

# IoT 특론

4차시

AI첨단기술학과

이의혁

## 2. 사물 인터넷 디바이스

### 2-3. IoT 디바이스 프로그래밍

1) IoT 디바이스 프로그래밍 준비

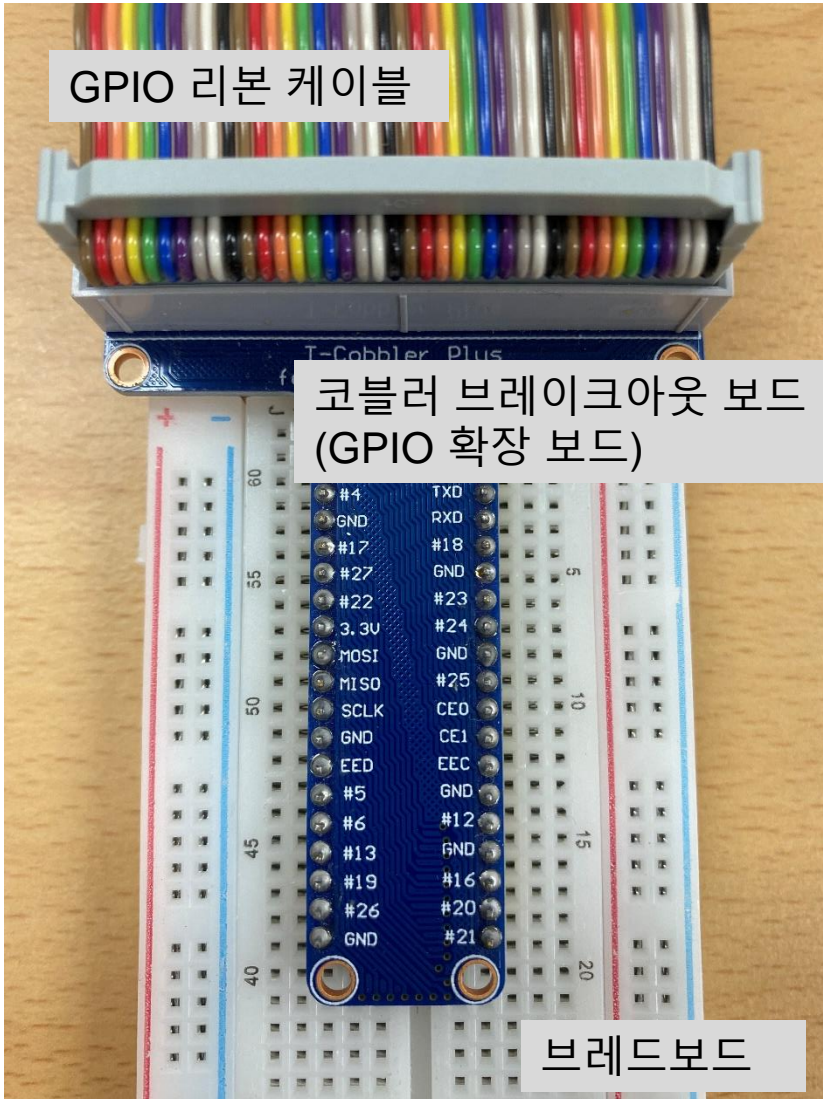
# IoT 디바이스 프로그래밍을 위한 준비

- 준비물
  - Raspberry Pi
    - Pi 3 Model B
  - 마이크로 SD 카드
    - Raspberry Pi OS 설치
  - 전원: 마이크로 USB (5V 1A 이상)
  - 키보드와 마우스
    - 초기 설정 시
  - HDMI 지원 모니터 및 HDMI 케이블
  - 케이스

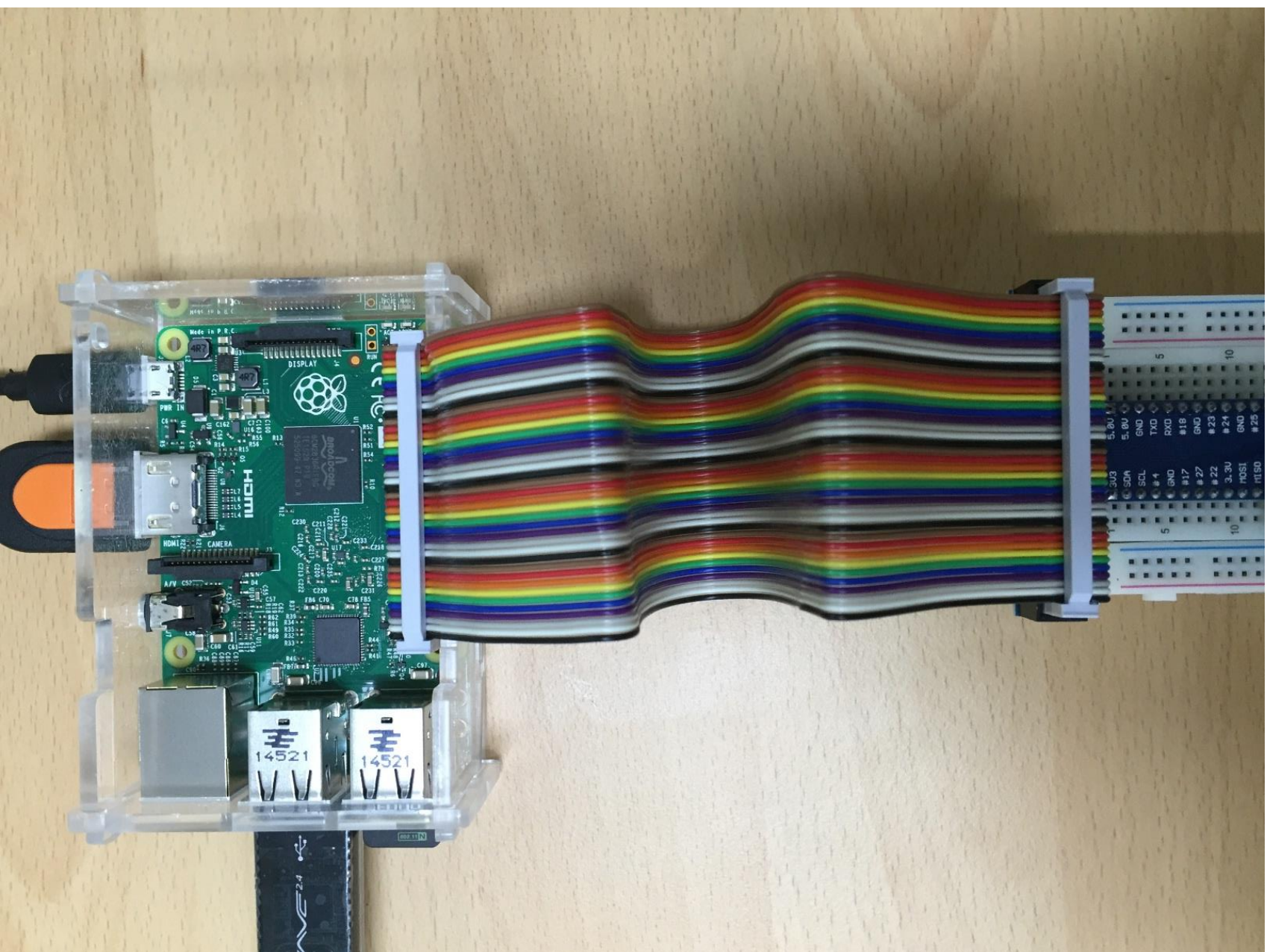


- 각종 센서와 액추에이터를 연결하고 사용하기 위한 부품
  - 브레드보드
    - 각종 센서, 액추에이터 장치들을 연결하는 회로를 구성하기 위한 용도
  - 코블러 브레이크아웃 보드(GPIO 확장 보드)
    - Raspberry Pi의 GPIO 핀을 이용하여 회로를 구성할 때 GPIO 핀을 좀 더 쉽게 사용할 수 있도록 해주는 커넥터 보드
  - 40핀 GPIO 리본 케이블
    - Raspberry Pi 본체의 GPIO 핀과 브레이크아웃 보드를 연결하는 용도로 사용
    - 리본 케이블에 연결된 브레이크아웃 보드는 브레드보드에 장착하여 원하는 회로를 구성
- 각종 센서, LED, 저항, 점퍼와이어(전선) 등

# Raspberry Pi와 브레드보드 연결







# Raspberry Pi OS 설치

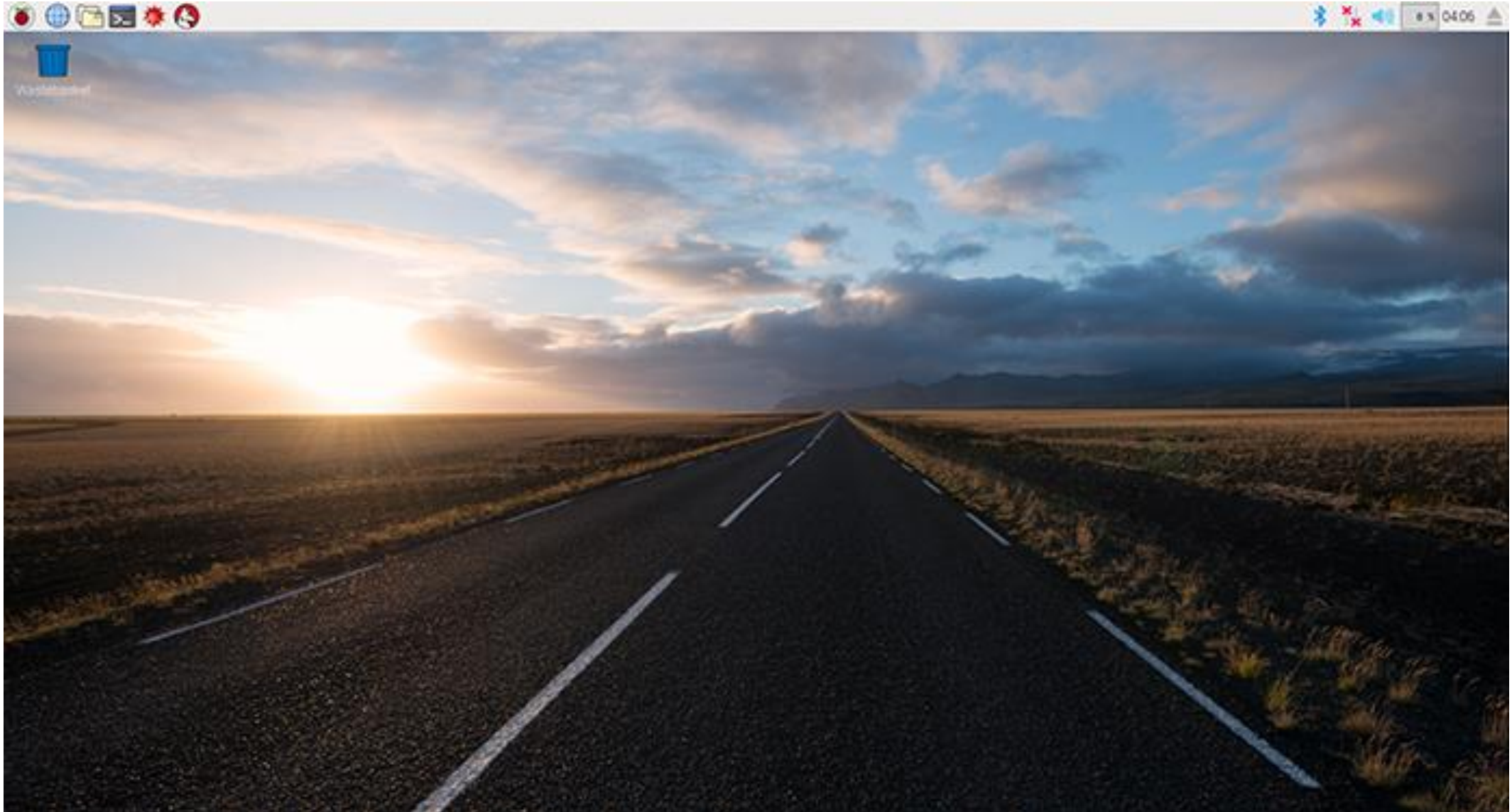
- Raspberry Pi Imager를 이용하여 설치
  - 라즈베리 파이 재단 홈페이지에서 다운로드 받을 수 있음
    - <https://www.raspberrypi.com/software/>



- SD 카드 리더기에 SD 카드를 넣고 이를 컴퓨터 USB에 꽂은 후 Raspberry Pi Imager 프로그램에서 Storage 선택 시 이 USB 드라이브를 선택



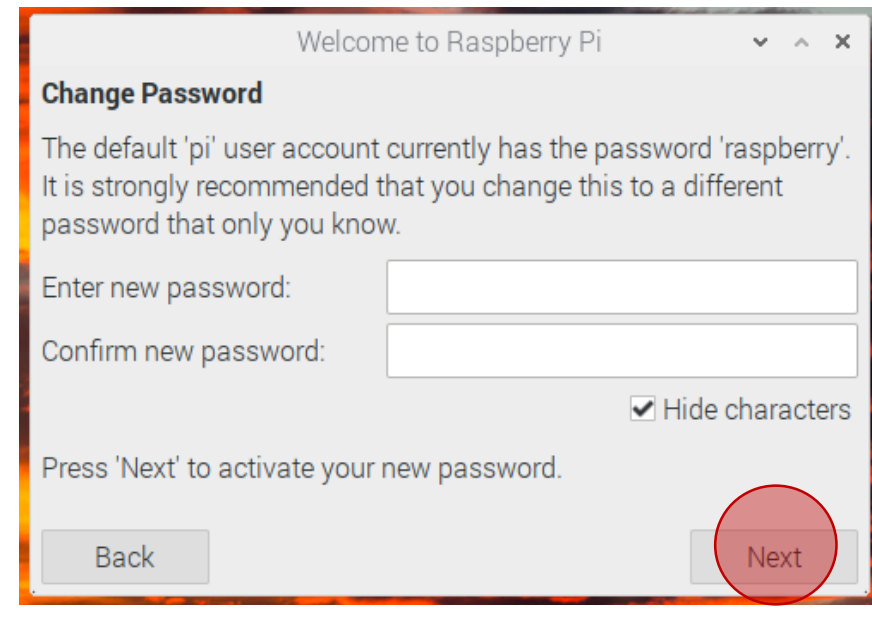
# Raspberry Pi OS 설치 후 데스크탑 모드로 실행된 화면



# 첫 부팅 후 초기 설정



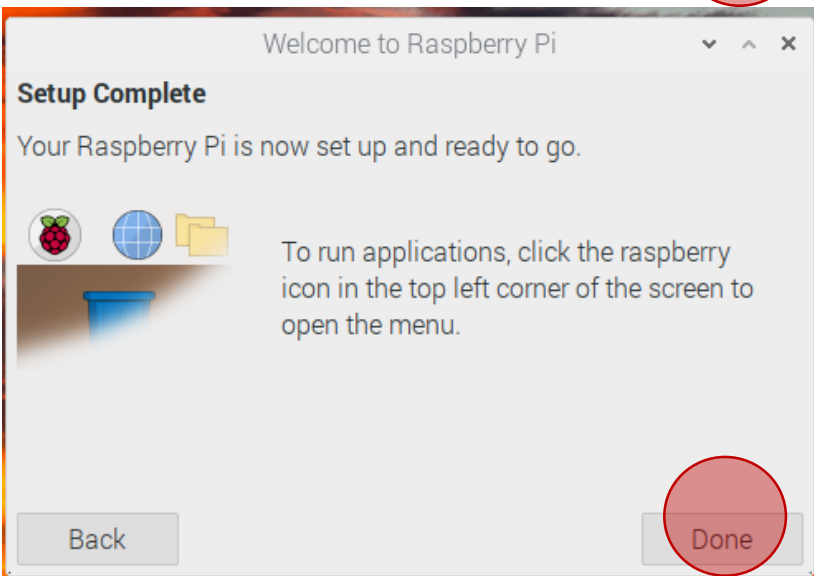
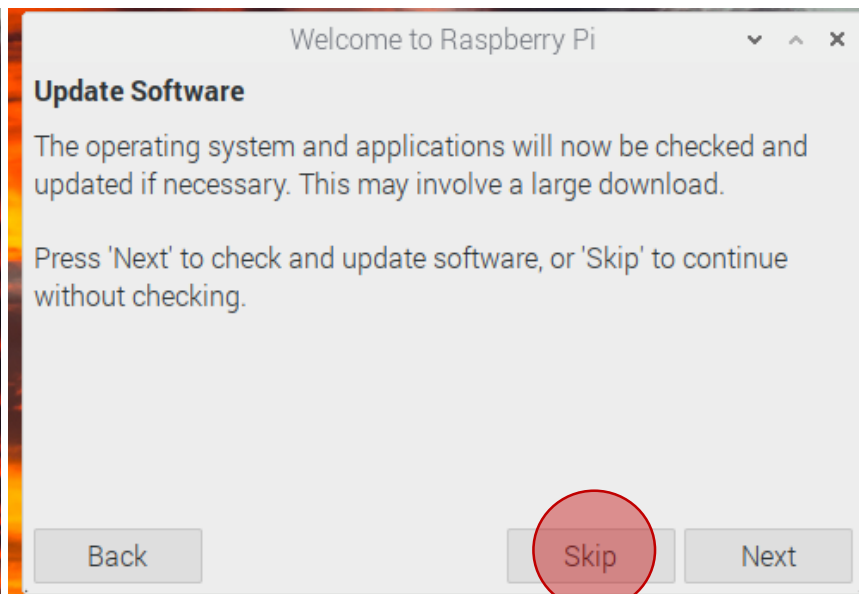
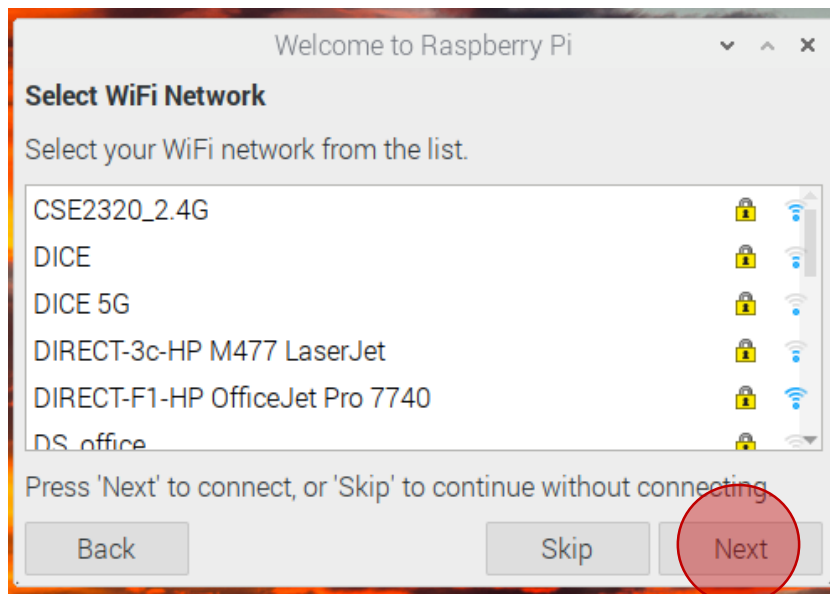
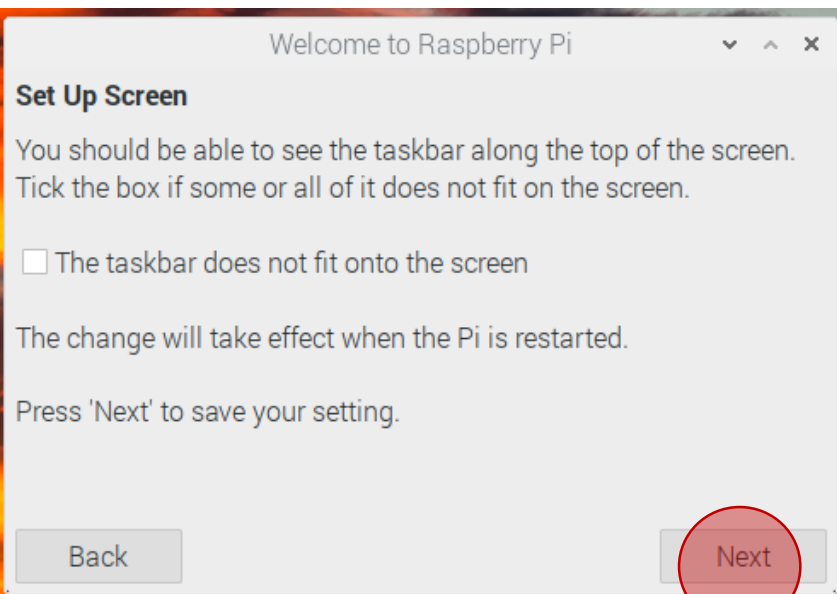
Raspberry Pi OS에서는 한글 사용을 위한 세팅이 안 되어 있기 때문에 English (US)로 먼저 진행



기본적으로 사용자 이름 pi, 패스워드 raspberry로 로그인할 수 있음

새 password를 설정할 수 있음

# 첫 부팅 후 초기 설정



WiFi 선택 후 패스워드 입력하면 연결이 이루어짐

소프트웨어 업데이트는 진행하지 않고 Skip 버튼을 눌러도 됨  
Next 버튼을 누르면 새로운 업데이트가 있는 경우 다운로드 후 설치 진행 될 것임 (시간 소요)

# 한글 설정 관련

```
pi@raspberrypi: ~  
파일(F) 편집(E) 탭(T) 도움말(H)  
pi@raspberrypi:~ $ sudo apt-get install ibus  
패키지 목록을 읽는 중입니다... 완료  
의존성 트리를 만드는 중입니다  
상태 정보를 읽는 중입니다... 완료  
ibus is already the newest version (1.5.19-4+deb10u1).  
0개 업그레이드, 0개 새로 설치, 0개 제거 및 86개 업그레이드 안 함.  
pi@raspberrypi:~ $ sudo apt-get install ibus-hangul  
패키지 목록을 읽는 중입니다... 완료  
의존성 트리를 만드는 중입니다  
상태 정보를 읽는 중입니다... 완료  
ibus-hangul is already the newest version (1.5.1-1+b5).  
0개 업그레이드, 0개 새로 설치, 0개 제거 및 86개 업그레이드 안 함.  
pi@raspberrypi:~ $ sudo apt-get install fonts-unfonts-core  
패키지 목록을 읽는 중입니다... 완료  
의존성 트리를 만드는 중입니다  
상태 정보를 읽는 중입니다... 완료  
fonts-unfonts-core is already the newest version (1:1.0.2-080608-16).  
0개 업그레이드, 0개 새로 설치, 0개 제거 및 86개 업그레이드 안 함.  
pi@raspberrypi:~ $
```

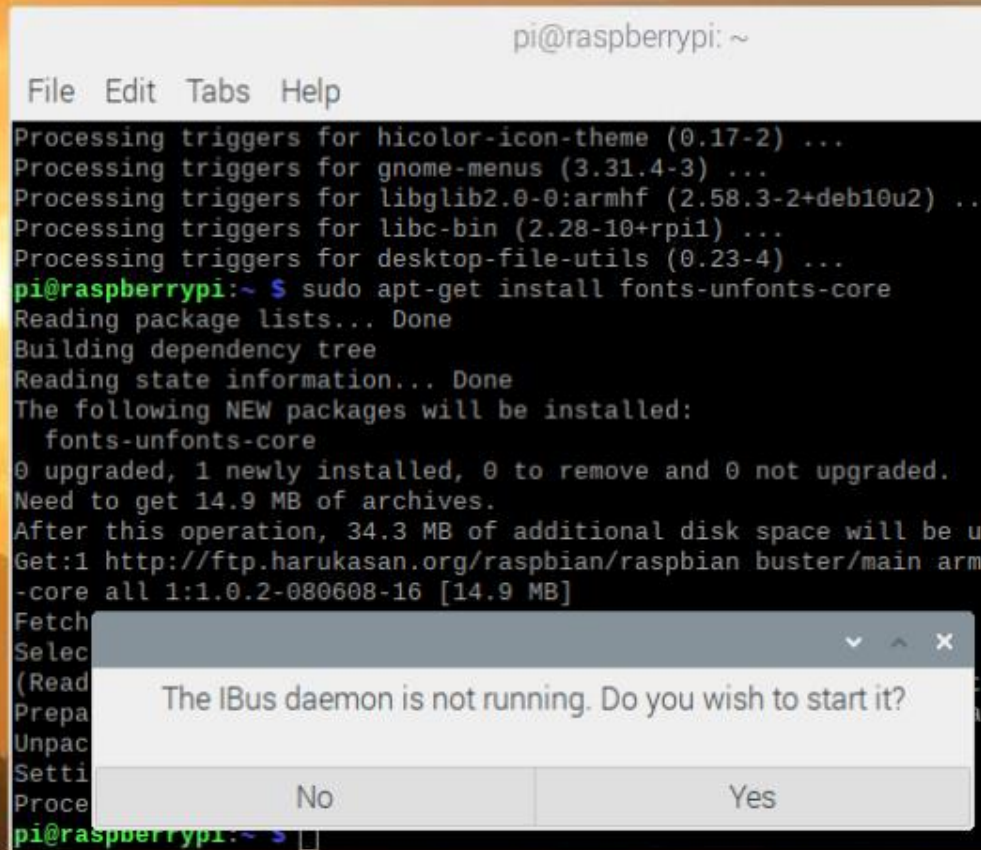
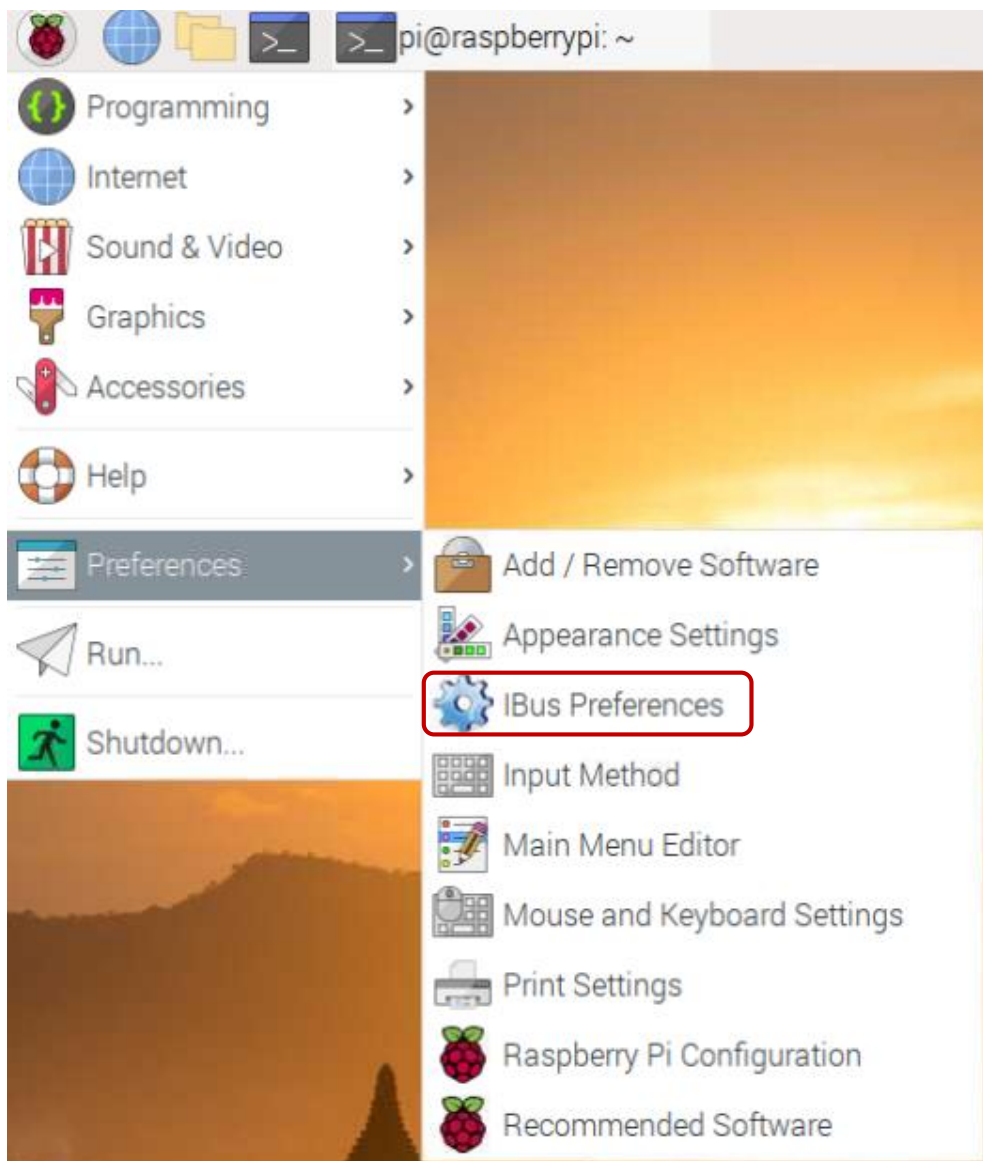
이미 설치되어 있는 상태에서 명령어를 실행했기 때문에 가장 최신 버전이 이미 설치되어 있다고 메시지가 나오고 끝난 것임

sudo apt-get install ibus 명령 실행 중 마지막에 Error가 발생했다고 나오면 sudo apt-get update 명령어 실행 후 다시 시도

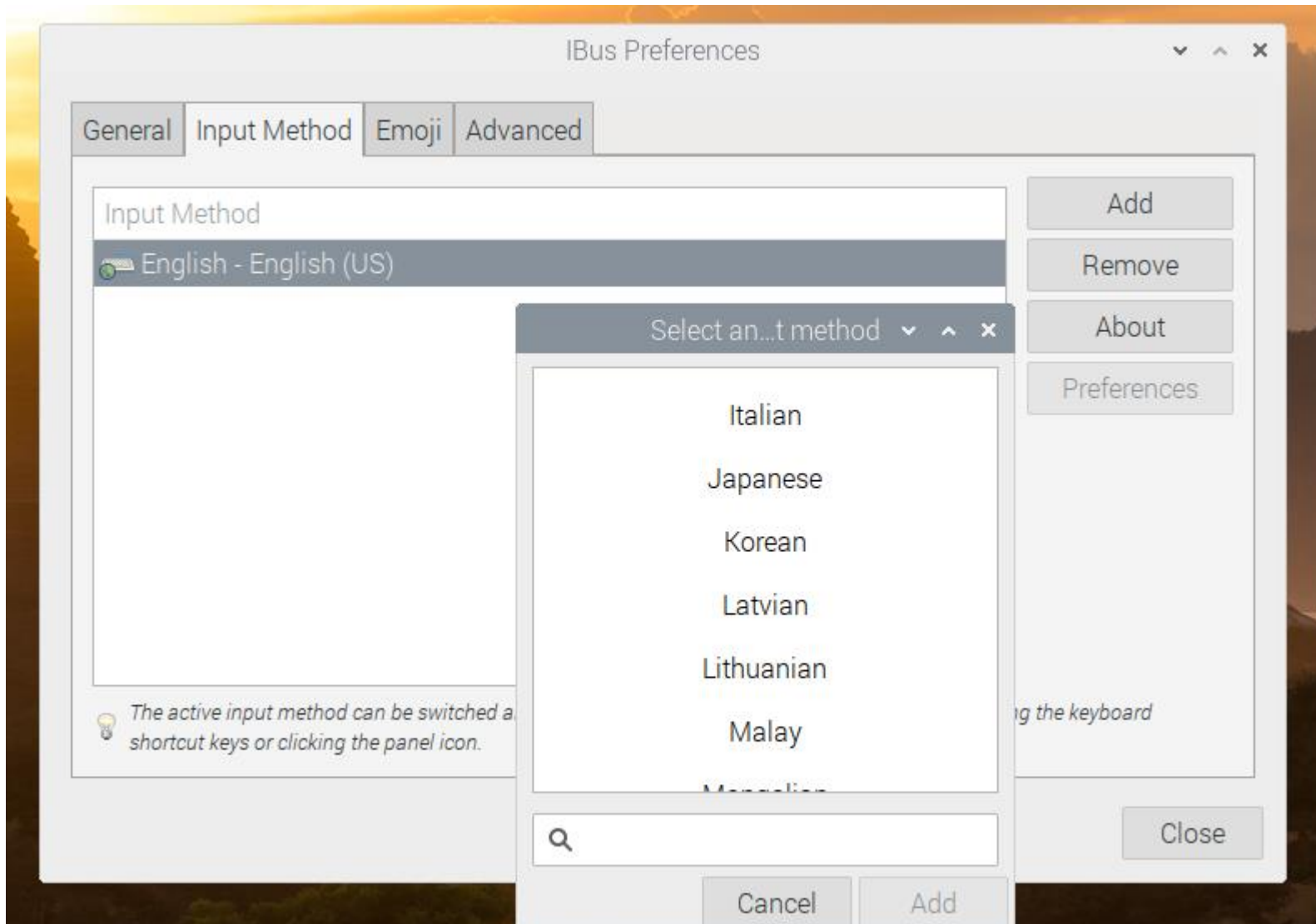


# 한글 설정 관련

메뉴 - 기본 설정(Preferences) > IBus 환경 설정(Preferences) 클릭  
후 IBus 데몬 실행 여부 묻는 메시지가 나오면 Yes(예) 선택



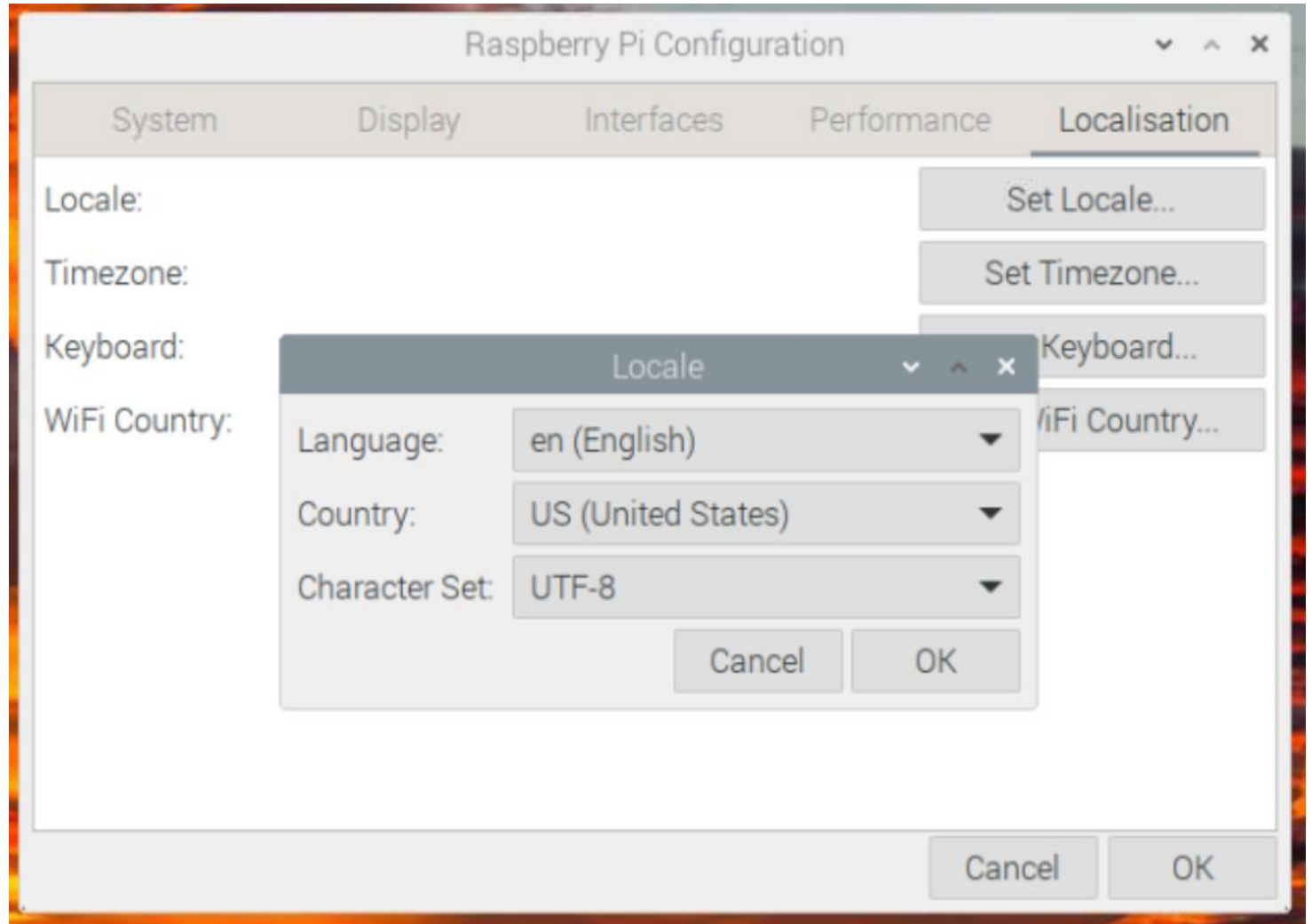
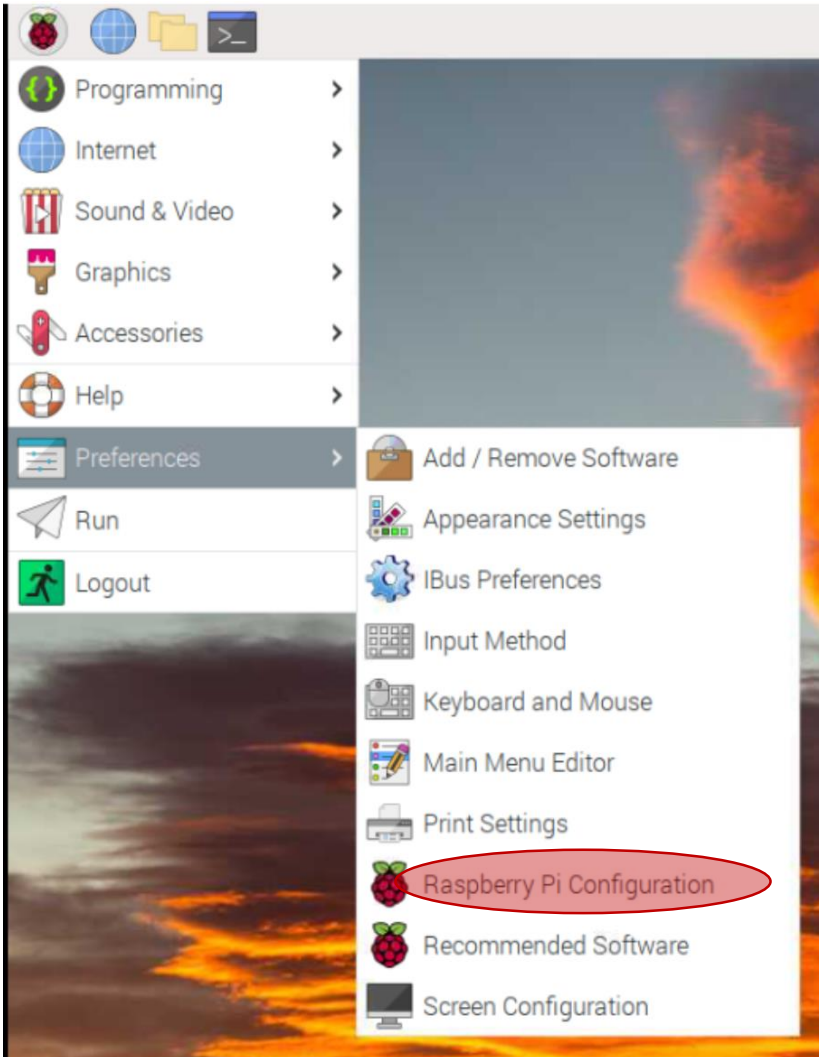
# 한글 설정 관련



Input Method 탭에서 Add 버튼을 누른 후 Korean 추가

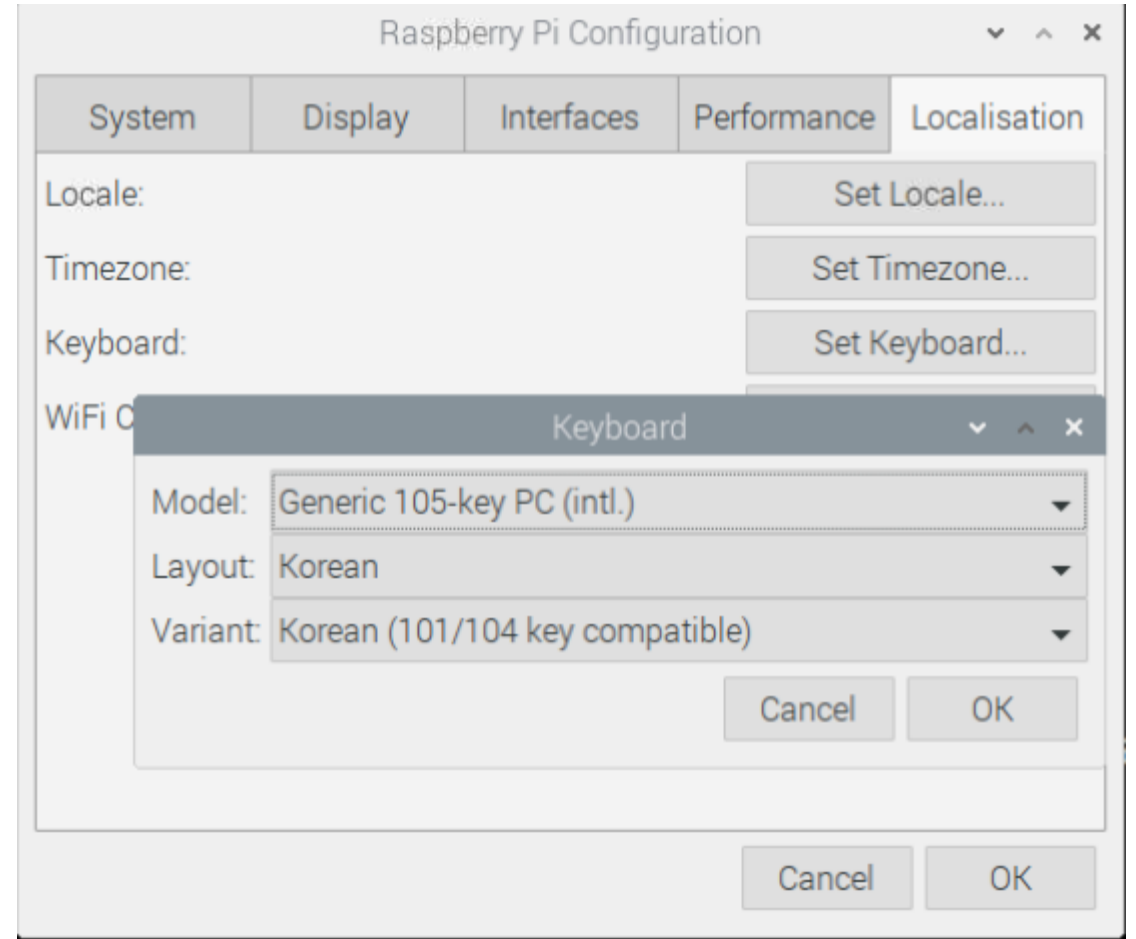
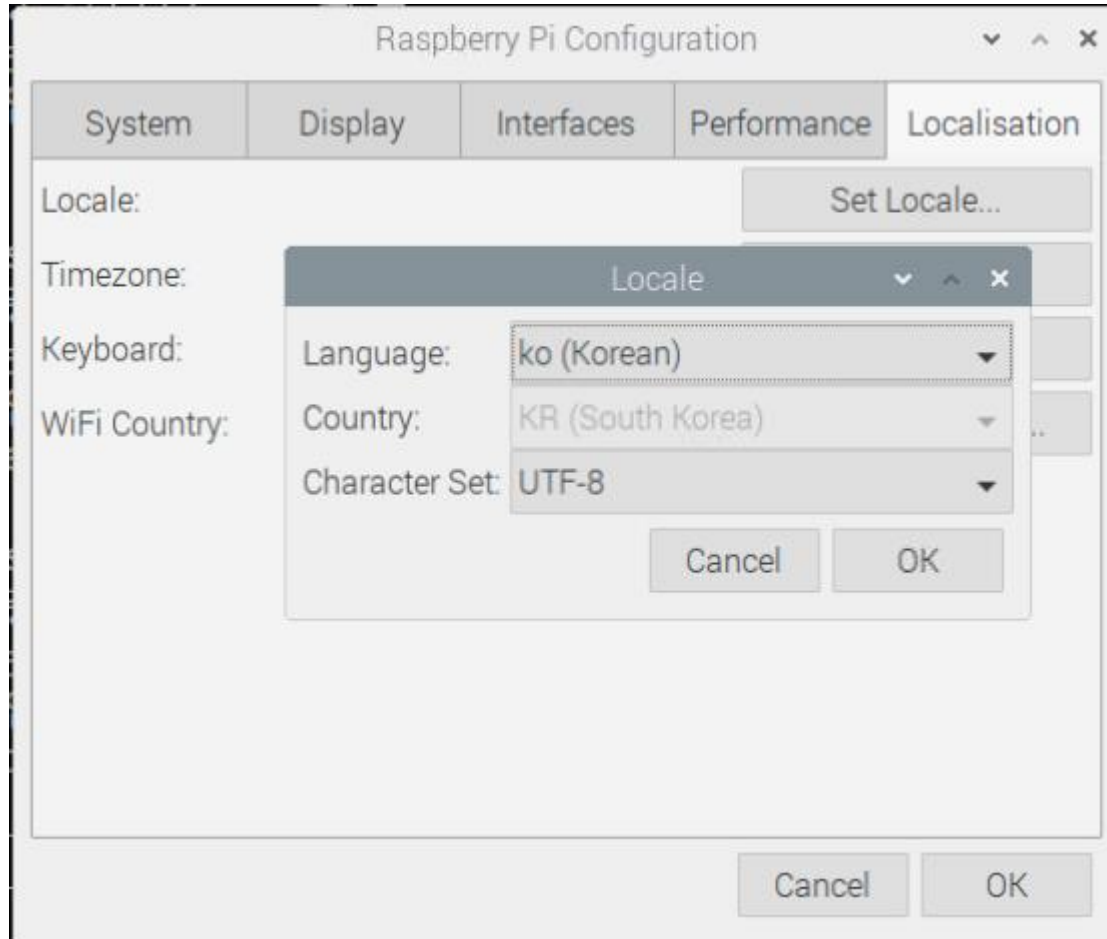


# Raspberry Pi Configuration



지역, 언어, 시간대, 키보드 설정 가능 (변경하게 되면 리부팅 함)

# 한글 설정 관련 (메뉴: 기본 설정(Preferences) > Raspberry Pi Configuration)

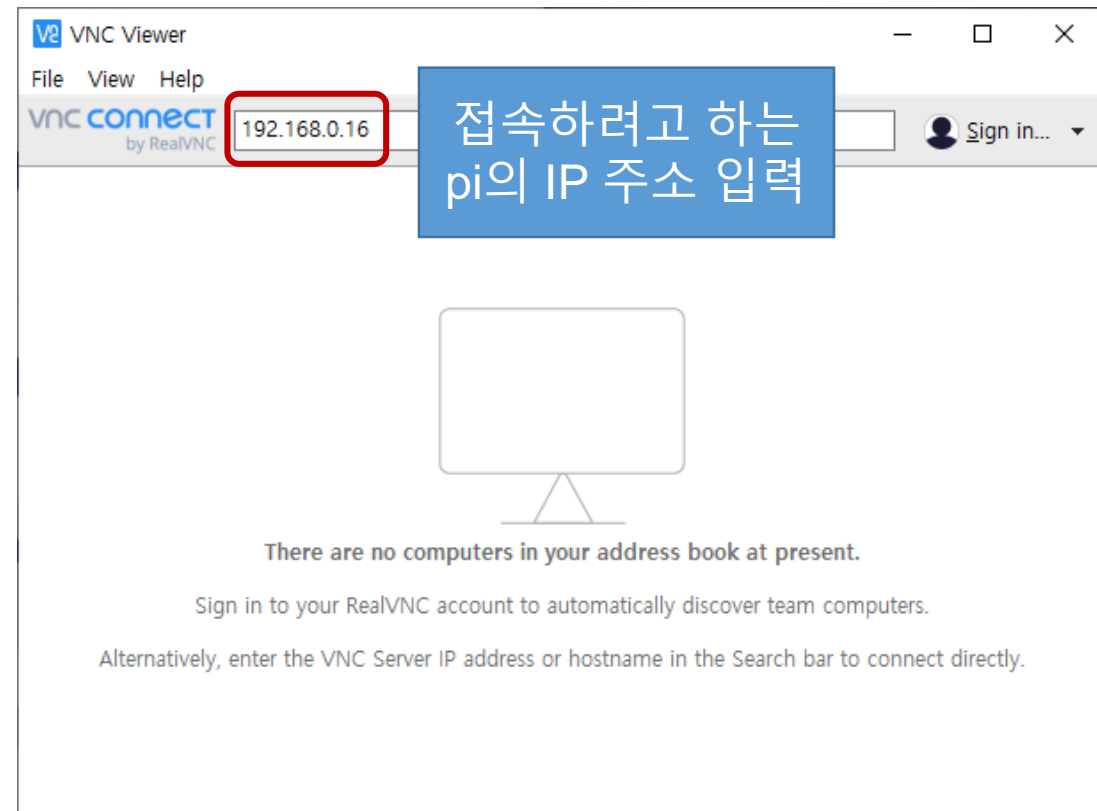


**Set WiFi Country는 변경하지 않고 그냥 둬**

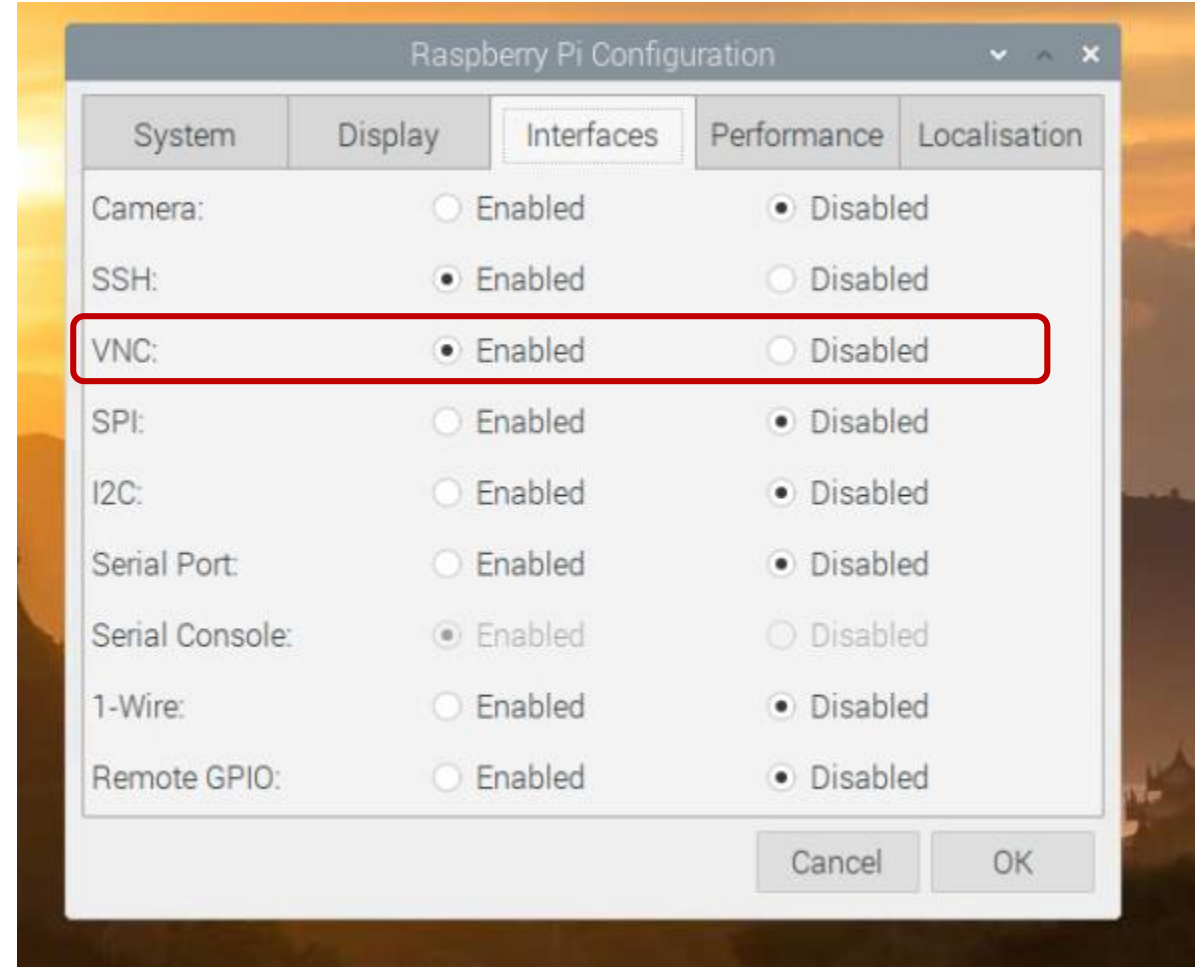
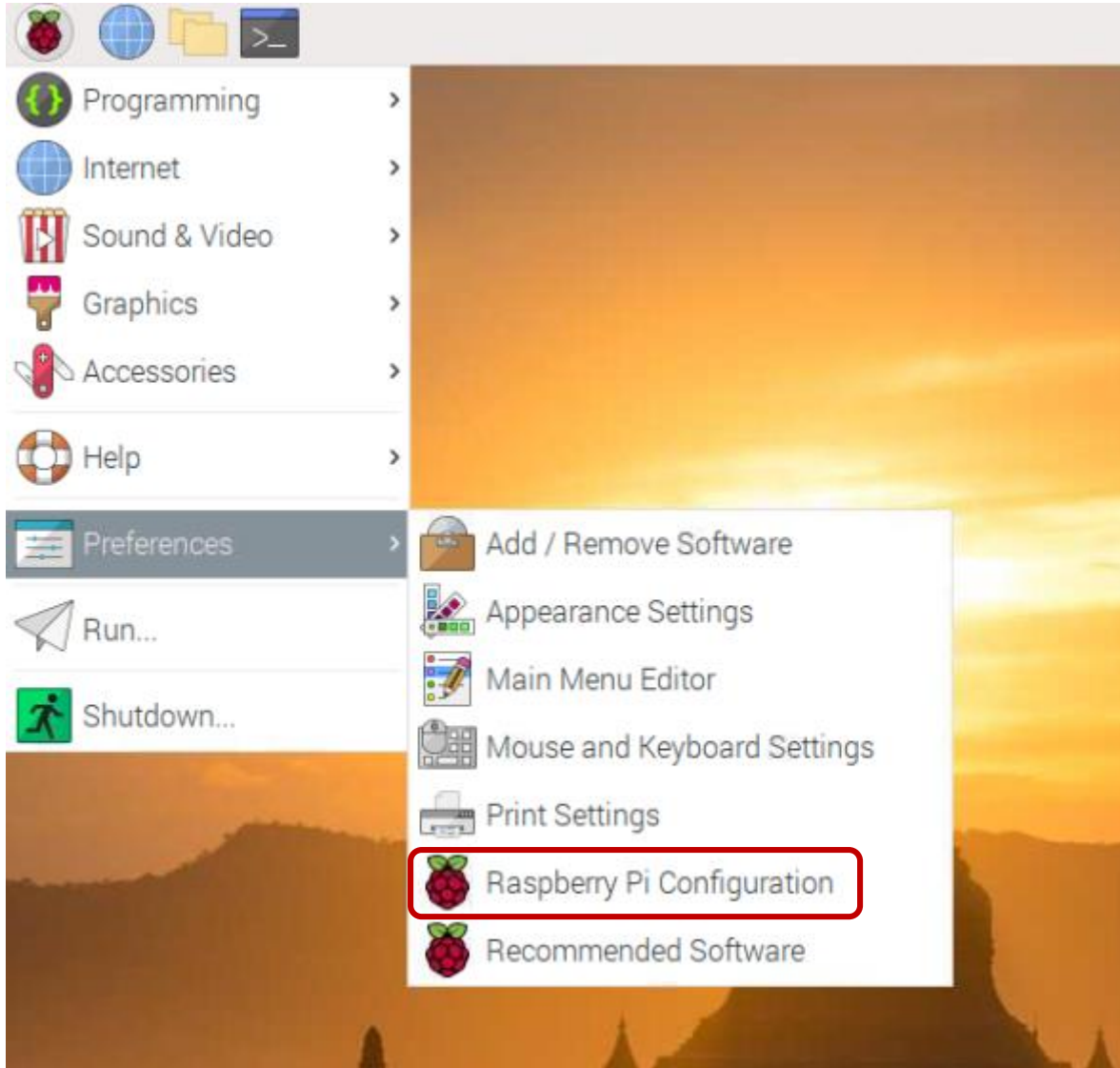
# VNC Viewer를 이용한 원격 접속

- VNC Server가 설치된 컴퓨터에 원격 접속하여 해당 컴퓨터의 화면을 원격으로 제어할 수 있음
  - <https://www.realvnc.com/en/connect/download/viewer/>
- Raspberry Pi에서 VNC 활성화 필요

원격 접속하려고 하는 컴퓨터와 Raspberry Pi가 같은 WiFi AP에 연결되어 있지 않으면 하기 어려움



# Raspberry Pi 원격 접속을 위한 VNC 활성화

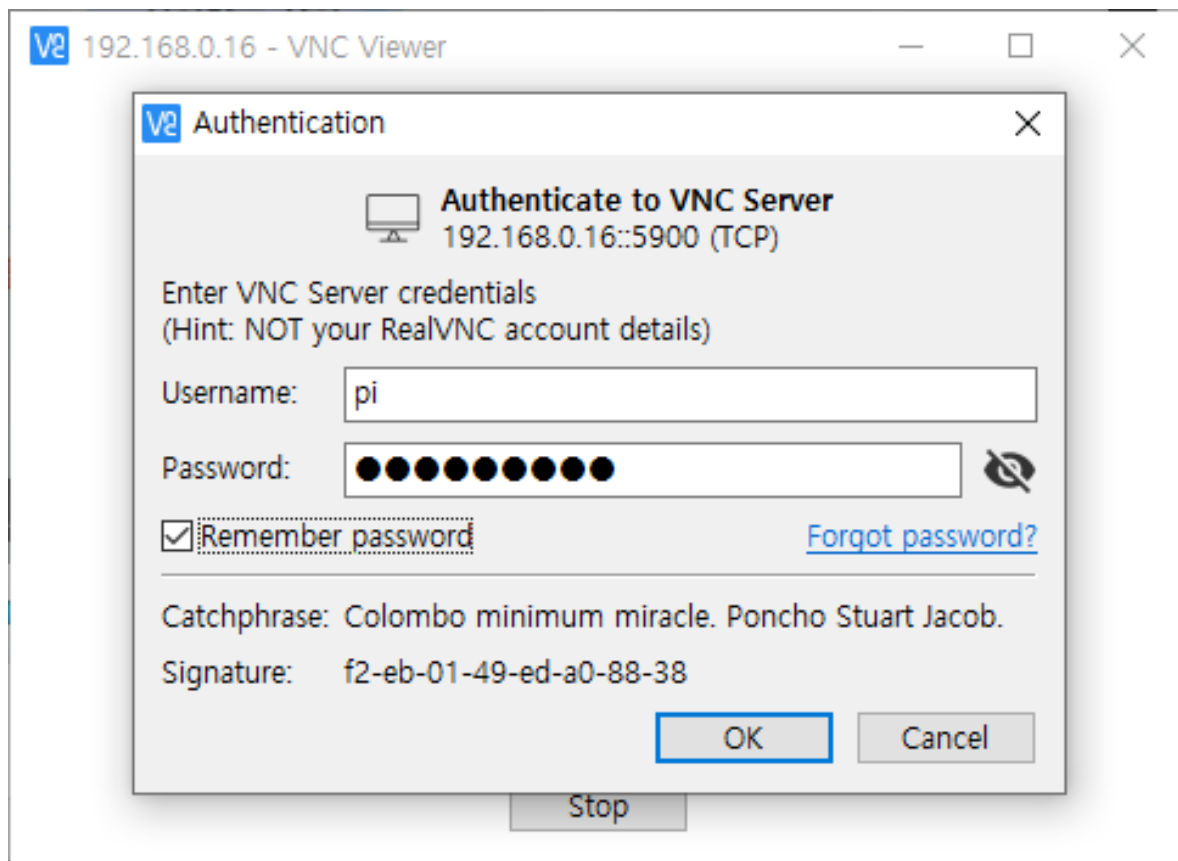
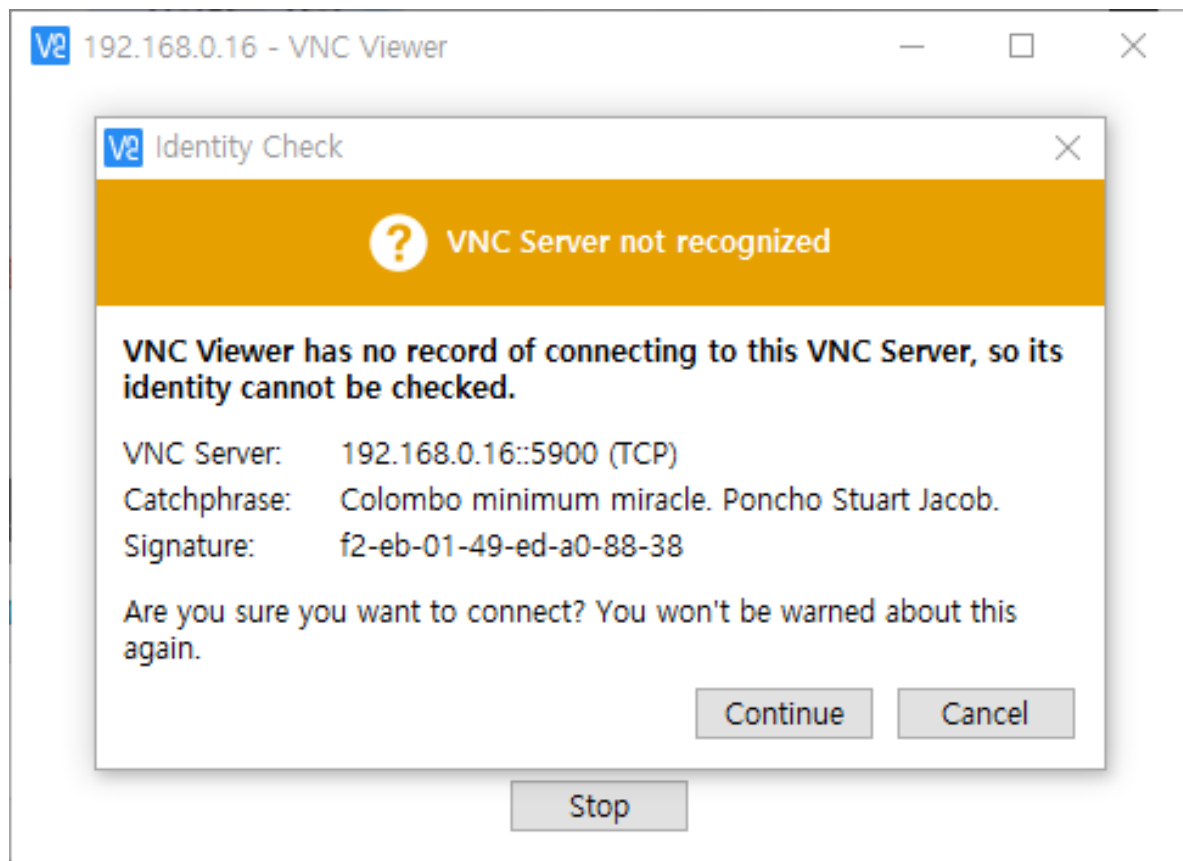


# Raspberry Pi IP 주소 확인

- 원격 접속을 위한 IP 주소 확인
  - ifconfig 명령어 실행
  - WiFi로 연결된 경우
    - wlan0의 inet addr 항목

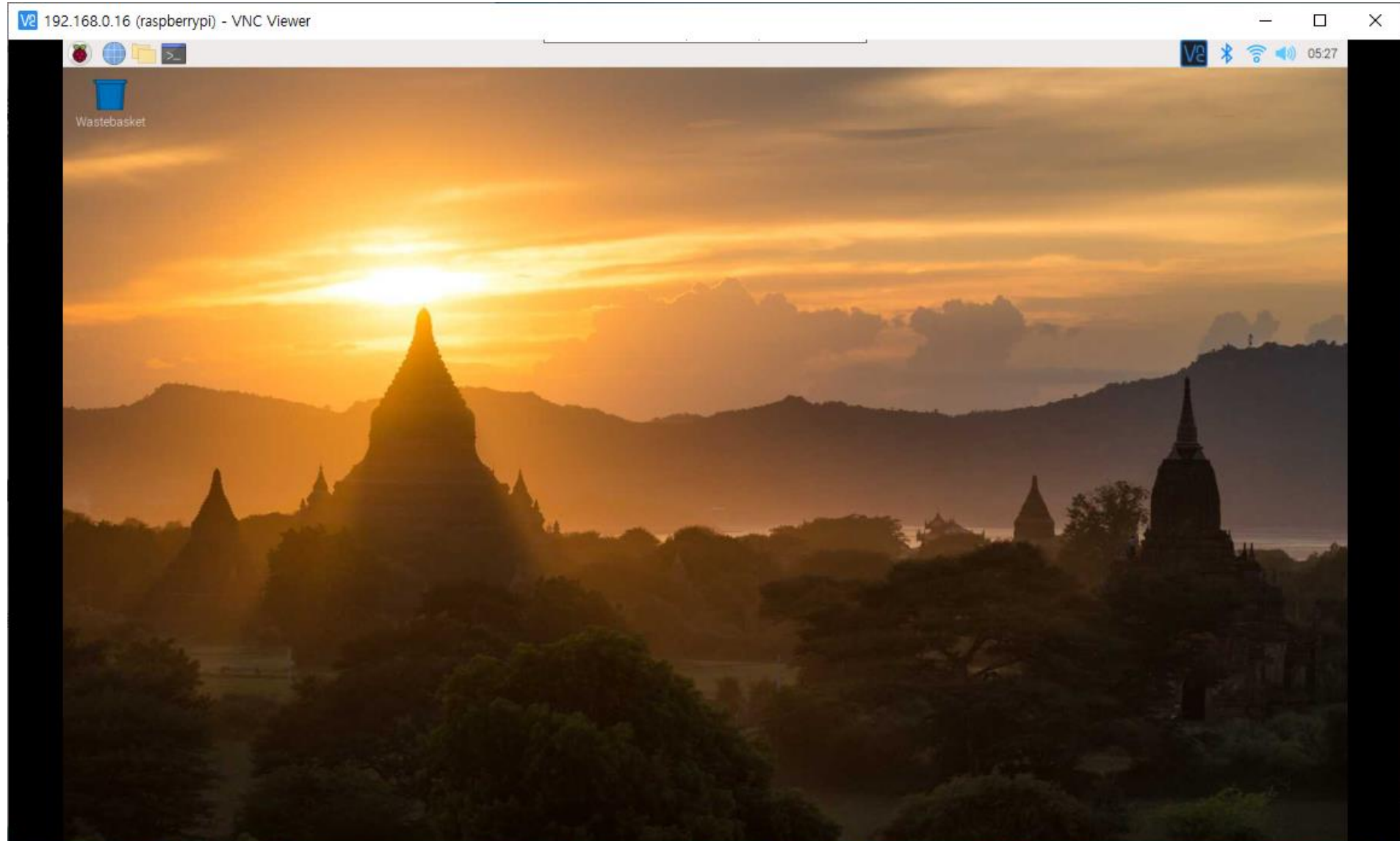
```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:8d:f6:3a  
          inet6 addr: fe80::ffac:18cc:6254:132a/64 Scope:Link  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:72 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:6288 (6.1 KiB)  TX bytes:6288 (6.1 KiB)  
  
wlan0     Link encap:Ethernet  HWaddr e8:4e:06:31:60:8b  
          inet addr:192.168.0.7  Bcast:192.168.0.255  Mask:255.255.255.0  
          inet6 addr: fe80::ea4e:6ff:fe31:608b/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:26824 errors:0 dropped:307 overruns:0 frame:0  
          TX packets:5967 errors:0 dropped:3 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:5473464 (5.2 MiB)  TX bytes:903603 (882.4 KiB)  
  
pi@raspberrypi ~ $
```

# VNC Viewer로 Pi 접속 과정





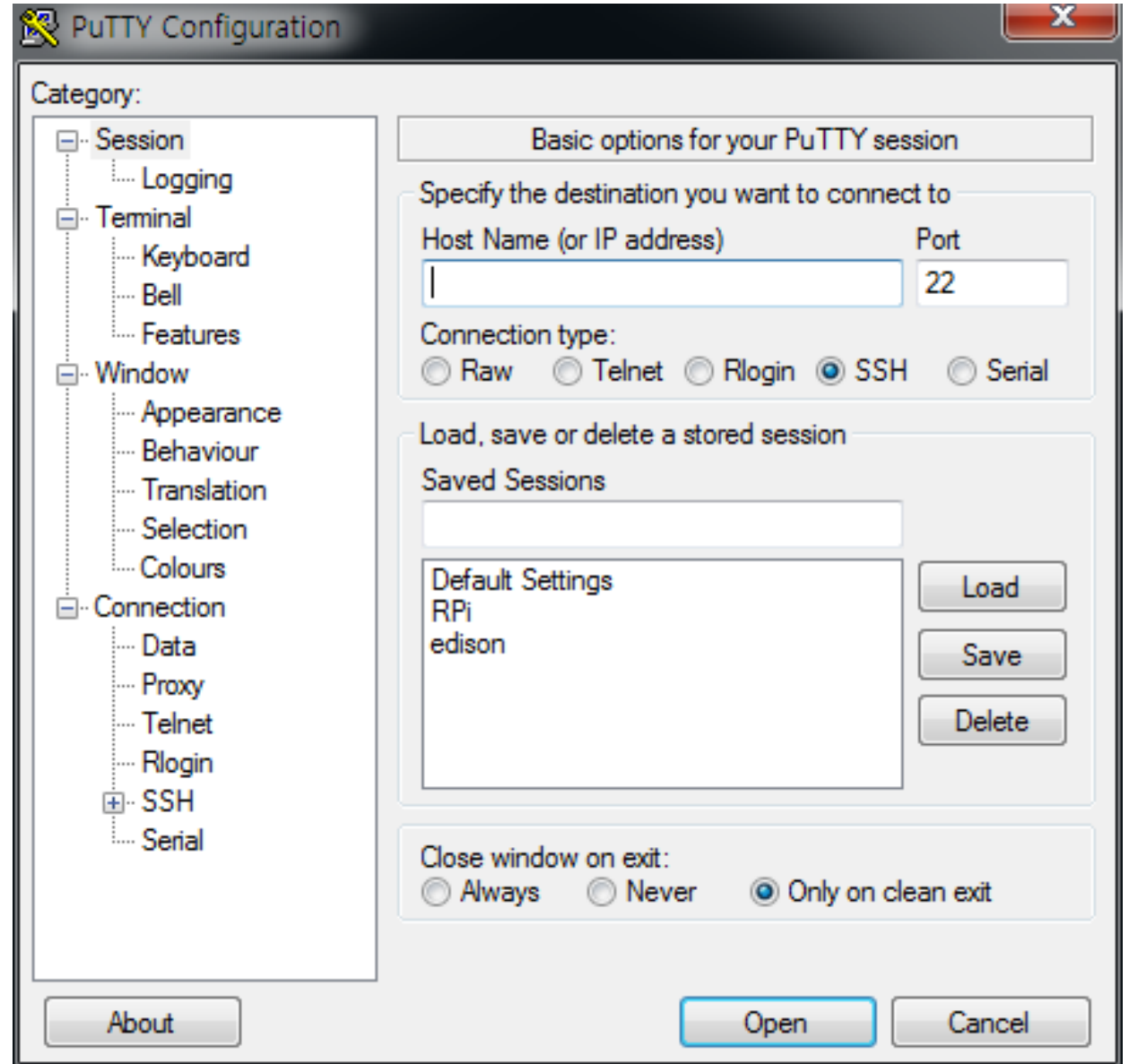
# VNC Viewer로 원격 접속한 라즈베리 파이 화면



# SSH(Secure Shell) 원격 접속

- SSH 원격 접속
  - 다른 컴퓨터에서 CLI 모드 터미널 창으로 라즈베리 파이 사용 가능
    - 명령 실행, 프로그램 작성/실행
  - SSH 서버 활성화 필요
    - VNC 활성화 한 설정 화면에서 할 수 있음
- SSH 접속 클라이언트
  - 윈도우 SSH 클라이언트 중 PuTTY 프로그램 사용 (무설치로 실행 가능)
    - <http://www.chiark.greenend.org.uk/~sgtatham/putty/>
  - 리눅스, 맥 OS의 경우 내장된 터미널 프로그램 이용

- PuTTY 실행 화면
  - Host Name (or IP address)
    - 라즈베리 파이의 IP 주소 입력

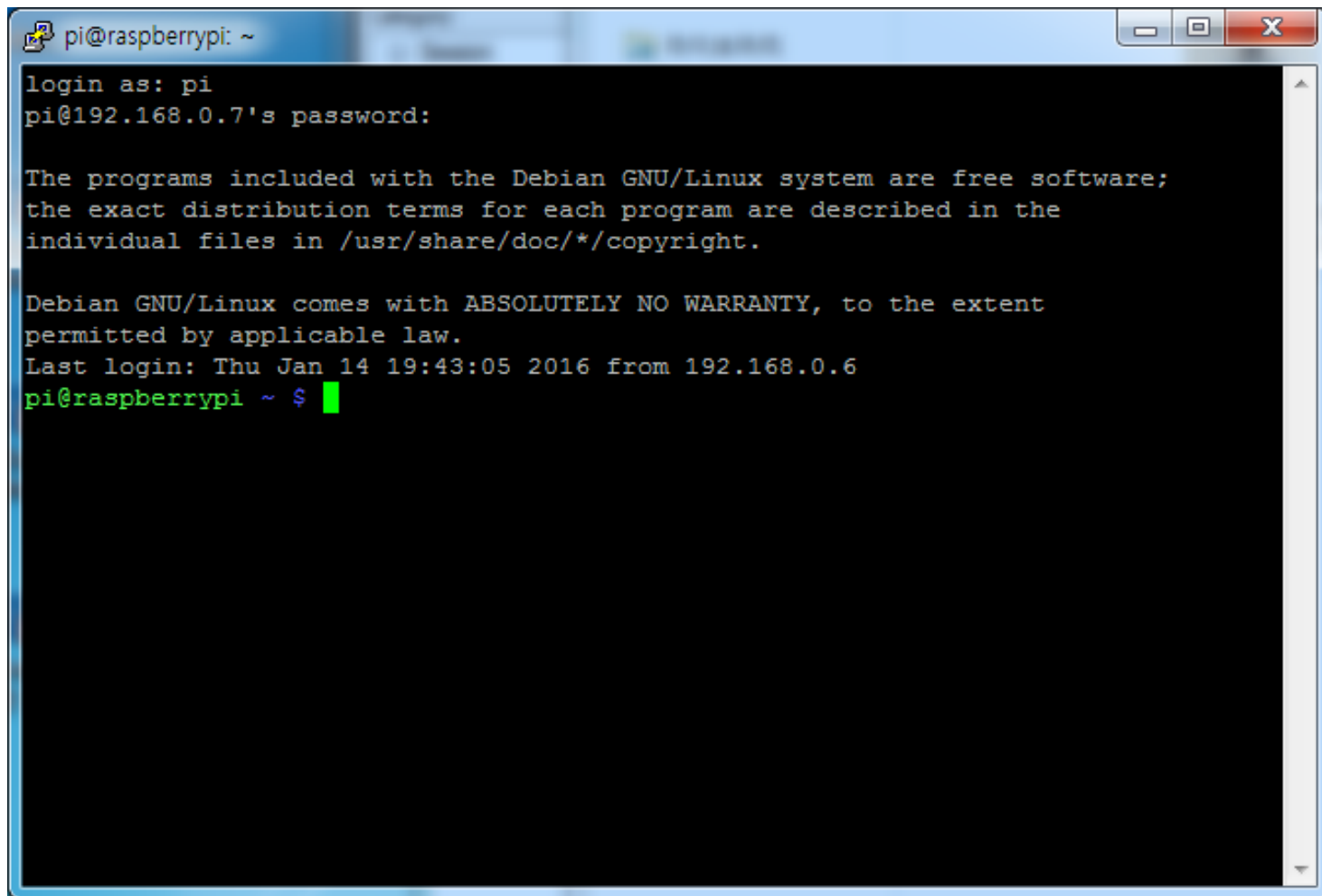


- PuTTY를 통한 SSH 접속 화면

- 기본 login
  - pi
- password
  - raspberry

- CLI 모드

- 리눅스 명령어를 입력하여 작업 수행



```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.7's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan 14 19:43:05 2016 from 192.168.0.6
pi@raspberrypi ~ $
```

# 리눅스 기본 명령어

- ls
  - 현재 디렉토리 내용을 표시
  - -a 옵션: 이름이 .으로 시작하는 파일도 보여줌
  - -l 옵션: 각 파일과 디렉토리에 대한 퍼미션, 소유자, 그룹, 크기 등 자세한 정보를 같이 보여줌
  - 두 가지 플래그를 같이 사용하고 싶으면 -al이라고 입력
- cd
  - 디렉토리를 변경. 현재 디렉토리 아래 temp라는 디렉토리가 있는 경우, cd temp라고 하면 temp 디렉토리로 이동
- pwd
  - 현재 디렉토리 이름을 표시
- mkdir
  - 새로운 디렉토리를 만들
  - mkdir iot : 현재 디렉토리 상에 iot라는 이름의 디렉토리를 생성
- rmdir
  - 디렉토리를 삭제
  - rmdir iot : 현재 디렉토리에 있는 iot 디렉토리를 삭제 (단, 삭제하려는 디렉토리가 비어있는 경우만 삭제)
- rm
  - 파일을 삭제
  - rm temp.txt : 현재 디렉토리에 있는 temp.txt 파일을 삭제 (-r 옵션을 사용하면 디렉토리 삭제 가능)

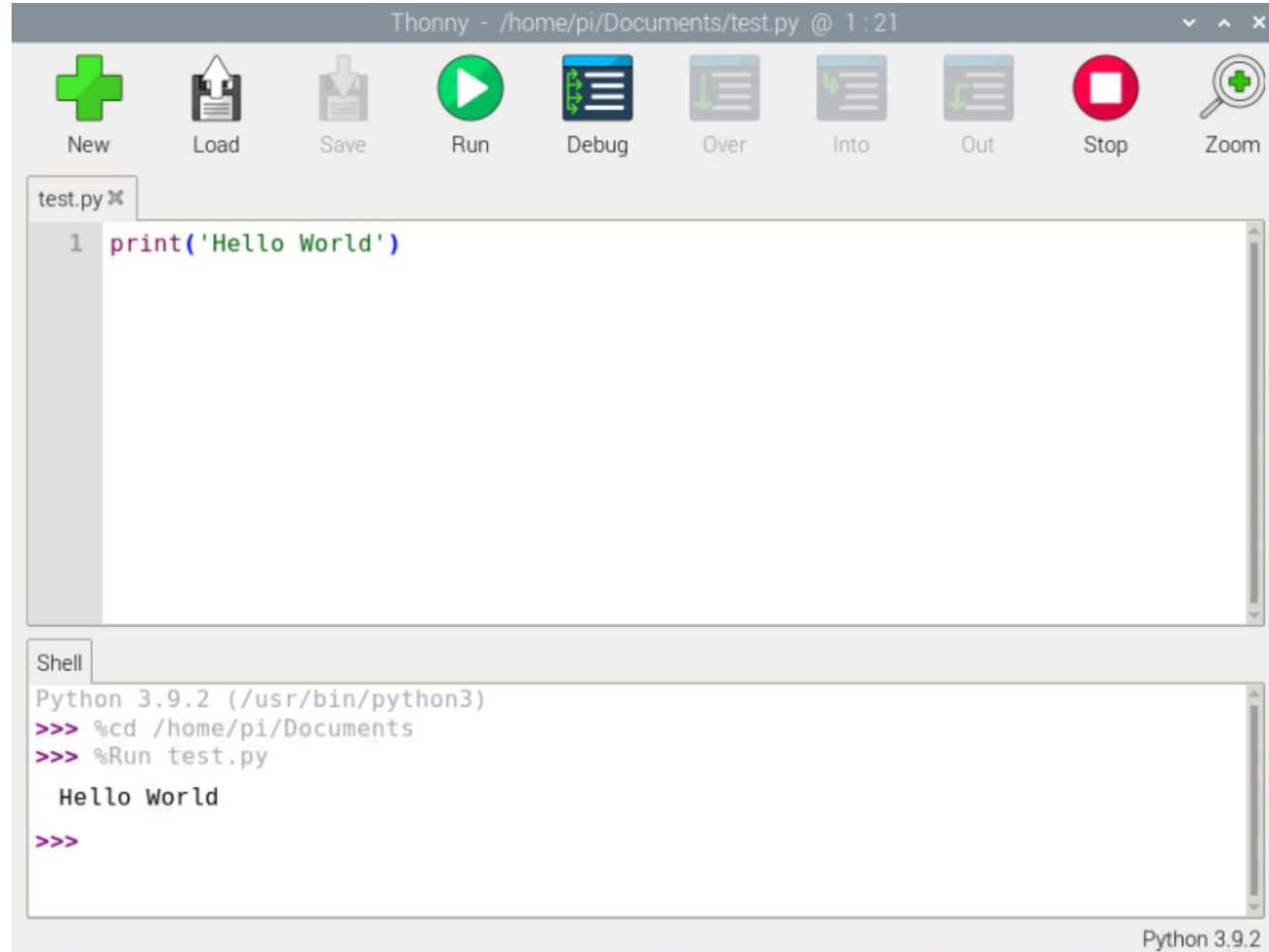
# 리눅스 기본 명령어

- cp
  - 파일에 대한 복사본 생성
  - cp temp.txt iot.txt : 현재 디렉토리에 있는 temp.txt 파일을 복사하여 iot.txt라는 이름으로 파일을 생성
  - cp temp.txt /home/pi2/iot/iot.txt : 현재 디렉토리의 temp.txt 파일을 복사하여 /home/pi2/iot 디렉토리 안에 iot.txt라는 이름으로 파일을 생성
- mv
  - 파일 혹은 디렉토리를 이동 혹은 파일이나 디렉토리 이름을 변경
  - mv temp.txt /home/pi2/iot/ : 현재 디렉토리의 temp.txt 파일을 /home/pi2/iot 디렉토리 아래로 이동
- cat
  - 파일의 내용을 표시
  - cat iot.txt : iot.txt 파일의 내용을 화면에 보여줌
- sudo
  - root (superuser) 권한으로 어떤 명령어를 실행할 때 사용
- passwd
  - 계정 암호 변경하기 위해 사용



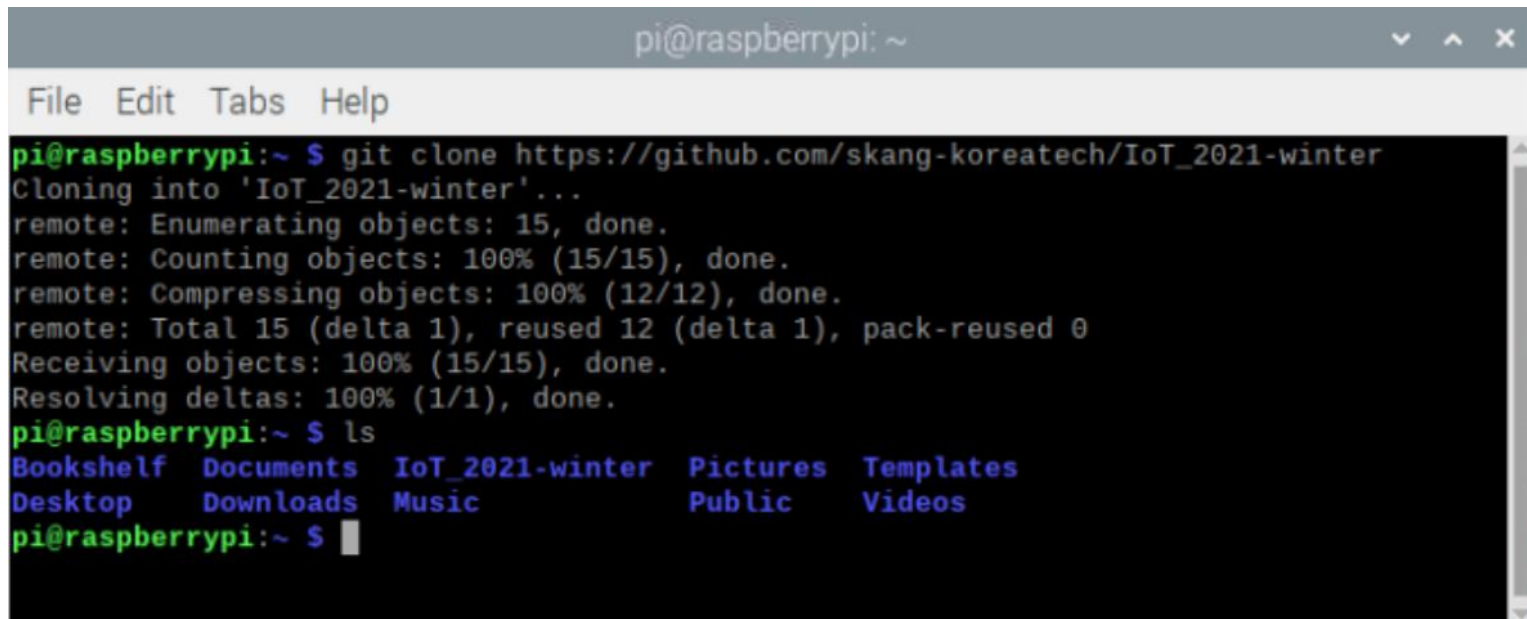
# Raspberry Pi Python 프로그래밍

- Thonny Python IDE 이용
  - Raspberry Pi OS에 기본 설치되어 있는 Python 프로그램 개발 도구



# 예제 프로그램 github 저장소

- [https://github.com/skang-koreatech/IoT\\_2023-winter](https://github.com/skang-koreatech/IoT_2023-winter)
- Raspberry Pi에 저장소의 코드를 복제하기
  - `git clone https://github.com/skang-koreatech/IoT_2023-winter`

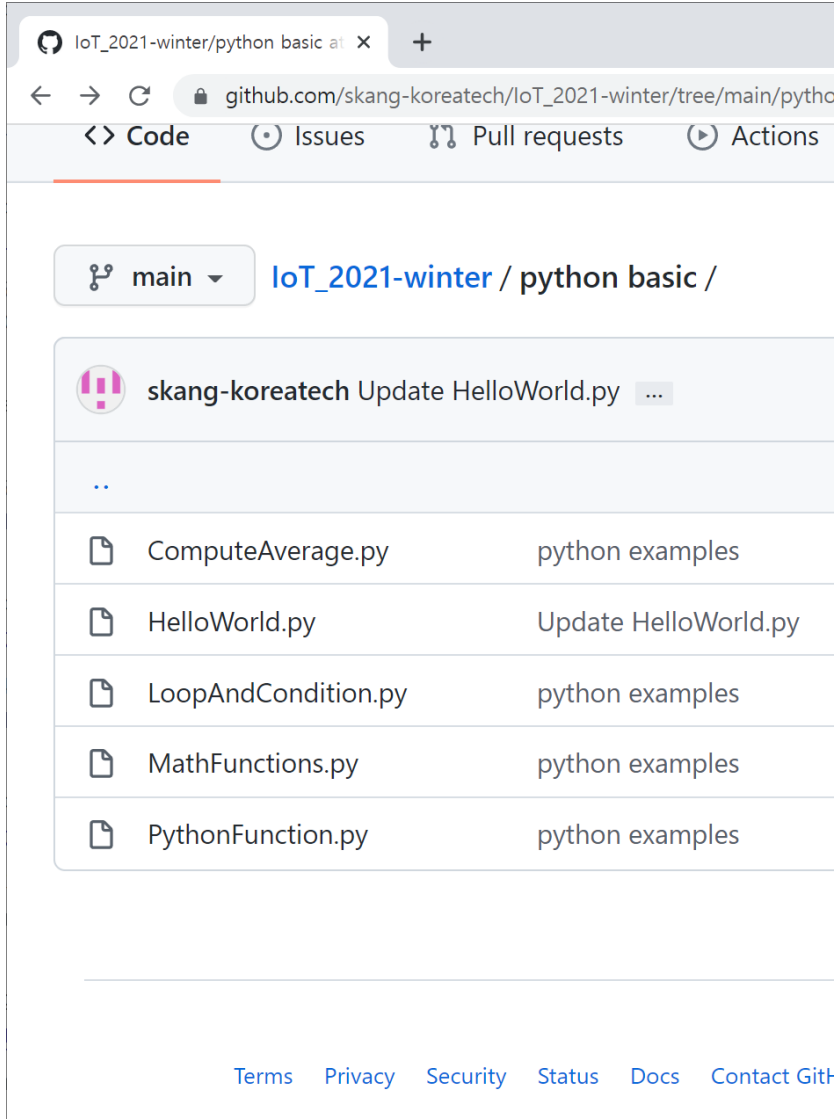


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ git clone https://github.com/skang-koreatech/IoT_2021-winter  
Cloning into 'IoT_2021-winter'...  
remote: Enumerating objects: 15, done.  
remote: Counting objects: 100% (15/15), done.  
remote: Compressing objects: 100% (12/12), done.  
remote: Total 15 (delta 1), reused 12 (delta 1), pack-reused 0  
Receiving objects: 100% (15/15), done.  
Resolving deltas: 100% (1/1), done.  
pi@raspberrypi:~ $ ls  
Bookshelf  Documents  IoT_2021-winter  Pictures  Templates  
Desktop    Downloads  Music            Public    Videos  
pi@raspberrypi:~ $
```

- 저장소에서 업데이트 된 코드를 받아오기
  - Raspberry Pi에 복제한 저장소 폴더로 이동 후
  - git pull

```
pi@raspberrypi: ~/IoT_2021-winter
File Edit Tabs Help
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 1), reused 12 (delta 1), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (1/1), done.
pi@raspberrypi:~ $ ls
Bookshelf  Documents  IoT_2021-winter  Pictures  Templates
Desktop    Downloads  Music            Public    Videos
pi@raspberrypi:~ $ cd IoT_2021-winter
pi@raspberrypi:~/IoT_2021-winter $ ls
'python basic'  README.md
pi@raspberrypi:~/IoT_2021-winter $ git pull
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only        # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
Already up to date.
pi@raspberrypi:~/IoT_2021-winter $
```

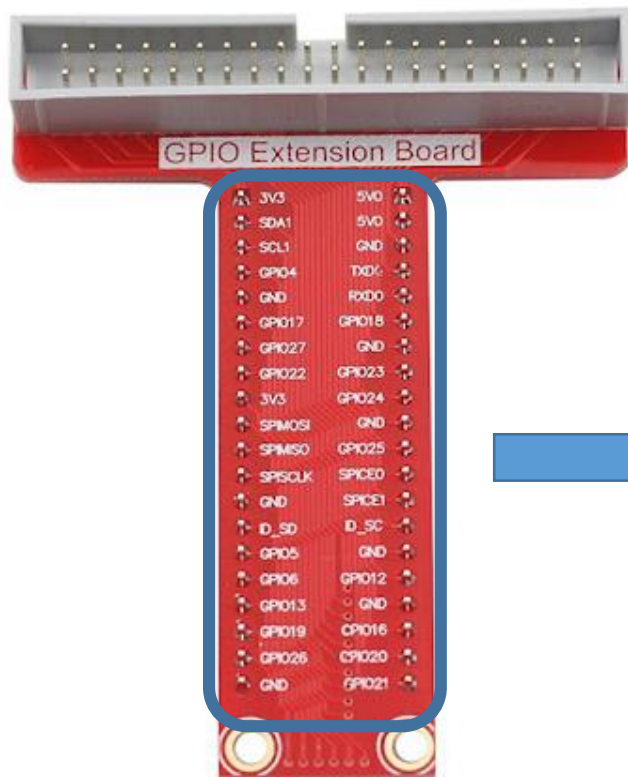
# Python 예제 프로그램 둘러보기



- HelloWorld.py
  - 텍스트 출력
- ComputeAverage.py
  - 사용자 데이터 입력, 연산, 출력
- MathFunctions.py
  - 여러 가지 수학 함수 연산 기능을 구현한 모듈 이용
    - 모듈: 미리 만들어진 특정 기능을 수행하는 프로그램의 모음
- LoopAndCondition.py
  - 반복문, 조건문 이용
- PythonFunction.py
  - 함수를 정의하고 이를 이용하여 프로그램 작성

# Raspberry Pi GPIO

- 라즈베리 파이에서 센서를 제어하고  
센서 데이터를 읽거나 액추에이터를  
제어하기 위해서 GPIO를 이용
- 이를 위해서 GPIO 프로그래밍 필요



## Raspberry Pi 3 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)	DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)	(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

# RPi.GPIO 모듈

- RPi.GPIO

- GPIO (General Purpose Input Output) 핀을 사용하기 위한 Python 모듈
- 이를 이용하여 센서/액추에이터를 이용하는 프로그램 구현
- <http://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>
- Raspberry Pi OS에 기본 설치되어 있음
  - 확인 : Command line 터미널에 python 입력 후 모듈을 import 해 본다.  
에러가 발생하지 않으면 이용 가능한 것임





pi@raspberrypi: ~

[GitHub - skang-korea...

pi@raspberrypi: ~

[pi]



07:50



Wastebasket

pi@raspberrypi: ~

File Edit Tabs Help

```
pi@raspberrypi:~ $ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

File Edit Tabs Help

```
pi@raspberrypi:~ $ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import RPi.GPIO as gpio
>>> gpio.setmode(gpio.BCM)
```

# RPi.GPIO 모듈 기본

- 모듈 import
  - `import RPi.GPIO as gpio`
- 초기 설정
  - `gpio.setmode(gpio.BCM)`
  - `gpio.setup(channel, gpio.IN)`
    - 여기서 `channel`은 사용하려고 하는 핀 번호
      - 리스트를 이용하여 동시에 여러 개의 핀 번호를 설정할 수 있음
    - 두번째 파라미터는 입력용/출력용을 설정
    - 출력용으로 사용하고자 하는 경우, `gpio.OUT`

# RPi.GPIO 모듈 기본

- 입력/출력

- `gpio.input(channel)`

- `channel` 변수로 명시된 핀에서 값을 읽음
    - 0 / `gpio.LOW` / `False` 또는 1 / `gpio.HIGH` / `True`를 반환

- `gpio.output(channel, state)`

- `channel` 변수로 명시된 핀에 `state`에 해당하는 값을 출력
    - `state`는 0 / `gpio.LOW` / `False` 또는 1 / `gpio.HIGH` / `True` 가 될 수 있음

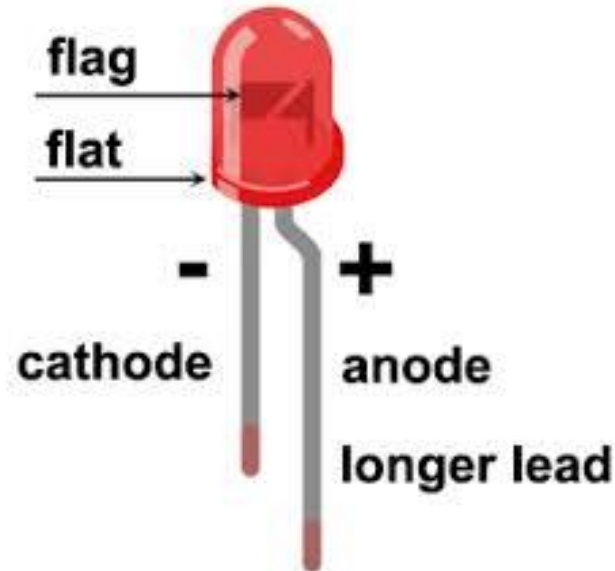
- 종료 시

- `gpio.cleanup()`

## 2) IoT 디바이스 프로그래밍

# LED 제어하기

- Raspberry Pi GPIO 핀을 통해 LED를 켜고 끄는 것을 제어
- 준비
  - Raspberry Pi와 GPIO 케이블로 연결된 코블러 브레이크아웃 보드와 브레드보드
  - LED
  - 저항
  - 점퍼 와이어



- 회로 구성

- 전원 공급용 GPIO 핀 연결

- 이 예제에서는 5번 핀 사용
    - 코블러 브레이크아웃 보드에서 GPIO 5번 핀을 브레드보드의 빨간선으로 표시된 (+) 홀에 와이어로 연결

- 그라운드 연결

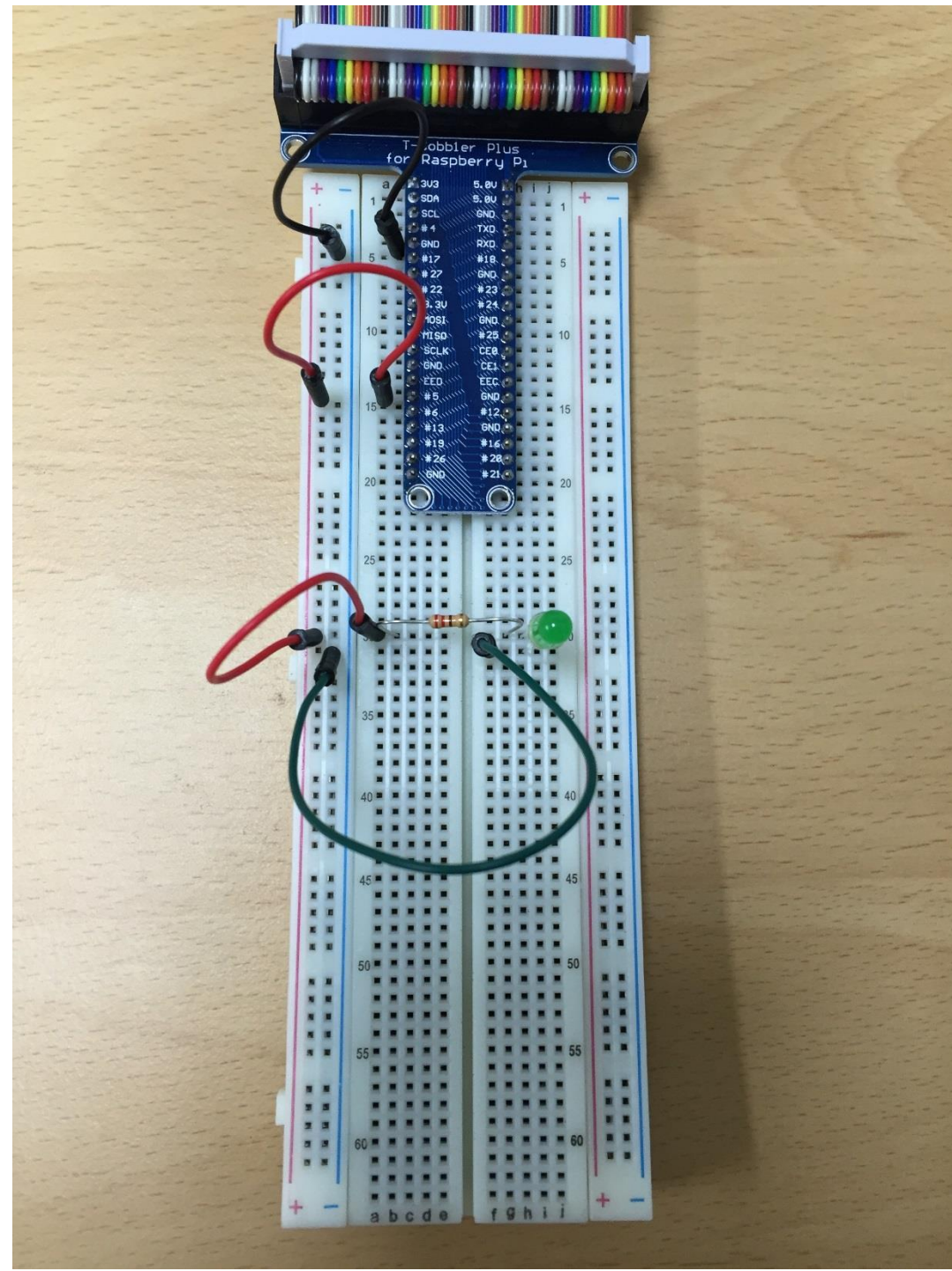
- 코블러 브레이크아웃 보드의 그라운드(GND) 핀을 와이어로 브레드보드의 파란선 (-) 홀에 연결

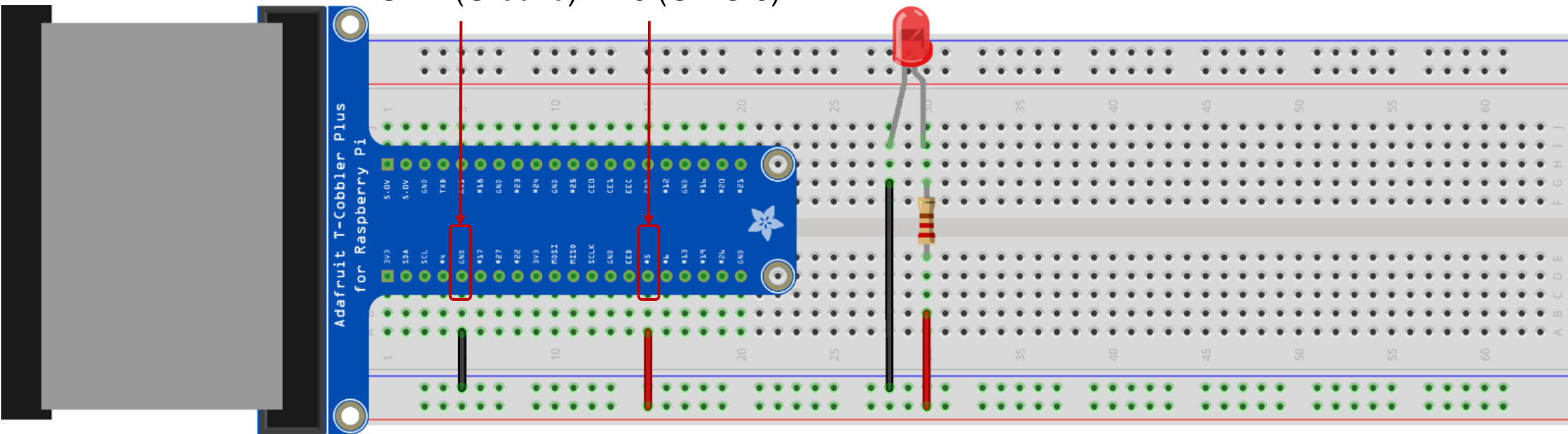
- LED 장착

- LED의 긴 전극이 양극, 짧은 전극이 음극이므로 음극을 브레이크아웃 보드의 GND 핀에 연결된 파란선 (-) 홀과 와이어로 연결

- 저항 연결

- LED의 양극과 브레드보드 빨간선 (+) 홀 사이에 작은 저항 하나를 연결







```
import RPi.GPIO as gpio
import time
led_pin = 5
gpio.setmode(gpio.BCM)
gpio.setup(led_pin, gpio.OUT)

gpio.output(led_pin, True)
time.sleep(0.5)
gpio.output(led_pin, False)
time.sleep(0.5)
gpio.output(led_pin, True)
time.sleep(0.5)
gpio.output(led_pin, False)
time.sleep(0.5)

print("Blink Finished")
gpio.cleanup()
```

- 예제 코드: /actuator\_led/simpleLedBlink.py
- 0.5초 간격으로 LED를 켜다 껐다 2번 반복하는 예제 프로그램
- 5번 GPIO 핀을 출력핀으로 사용하여 LED 제어
  - 5번 핀이 아닌 다른 핀을 사용하는 경우 3번 라인에서 해당 번호에 맞게 숫자를 변경해주어야 함
- 10번 켜다 껐다 반복하도록 프로그램을 변경한다면?

## 예제 코드: /actuator\_led/blinkLed.py

```
import RPi.GPIO as gpio
import time
led_pin = 5
gpio.setmode(gpio.BCM)
gpio.setup(led_pin, gpio.OUT)

def blinkLED(numTimes, speed):
    for i in range(0, numTimes):
        print("Iteration " + str(i+1))
        gpio.output(led_pin, True)
        time.sleep(speed)
        gpio.output(led_pin, False)
        time.sleep(speed)
    print("Blink Finished")
    gpio.cleanup()
```

```
try:
    iterations = input("Enter total number of times to blink: ")
    speed = input("Enter length of each blink(seconds): ")
    blinkLED(int(iterations), float(speed))
except KeyboardInterrupt:
    gpio.cleanup()
```