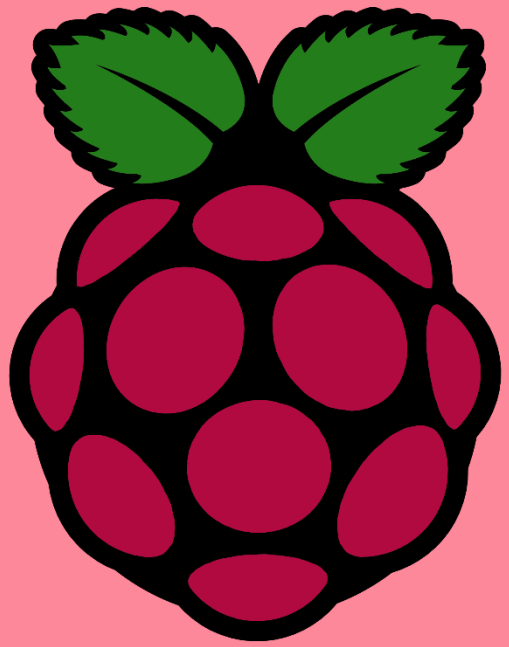
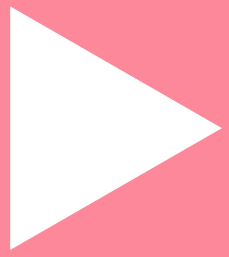


RasberryPi

Smart Media
스마트미디어인재개발원
Smart Media Human Resource Development

스마트미디어인재개발원
나 예 호

라즈베리파이 LED Web으로 제어하기



Python WSGI 마이크로프레임워크 (파이썬으로 웹어플리케이션 제작)

WSGI : Web Server Gateway Interface.

- 서버가 웹 어플리케이션과 통신하기 위한 Interface.
- 최소한의 기능만을 제공하여 유연하게 애플리케이션 작성 가능

1. pip를 이용 Flask 설치

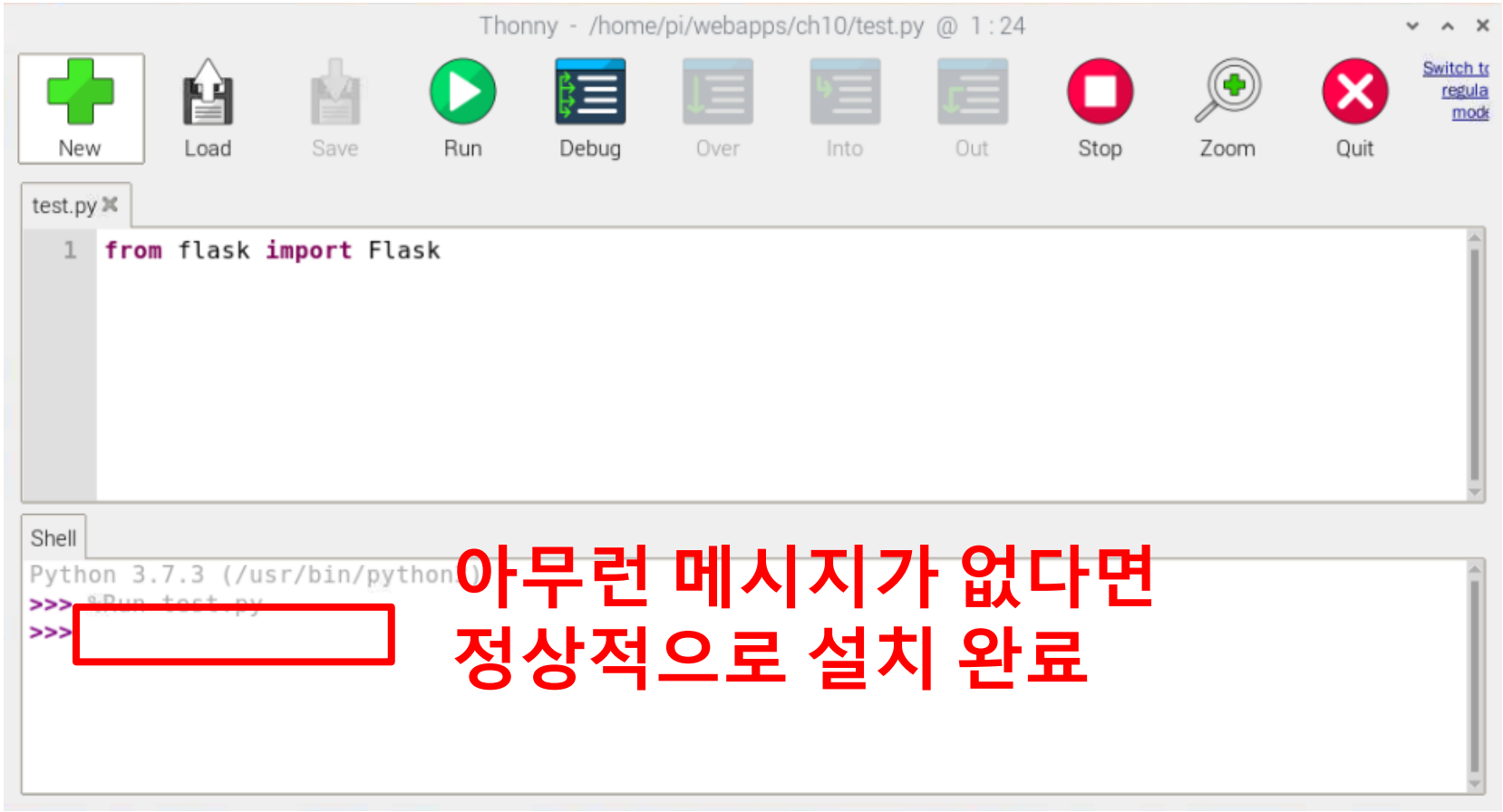
$ sudo pip3 install flask

※ pip버전이 낮아 설치 오류 시

$ sudo pip install –upgrade pip

$ sudo pip install flask

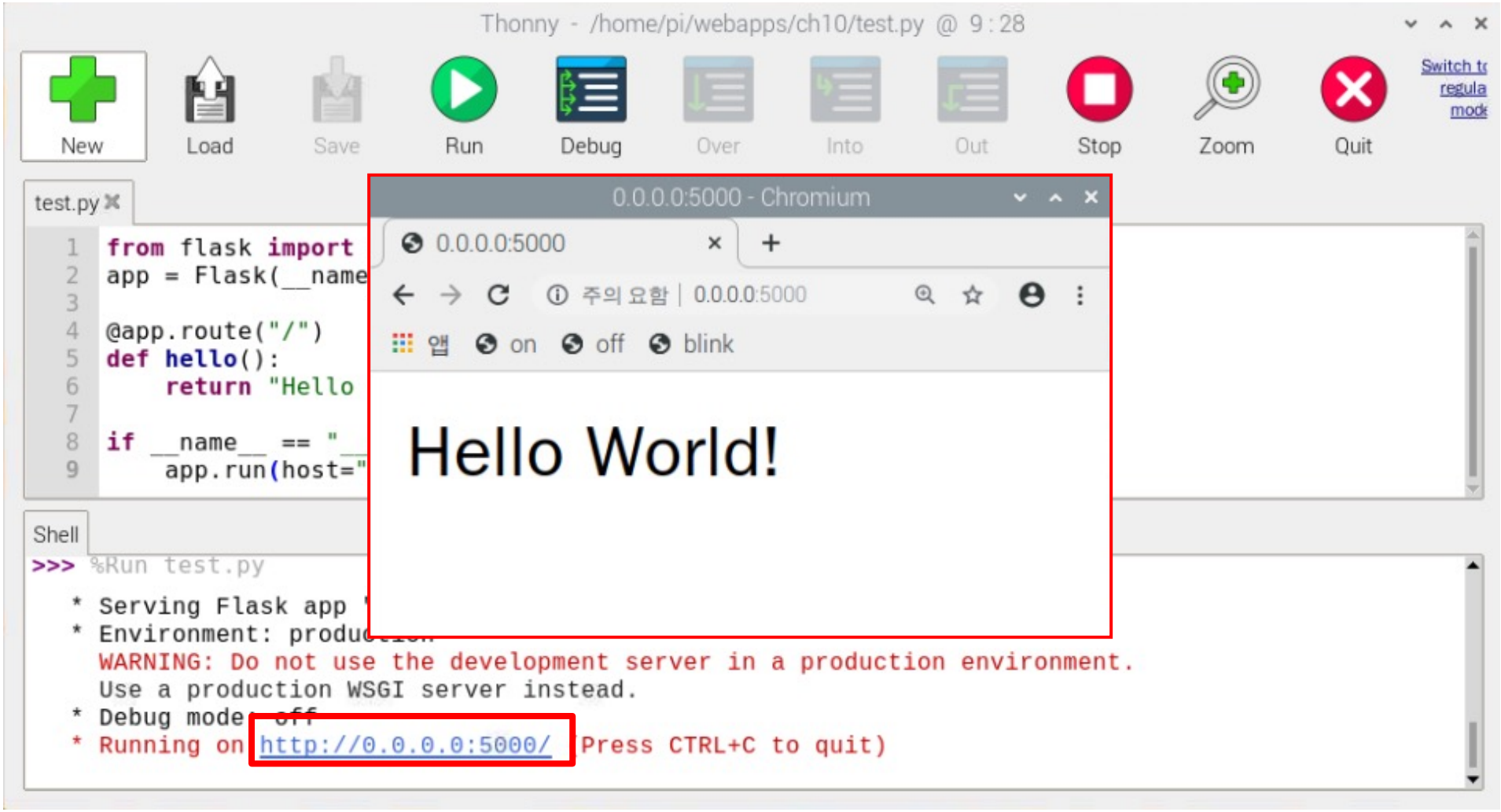
2. Thonny Python IDE 실행

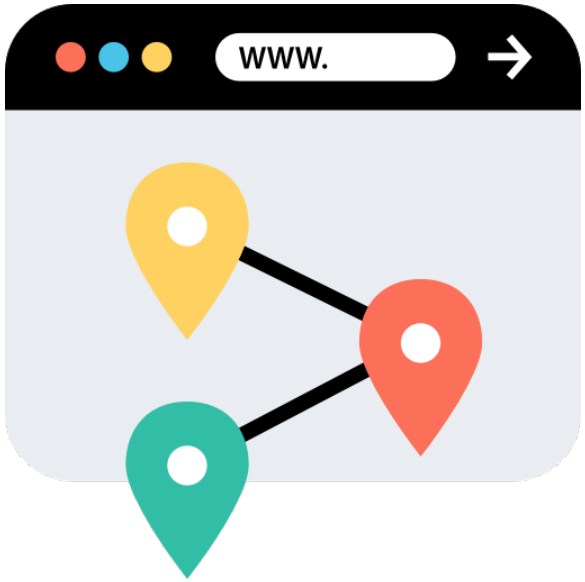


```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
        return "Hello World!"

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

Flask 웹서버 구축하기

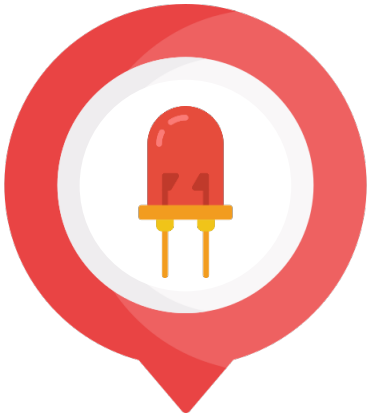
RASPBERRYPI





라우팅

네트워크 계층에서 목적지에 도달할
최적의 경로를 찾아 전송하는 것



@app.route(“/led/ON”)

Flask – 라우팅

RASPBERRYPI

Thonny - /home/pi/webapps/ch10/test.py @ 10:20

New Load Save Run Debug Over Into Out Stop Zoom Quit

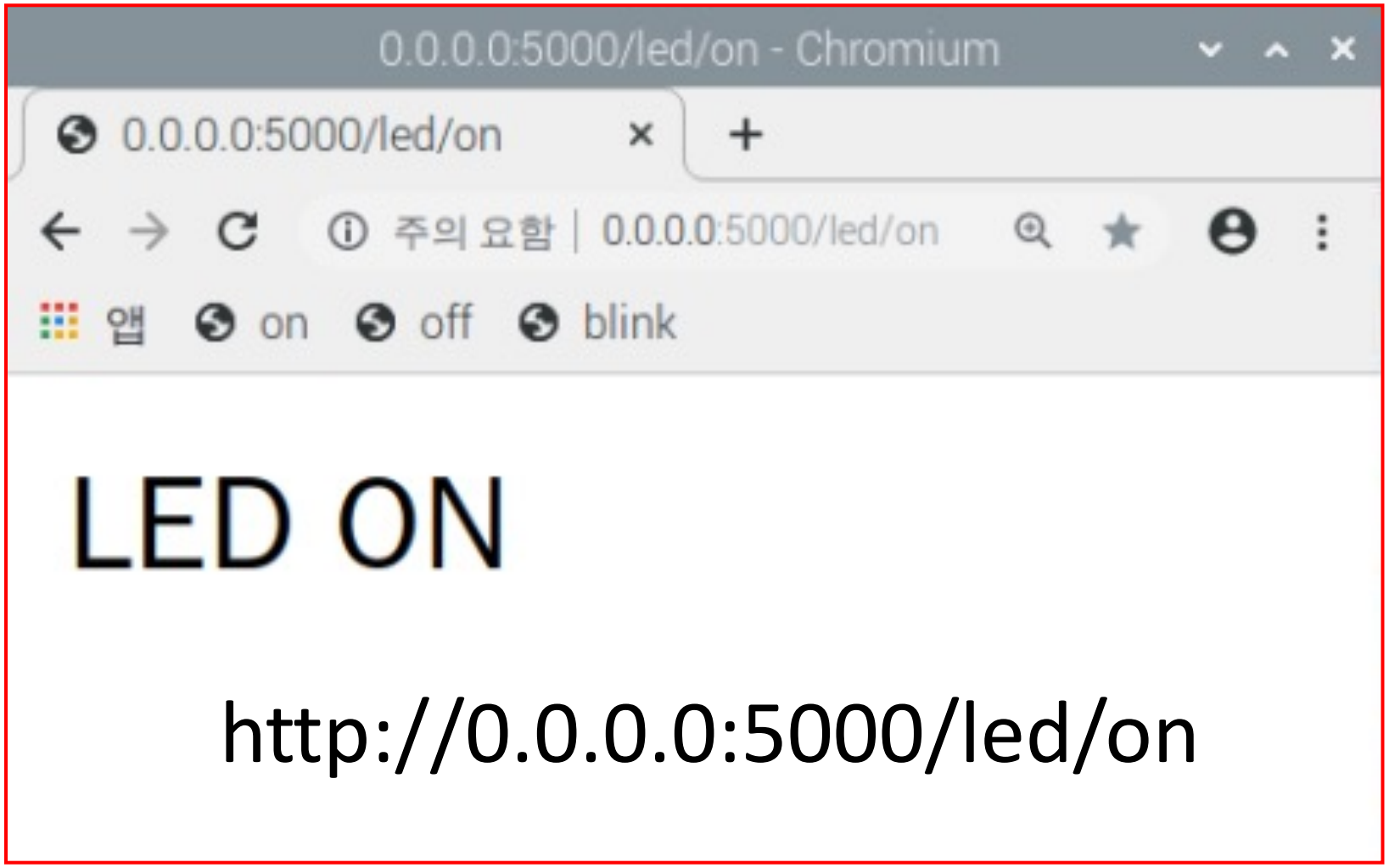
Switch to regular mode

test.py

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route("/")
5  def hello():
6      return "Hello World!"
7
8  @app.route("/led/on")
9  def led_on():
10     return "LED ON"
11
12 @app.route("/led/off")
13 def led_off():
14     return "LED OFF"
15
16 if __name__ == "__main__":
17     app.run(host="0.0.0.0")
```

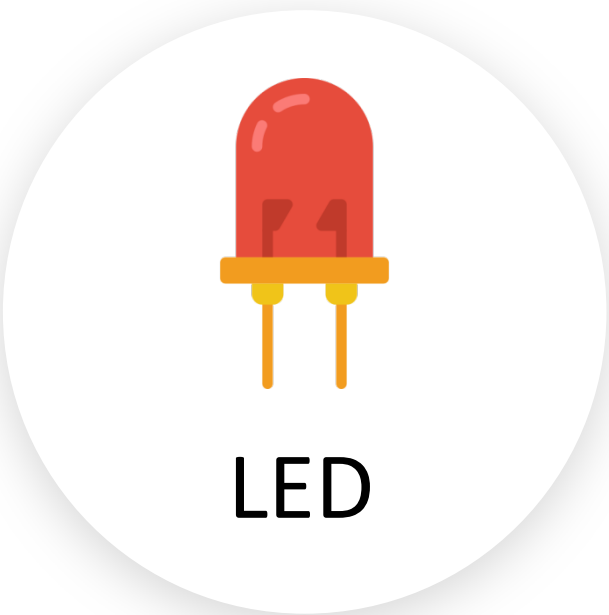
Shell

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run test.py

  * Serving Flask app "test" (lazy loading)
  * Environment: production
    WARNING: Do not use the development server in a production environment.
    Use a production WSGI server instead.
  * Debug mode: off
  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```



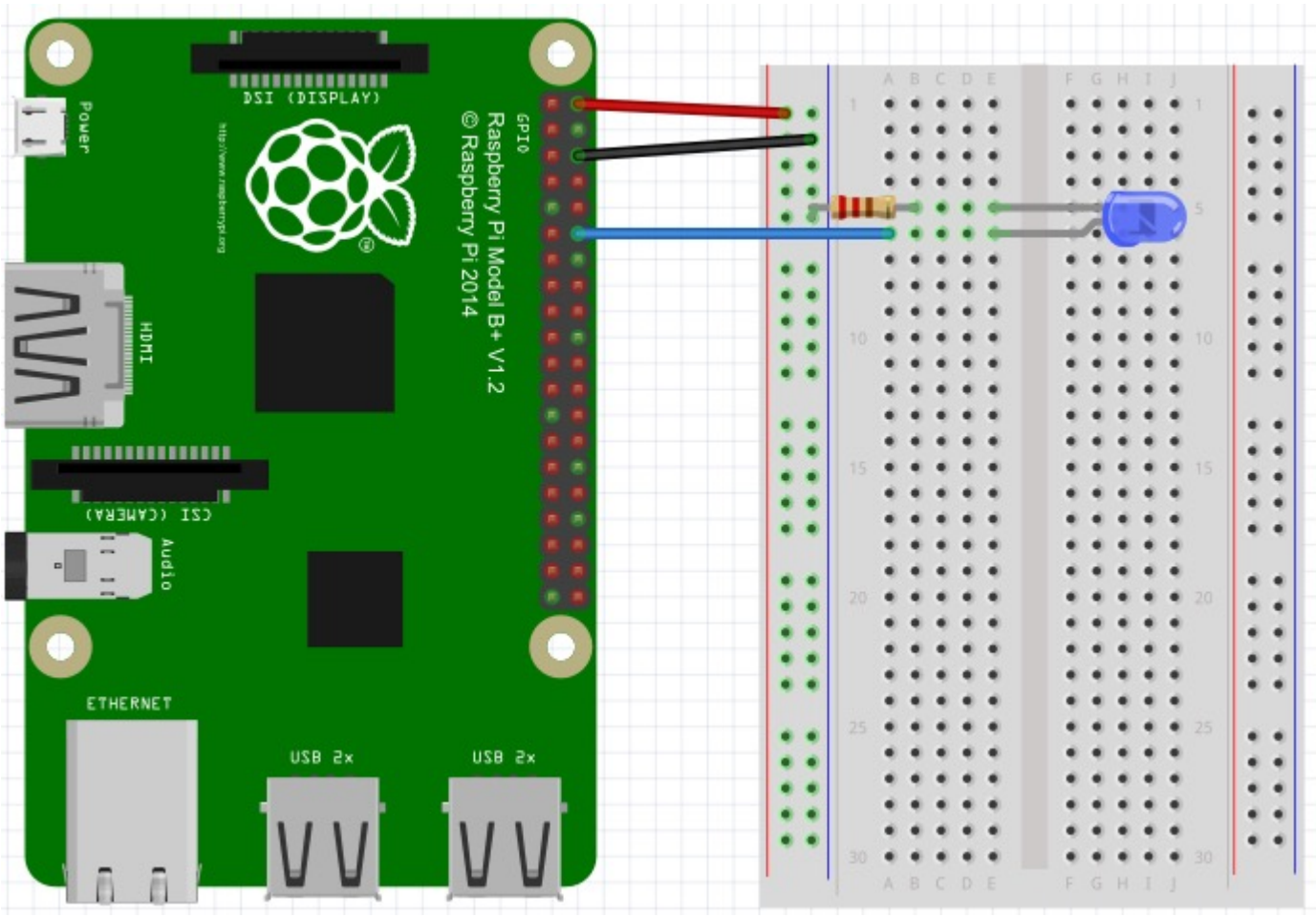
http://0.0.0.0:5000/led/on

Flask – LED 제어



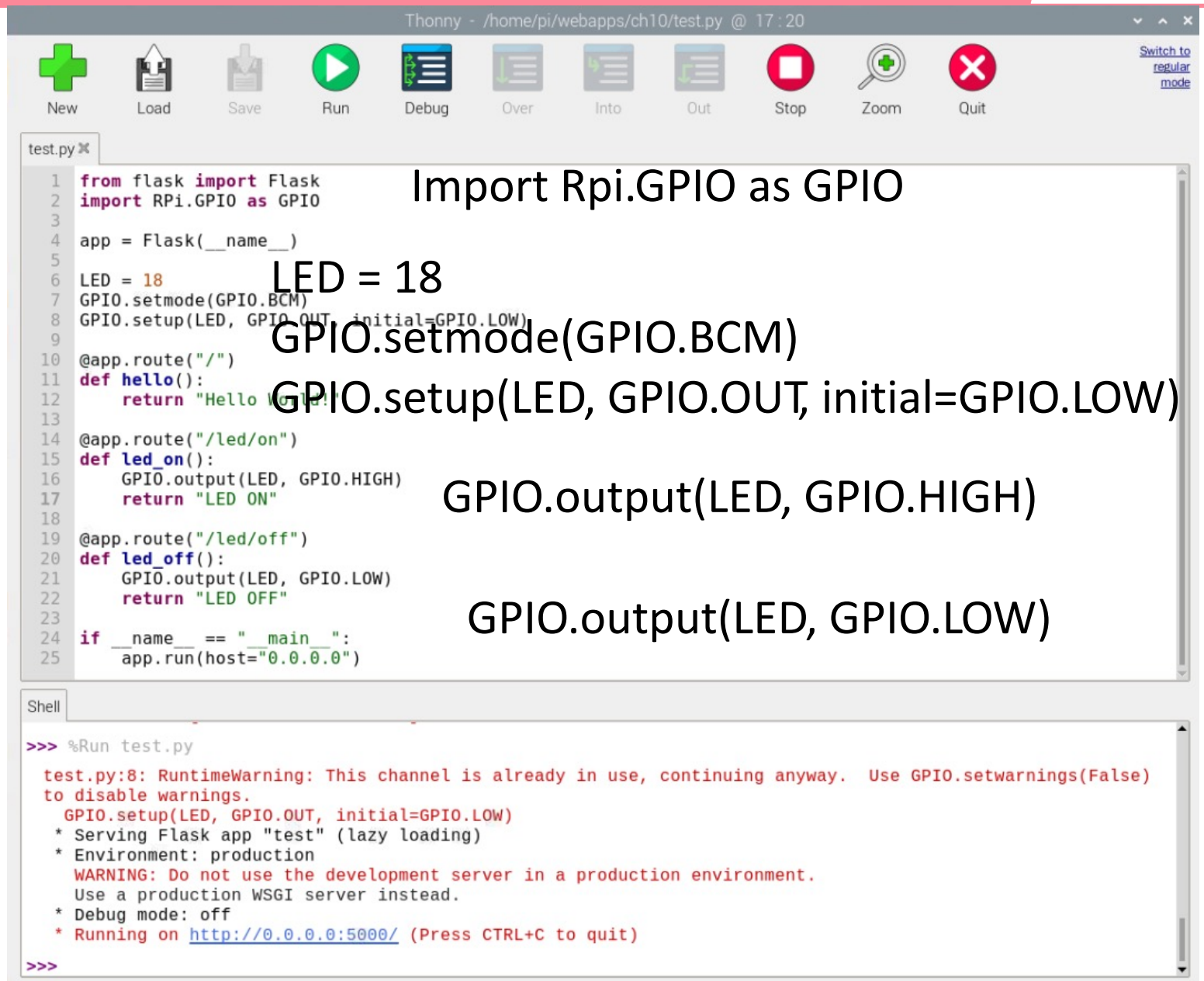
Flask로 웹어플리케이션 만들기

RASPBERRYPI



Flask – LED 제어

RASPBERRYPI



Import Rpi.GPIO as GPIO

LED = 18

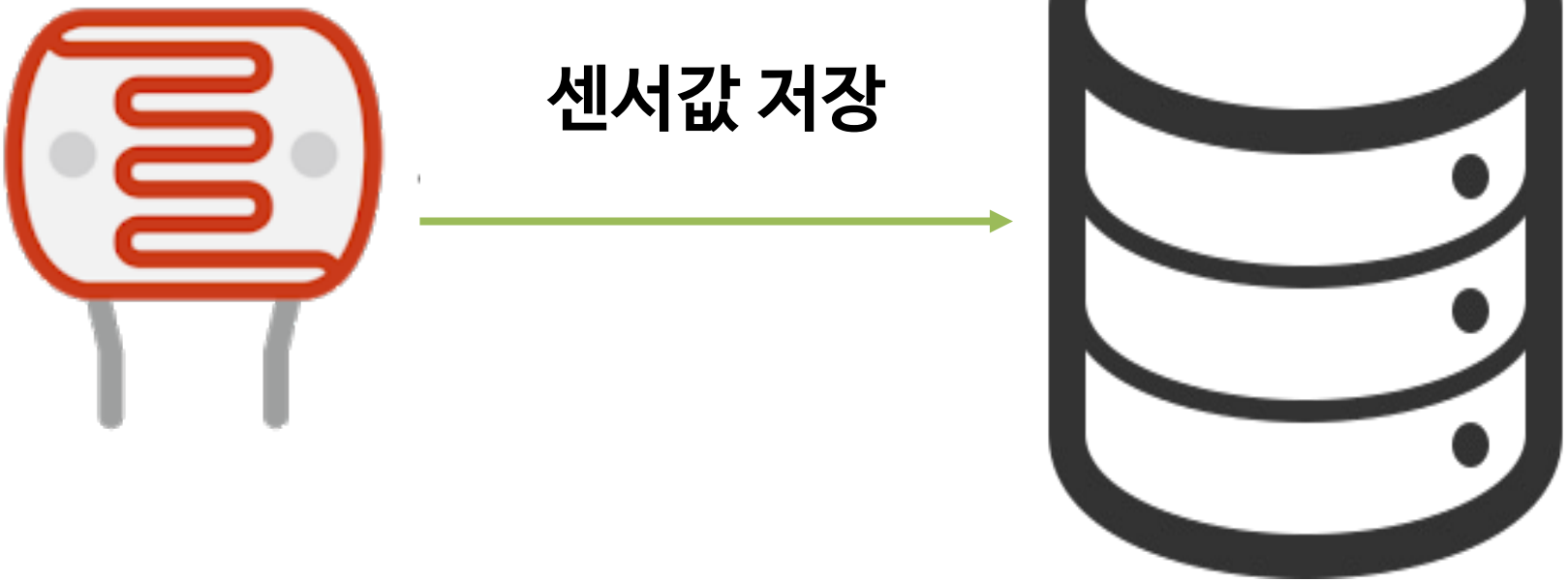
GPIO.setmode(GPIO.BCM)

GPIO.setup(LED, GPIO.OUT, initial=GPIO.LOW)

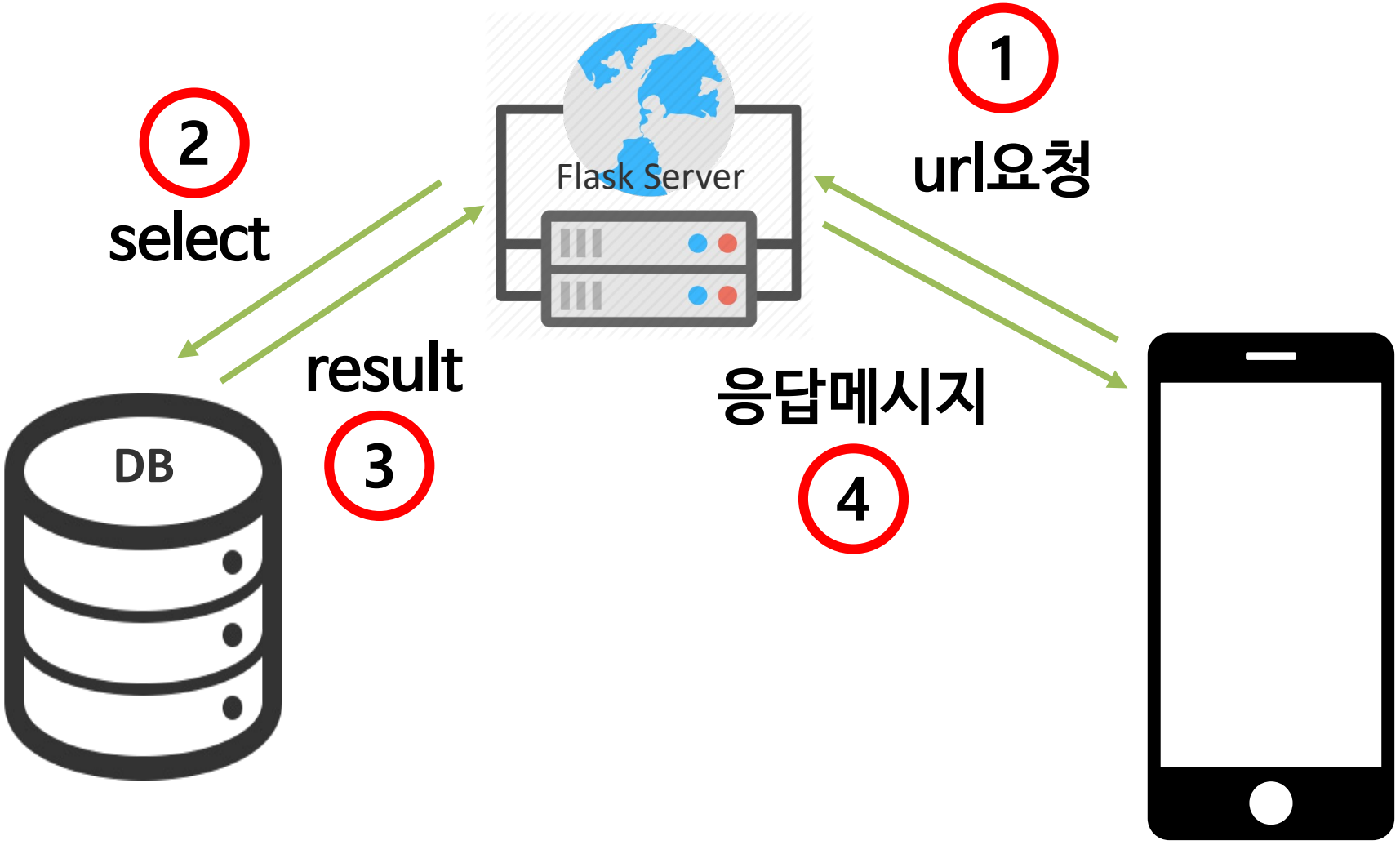
GPIO.output(LED, GPIO.HIGH)

GPIO.output(LED, GPIO.LOW)

Smart Media
스마트미디어인재개발원
Smart Media Human Resource Development


DB
센서값 저장

Multiproess Code



1. pip를 이용 MySQL 설치

$ sudo apt-get install mariadb-server

2. MySQL 접속

$ sudo mysql –u root

DB연동 – DB, Table생성

1. 데이터 베이스 생성

```
CREATE DATABASE Test;
```

2. 데이터 베이스 선택

```
USE Test;
```

3. 테이블 생성

```
CREATE TABLE sensordb (
        sensing INT,
        ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

1. pip를 이용 PyMySQL설치

$ sudo pip install PyMySQL

$ sudo apt install python3-pymysql

3. 테이블 생성

```
CREATE TABLE test (
        sensing INT,
        ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

PyMySQL 설치

- PyMySQL 사용 : import pymysql as ps

```
import pymysql as ps
```

- MySQL연결 : ps.connect( )

```
db = ps.connect(
    host="localhost",
    user="root",
    passwd="",
    db="Test",
    charset="utf8")
```

root권한 에러 발생

```
pymysql.err.InternalError: (1698, "Access denied for user 'root'@
'localhost'")
```

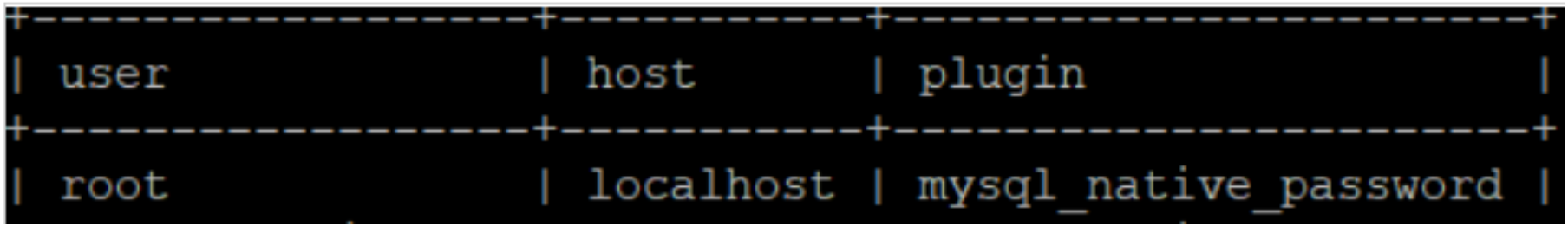
root 비밀번호 권한 변경

$ sudo mysql -u root -p
mysql > USE mysql
mysql > SELECT User, Host, plugin FROM mysql.user;

```
+------------------+-----------------------+
| User             | plugin                |
+------------------+-----------------------+
| root             | auth_socket           |
```

PyMySQL 설치


```
$ sudo mysql -u root -p
mysql > USE mysql
mysql > UPDATE user SET plugin='mysql_native_password'
        WHERE User='root';
mysql > FLUSH PRIVILEGES;
mysql > exit;
```



user	host	plugin
root	localhost	mysql_native_password

DB insert Code

- 데이터 삽입(INSERT)
  1. Cursor Object 가져오기 : cursor = db.cursor()
  2. SQL 실행하기 : cursor.execute(SQL)
  3. 실행 mysql 서버에 확정 반영하기 : db.commit()
  4. DB연결 닫기 : db.close( )

```
insert_sql = "insert into memberinfo (id, pw, name, age) values('a','456456','에이',22)"
```

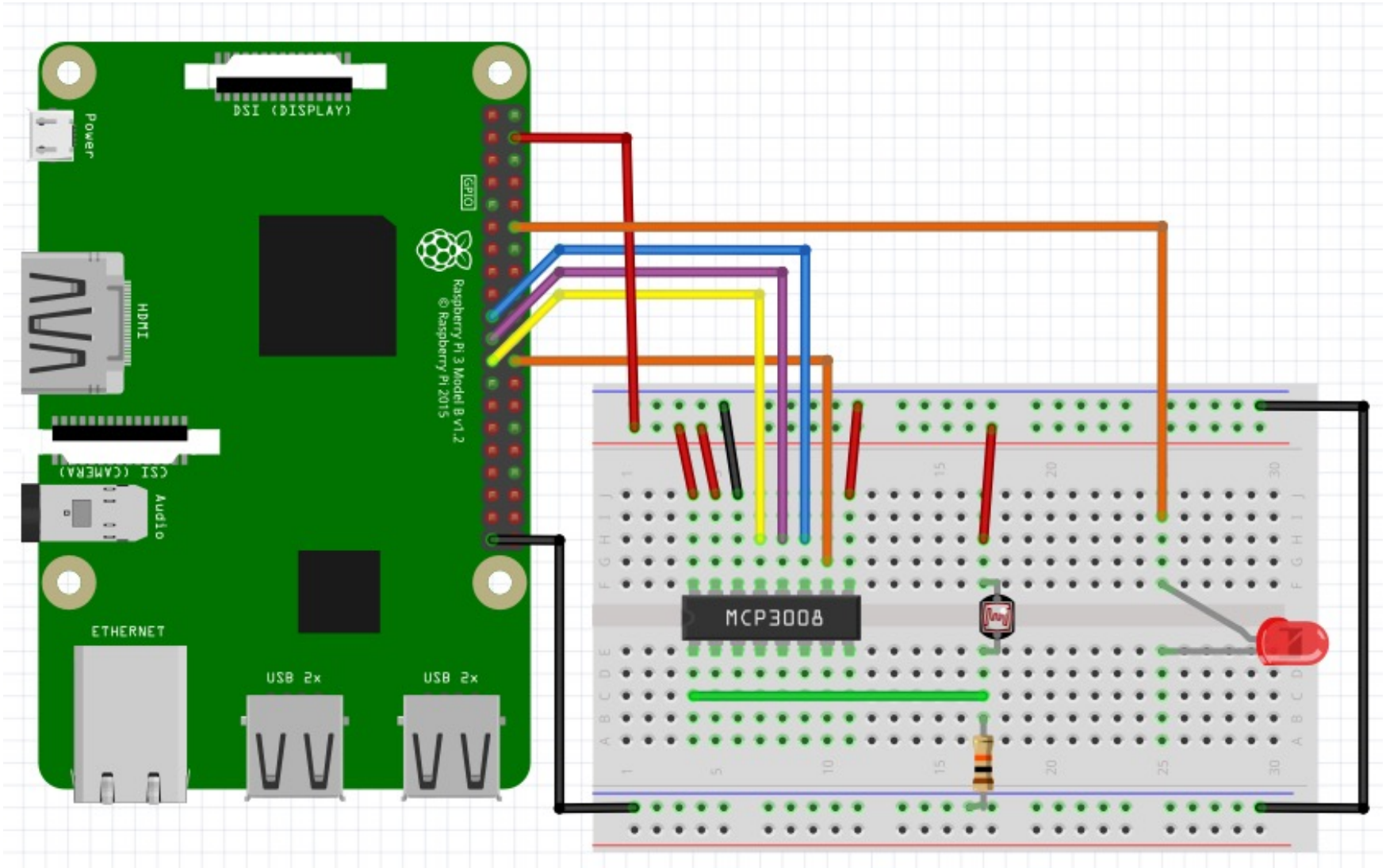
```
cursor.execute(insert_sql)
db.commit()
```

DB insert Code


```
curs = db.cursor()

sql = ‘insert into sensordb(sensing) values(1023)’

curs.execute(sql)

db.commit()
```

조도센서 회로도 / ADC

RASPBERRYPI



Analog Read Code


```
import spidevRead as sr

while True :

        readData = sr.analog_read(0)

        print(f'analogRead : {readData}')

        time.sleep(0.5)
```

Analog Read Data insert Code


```
while True :
        readData = sr.analog_read(0)
        print(f’analogRead : {readData}’)
        sql = f'insert into test(sensing) values({readData})'
        curs.execute(sql)
        db.commit()
        time.sleep(2)
```

Multiproess Code


```
from multiprocessing import Process
import time


def test_1(start, end):
        for i in range(start, end, 1) :
            print( i )
            time.sleep(1)
pro1 = Process(target = test_1, args=(1, 100))
pro1.start()
pro1.join()
```

- 데이터 조회(SELECT)
  1. Cursor Object 가져오기 : cursor = db.cursor()
  2. SQL 실행하기 : cursor.execute(SQL)
  3. mysql 서버로부터 데이터 가져오기 : fetch 메서드 사용
     - fetchall() : Fetch all the rows
     - fetchmany(size=None) : Fetch several rows
     - fetchone() : Fetch the next row

DB select Code

- 데이터 조회 : SELECT

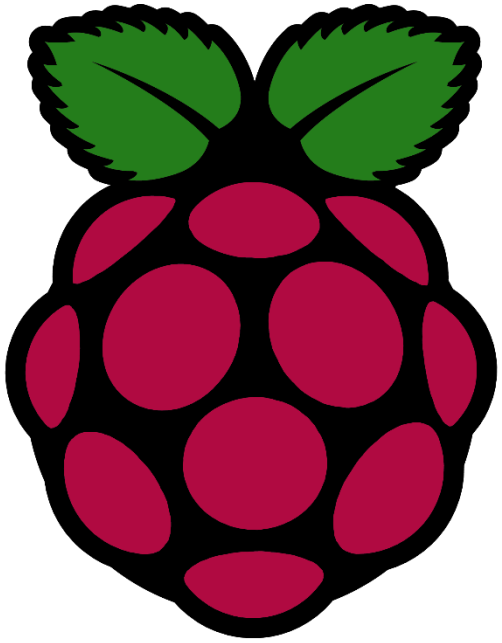
```
select_sql = "select * from memberinfo;"
cursor.execute(select_sql)
result = cursor.fetchall()
print(result)
```

Multiproess Code


```
def insertSensingData ():
        while True :
            readData = sr.analog_read(0)
            print(f'analogRead : {readData}')
            sql = f'insert into test(sensing) values({readData})'
            curs.execute(sql)
            db.commit()
            time.sleep(2)
pro1 = Process(target = insertSensingData)
pro1.start()
pro1.join()
```

Multiproess Code


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

@app.route("/sensor")
def getSensor():
        select_sql = 'select * from test;'
        curs.execute(select_sql)
        result= curs.fetchall()
        r=','.join(map(str,result))
        return r

if __name__  ==  " __main__ " :
        pro1 = Process(target=insertSensingData)
        pro1.start()
        app.run(host="192.168.137.131")
        pro1.join()
```



RasberryPi

Smart Media
스마트미디어인재개발원

스마트미디어인재개발원
나 예 호