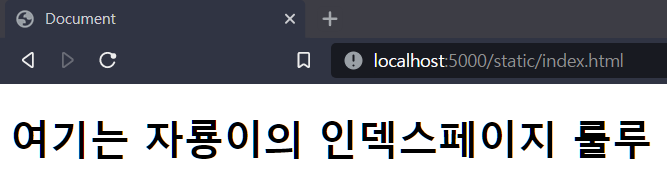
def abc():

return "<h1>this is abc.html</h1>"

app.run(host='0.0.0.0', port=8080)

"""

결과:



좋다. 여기까지 하고, 바로 opencv 프로젝트로 넘어가자.

지금부턴 카메라가 있고, opencv 테스트를 마친 라즈베리파이에 ssh 접속해서, 그곳에

virtualenv 가상환경을 만들고, 다음 프로젝트를 작동시킬것이다.

\* 아래 코드는 작동되지 않는다. 안되는 이유:

pip 로 설치한 opencv 에서 cv 를 못 읽어옴.

cam.js 파일 제공 안하고 node.js 환경설정 관련 내용 전부 다 누락

파이썬 코드 작성하면서 들여쓰기도 불분명하게 적은 코드가 너무 많아 로직이 맞는지조차 알 수 없음

cam.html

"""

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8" />

<title>Insert title here</title>

<style type="text/css">

#img\_wrap {

width: 320px;

height: 240px;

border: 1px solid #000;

}

img:fullscreen {

width: 100%;

height: 100%;

}

img:-webkit-full-screen {

width: 100%;

height: 100%;

}

img:-moz-full-screen {

width: 100%;

height: 100%;

}

</style>

<script type="text/javascript" charset="utf-8">

window.onload = function () {

var img = document.querySelector("#img");

var btn\_start = document.querySelector("#btn\_start");

var btn\_stop = document.querySelector("#btn\_stop");

var b64Prefix = "data:image/png;base64,";

var rate = 1000 / 10;

varsocket = null;

var running = true;

var timer;

btn\_stop.onclick = function () {

clearInterval(timer);

};

btn\_start.onclick = function () {

timer = setInterval(function () {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/cctv");

xhr.onreadystatechange = function () {

if (xhr.readyState == 4) {

img.src = b64Prefix + xhr.responseText;

}

};

xhr.send();

}, rate);

};

img.onclick = function () {

if (img.requestFullscreen) {

img.requestFullscreen();

} else if (img.mozRequestFullScreen) {

img.mozRequestFullScreen();

} else if (img.webkitRequestFullscreen) {

img.webkitRequestFullscreen();

}

};

};

</script>

</head>

<body>

<button id="btn\_start">start</button><button id="btn\_stop">stop</button

><br />

<div id="img\_wrap">

<img src="" id="img" />

</div>

</body>

</html>

"""

cam.py

"""

from flask import Flask, redirect, request

import cv2

import base64

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = 'secret'

cam = cv2.VideoCapture(0)

cam.set(cv2.cv.CV\_CAP\_PROP\_FRAME\_WIDTH, 320)

cam.set(cv2.cv.CV\_CAP\_PROP\_FRAME\_HEIGHT, 240)

@app.route('/')

def main():

return redirect('/static/cam.html')

@app.route('/cctv')

def oncctv():

print('cctv')

ret, data = cam.read()

if ret:

ret, png = cv2.imencode('.png', data)

b64 = base64.encodestring(png)

return b64

return 'camera is not ready.'

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host="0.0.0.0")

"""