

1. Raspi의 hostname : Kminpi & IP adress :  
192.168.137.69 & Local PC의 IP address :  
192.168.137.1

2. Local PC 내의 host파일 내용

```
C:\Users\hwp>cat ../../Windows/System32/drivers/etc/hosts
192.168.137.69 draw.kmin.com
192.168.137.69 kmin.com
```

3. 호스트1(kmin)

서버네임:kmin.com

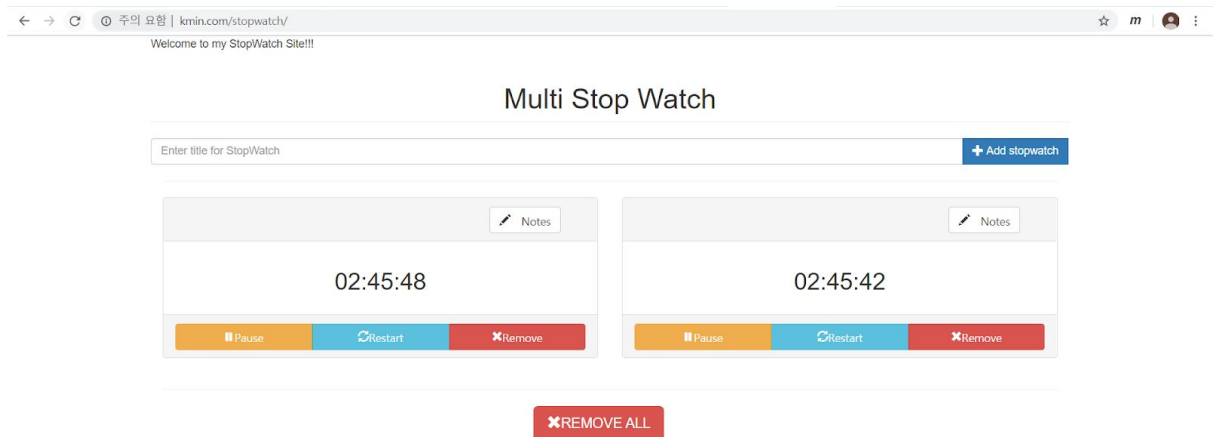
사용된 github저장주소 :

저장된 Raspi내의 디렉토리 경로 :

<https://github.com/Kim-Min-Hyeok/Multi-Stop-Watch/>

home/kmin/html/stopwatch

웹브라우저 화면 캡처 :



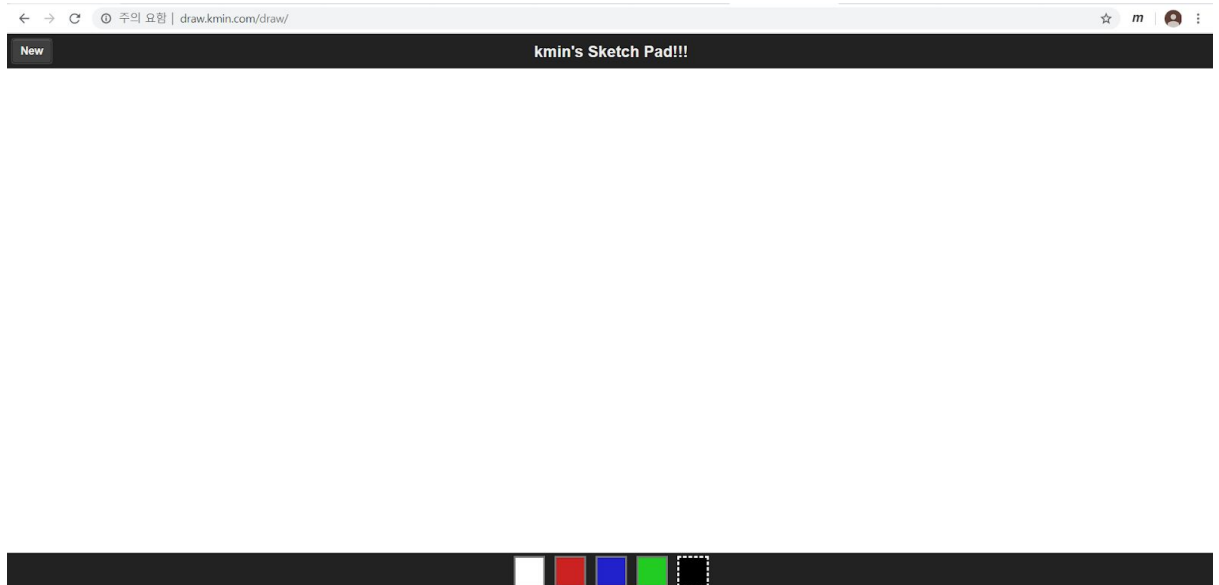
4. 호스트1(kjun)

서버네임:draw.kmin.com

사용된 github저장주소 :

<https://github.com/Kim-Min-Hyeok/html5-canvas-drawing-app>

저장된 Raspi내의 디렉토리 경로 : /html/stopwatch  
웹브라우저 화면 캡처 :



## 실습미션 #1

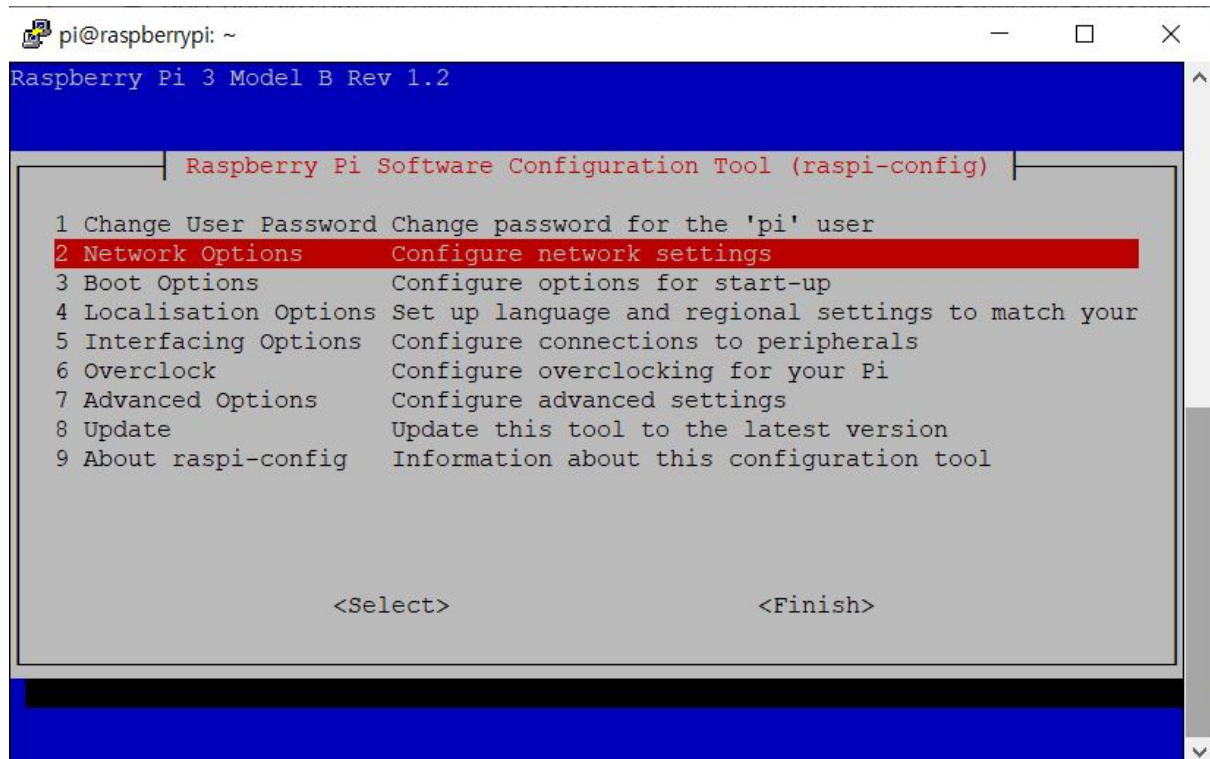
### 1. 자신의 라즈베리파이의 고유한 hostname 변경

```
pi@raspberrypi:~ $ sudo su -  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
root@raspberrypi:~#
```

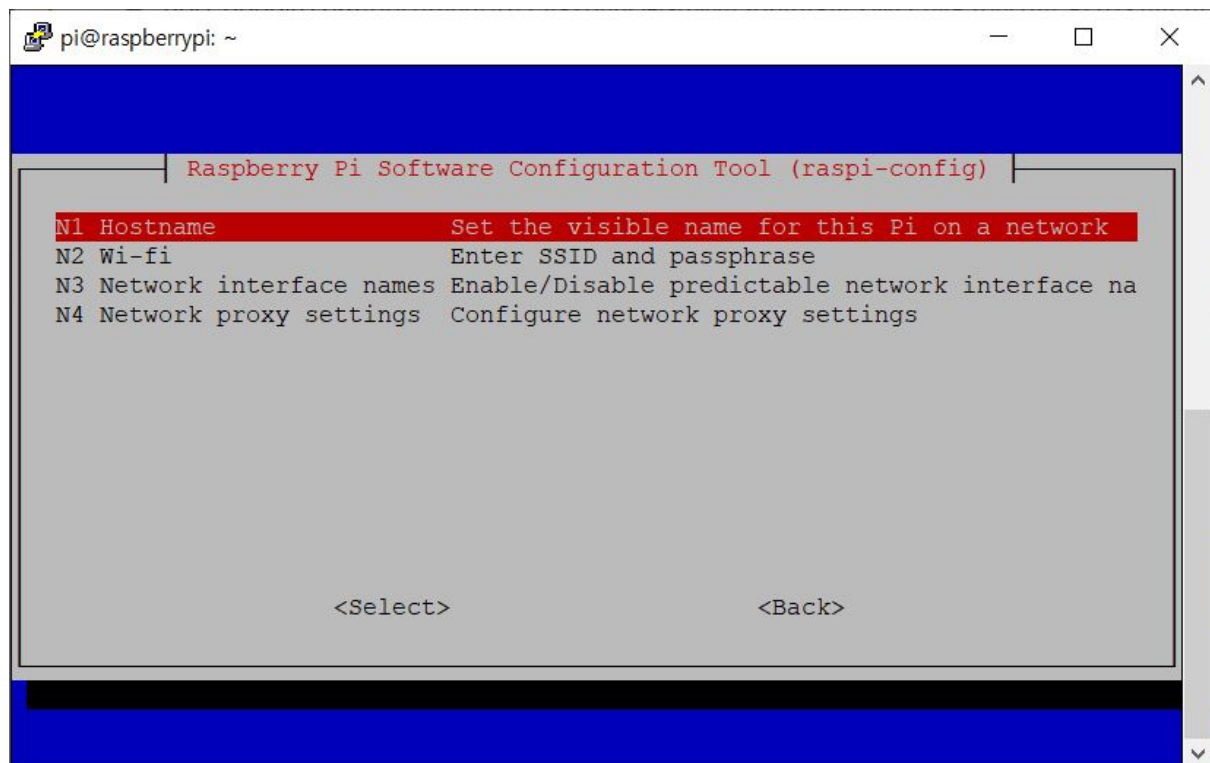
super유저(root)로 변경

```
root@raspberrypi:~# raspi-config
```

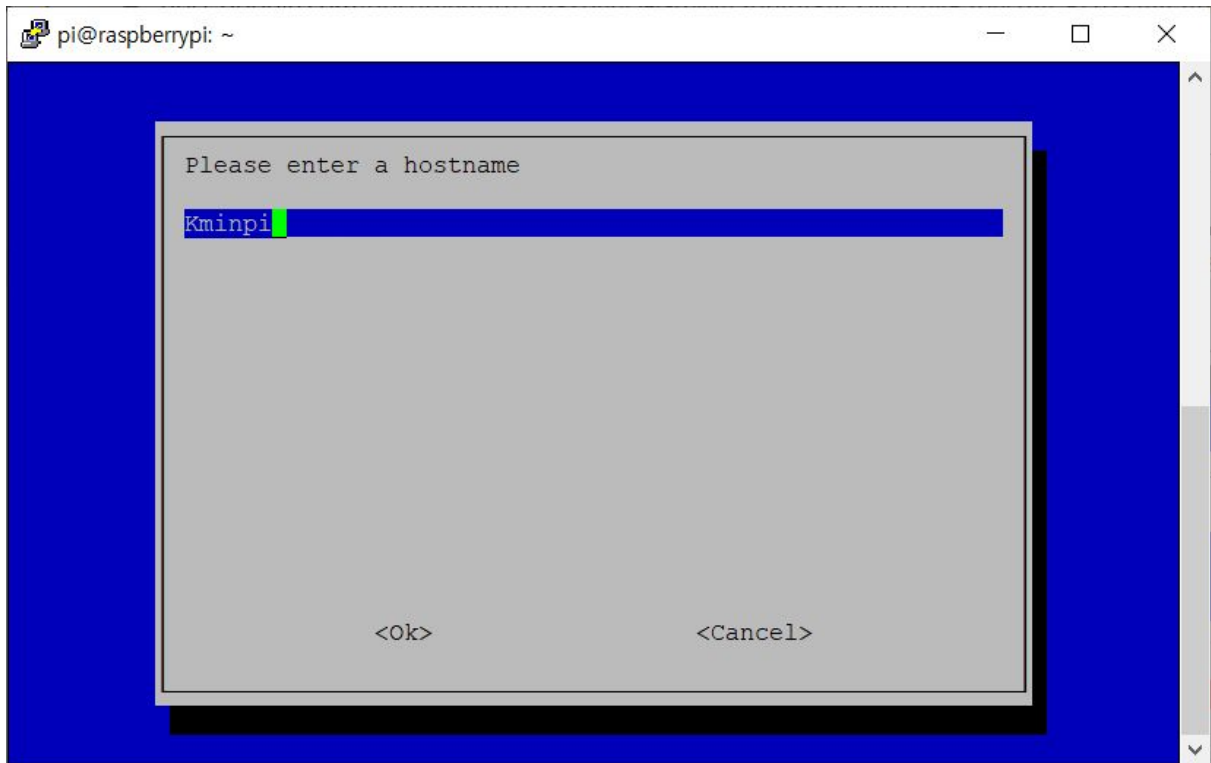
설정 변경



## 2번메뉴 선택



## 1번메뉴의 host이름을 변경하는 메뉴를 선택



원하는 이름으로 호스트네임을 변경하여 준다.

```
pi@Kminpi:~ $ █
```

호스트네임을 변경한 후 reboot해주면 다음과 같이 호스트네임이 변경된다.

## 2. 사용자 계정 생성 및 도메인 이름 결정

```
pi@Kminpi:~ $ sudo su -  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
root@Kminpi:~# █
```

또다시 슈퍼유저가 되기 위해 sudo su -를 한다.

```
root@Kminpi:~# useradd -m kmin
```

이번에는 사용자 계정을 생성한다. (kmin이라는 유저)

```
root@Kminpi:~# passwd kmin  
New password:  
Retype new password:  
passwd: password updated successfully
```

kmin이라는 유저의 password를 생성하여 준다.

### 3. 사용자의 웹사이트 세팅 파일을 제작한다.(도메인 이름에 대한 nginx 가상호스트 설정)

```
root@Kminpi:~# vi /etc/nginx/sites-available/kmin.com
```



```
server {  
    listen 80;  
    listen [::]:80;  
  
    server_name kmin.com;  
  
    root /home/kmin/html;  
    index index.html;  
}
```

```
root@Kminpi:~# ln -s /etc/nginx/sites-available/kmin.com /etc/nginx/sites-enabled/kmin.com
```

다음 세팅 파일을 sites-available 디렉토리와 심볼릭 링크를 제작한다.

```
root@Kminpi:~# nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

웹서비스 세팅 테스트

```
root@Kminpi:~# service nginx restart
```

웹서비스 재시작

### 4. kmin유저가 ssh접속 전에 hosts파일을 변경(로컬 hosts파일 변경)

```
C:\Windows\System32\drivers\etc>ls  
hosts hosts.ics lmhosts.sam networks protocol services
```

위와 같은 경로에 hosts파일이 있다.

```
C:\. 관리자: 명령 프롬프트 - vim hosts
192.168.137.112 kmin.com
~
~
~
```

hosts파일에 ip와 도메인네임을 넣어준다.

## 5. kmin유저 ssh에 접속, 초기 비밀번호 변경

```
kmin@Kminpi:~ $ passwd kmin
Changing password for kmin.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

아까 슈퍼유저가 지정해준 비밀번호를 바꾼다.

```
login as: kmin
kmin@192.168.137.150's password:
Linux Kminpi 4.19.97-v7+ #1294 SMP Thu Jan 30 13:15:58 GMT 2020 armv7l

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.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

kmin@Kminpi:~ $
```

다시 로그인해보면 바뀐 비밀번호가 사용된다.

```
kmin@Kminpi:~/html $ vi index.html
```

## 6. 사이트 시험(사용자 계정으로 접속하여 웹 사이트 접속 테스트)

사이트가 잘 돌아가는지 시험해보기위해 index.html파일을 만들어서 html디렉토리 안에 넣어준다.

```
kmin@Kminpi: ~/html
kmin's web site
```



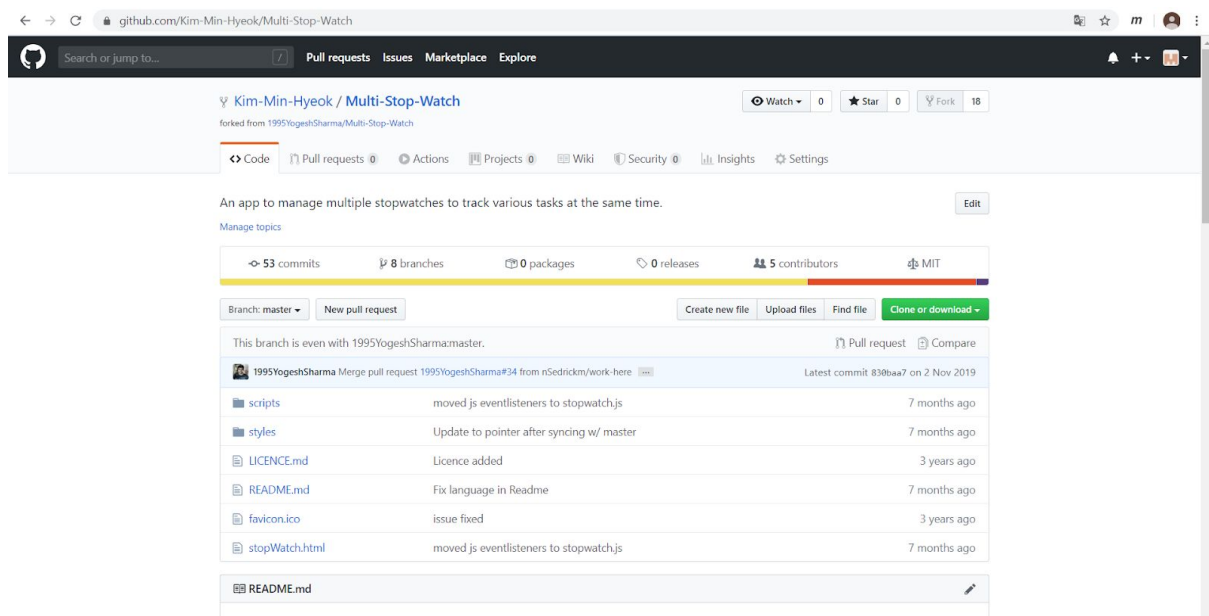
<index.html>



kmin's web site!!

hosts에서 수정해준 도메인 네임을 치면 다음과 같이 index.html의 내용이 나온다.

7. 마음에 드는 웹 템플릿 사용(github에서 적절한 웹사이트 소스 저장소를 찾아 fork)



8. fork 한 자신의 github 저장소에서 clone하여 웹사이트 페이지 제작

```
kmin@Kminpi:~/html $ git clone https://github.com/Kim-Min-Hyeok/Multi-Stop-Watch
.git
Cloning into 'Multi-Stop-Watch'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 139 (delta 17), reused 26 (delta 10), pack-reused 99
Receiving objects: 100% (139/139), 38.05 KiB | 197.00 KiB/s, done.
Resolving deltas: 100% (65/65), done.
kmin@Kminpi:~/html $
```

```
kmin@Kminpi:~/html $ mv Multi-Stop-Watch stopwatch
```

```
kmin@Kminpi:~/html $ ls
index.html  stopwatch
```

← → ↻ ⓘ 주의 요람 | kmin.com/stopwatch/

☆ m | 👤 ⋮

403 Forbidden

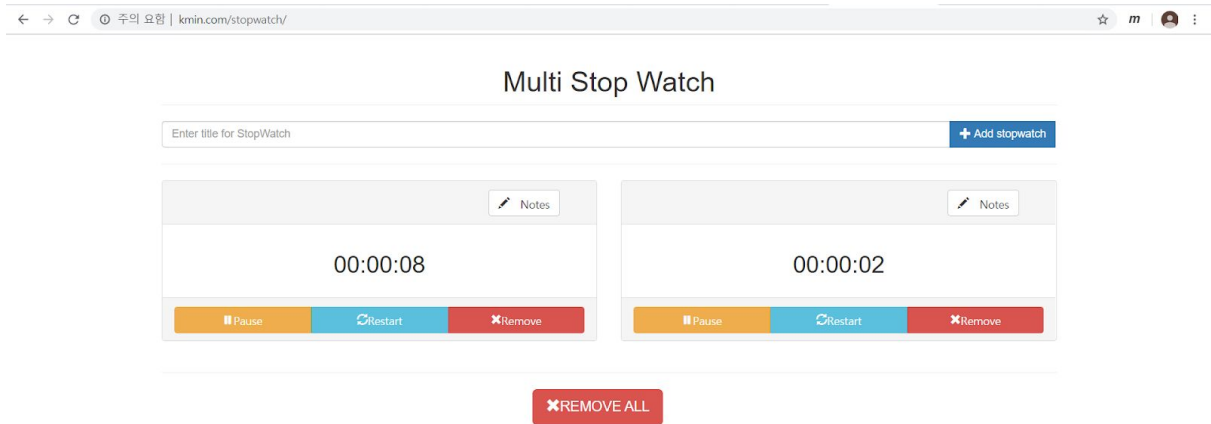
nginx/1.14.2

그러나 도메인 네임을 입력하면 다음과 같이 뜬다.

```
kmin@Kminpi:~/html $ cd stopwatch
kmin@Kminpi:~/html/stopwatch $ ls
favicon.ico  LICENCE.md  README.md  scripts  stopWatch.html  styles
kmin@Kminpi:~/html/stopwatch $ ln -s stopWatch.html index.html
kmin@Kminpi:~/html/stopwatch $ ls
favicon.ico  index.html  LICENCE.md  README.md  scripts  stopWatch.html  styles
kmin@Kminpi:~/html/stopwatch $
```

그래서 stopwatch디렉토리를 살펴보았더니 index.html파일이 없다 그래서 이 template의 html파일을 index.html로 링크를 걸어주었다.





그랬더니 스톱워치 웹사이트가 제대로 실행된다.

## 9. 적절한 내용을 수정하여 웹 브라우저에서 테스트

```

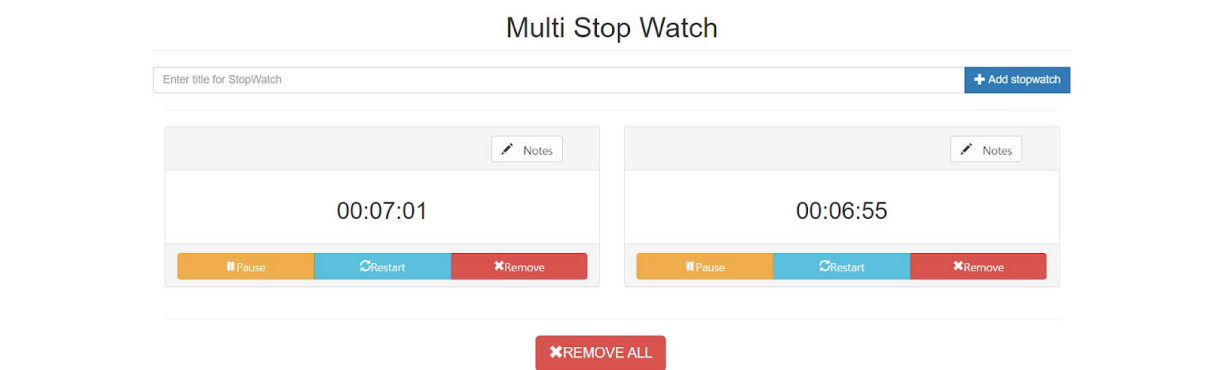
</div>
<div class="modal-body" id="notesModalBody">
  <p> to be filled by function</p>
</div>
<hr>
<div class="modal-footer" id="notesModalFooter">
  <p> to be filled by function </p>
</div>
</div>
</div>
</div>
</div>

<body class='container'>
  <div class="row">
    <div class="container-fluid">
      <h1 class="page-header text-center"> Multi Stop Watch </h1>
      <div class="input-group">
        <input id='title' type='text' class="form-control" placeholder="Enter ti
        tle for Stopwatch" />
        <span class="input-group-btn">
          <button id="addWatch-btn" class="form-control btn btn-primary"><span c
          lass="glyphicon glyphicon-plus"></span>

```

```
</div>
<div class="modal-body" id="notesModalBody">
  <p> to be filled by function</p>
</div>
<hr>
<div class="modal-footer" id="notesModalFooter">
  <p> to be filled by function </p>
</div>
</div>
</div>
</div>
</div>

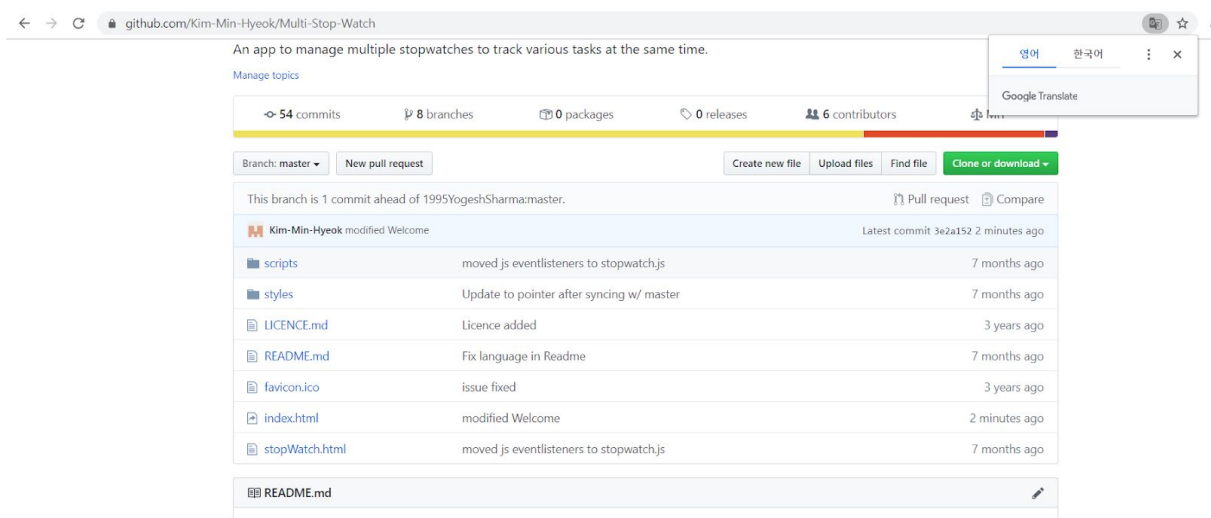
<body class='container'>
  <div class="row">
    <div class="container-fluid">
      Welcome to my StopWatch Site!!!
      <h1 class="page-header text-center"> Multi Stop Watch </h1>
      <div class="input-group">
        <input id='title' type='text' class="form-control" placeholder="Enter ti
        tle for StopWatch" />
        <span class="input-group-btn">
```



다음과 같이 vi를 통하여 index.html의 title text를 수정하였다.(위에 Welcome to my StopWatch Site!!!가 떠 있는 것을 볼 수 있다.)

```
kmin@Kminpi:~/html/stopwatch $ git add index.html
kmin@Kminpi:~/html/stopwatch $ git commit -m "modified Welcome"
[master 3e2a152] modified Welcome
 1 file changed, 1 insertion(+)
 create mode 120000 index.html
kmin@Kminpi:~/html/stopwatch $ git push https://github.com/Kim-Min-Hyeok/Multi-Stop-Watch.git
Username for 'https://github.com': Kim-Min-Hyeok
Password for 'https://Kim-Min-Hyeok@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 147.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Kim-Min-Hyeok/Multi-Stop-Watch.git
 830baa7..3e2a152  master -> master
kmin@Kminpi:~/html/stopwatch $
```

변경한 내용을 git add, commit, push를 통하여 내 github에 업데이트 해주었다.



다음과 같이 index.html이 변경된 것을 알 수 있다.(commit 메시지)

## 실습미션 #2

user를 만들어 준다.

```
root@Kminpi:~# useradd -m kmin
useradd: user 'kmin' already exists
```

kmin은 지난번에 생성했으므로 이렇게 뜬다.

```
root@Kminpi:~# useradd -m kjun
root@Kminpi:~# passwd kjun
New password:
Retype new password:
passwd: password updated successfully
root@Kminpi:~#
```

이번에는 kjun이라는 유저이름으로 user를 생성하고 비밀번호도 설정해준다.

```
root@Kminpi:~# cd /etc/nginx/sites-available/
root@Kminpi:/etc/nginx/sites-available# ls -al
total 16
drwxr-xr-x 2 root root 4096 May 20 10:30 .
drwxr-xr-x 8 root root 4096 May 17 15:36 ..
-rw-r--r-- 1 root root 2412 Aug 13 2019 default
-rw-r--r-- 1 root root 108 May 20 10:30 kmin.com
root@Kminpi:/etc/nginx/sites-available#
```

현재 sites-available디렉토리 안에는 kmin.com이라는 저번에 만들어 준 사이트가 있다.

```
root@Kminpi:/etc/nginx/sites-available# cat kmin.com
server {
    listen 80;
    listen [::]:80;

    server_name kmin.com;

    root /home/kmin/html;
    index index.html;
}
root@Kminpi:/etc/nginx/sites-available#
```

이것이 지난번 만든 kmin.com세팅 파일이다.

```
root@Kminpi:/etc/nginx/sites-available# cp kmin.com draw.kmin.com
```

이번에는 draw.kmin.com을 또 한번 세팅하기 위해 저번에 세팅한 파일을 copy하여 조금만 수정하기로 한다.

```
server {
    listen 80;
    listen [::]:80;

    server_name draw.kmin.com;

    root /home/kjun/html;
    index index.html;
}
```

draw.kmin.com을 세팅하기 위하여 서버네임과  
유저네임을 수정하였다.

```
root@Kminpi:/etc/nginx/sites-available# ls -al
total 20
drwxr-xr-x 2 root root 4096 May 24 09:58 .
drwxr-xr-x 8 root root 4096 May 17 15:36 ..
-rw-r--r-- 1 root root 2412 Aug 13 2019 default
-rw-r--r-- 1 root root 113 May 24 09:58 draw.kmin.com
-rw-r--r-- 1 root root 108 May 20 10:30 kmin.com
```

draw.kmin.com파일이 생성된 것을 볼 수 있다.

```
root@Kminpi:/etc/nginx/sites-available# cd ../
root@Kminpi:/etc/nginx# ls
conf.d          koi-win         nginx.conf      sites-enabled
fastcgi.conf    mime.types      proxy_params    snippets
fastcgi_params  modules-available  scgi_params     uwsgi_params
koi-utf         modules-enabled  sites-available  win-utf
```

```
root@Kminpi:/etc/nginx# cd sites-enabled
root@Kminpi:/etc/nginx/sites-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 34 May 17 15:36 default -> /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root 35 May 20 10:31 kmin.com -> /etc/nginx/sites-available/kmin.com
```

실제로 안에 세팅파일을 넣어야 하는  
site-enabled디렉토리이다.

```
root@Kminpi:/etc/nginx/sites-enabled# ln -s /etc/nginx/sites-available/draw.kmin.com draw.kmin.com
root@Kminpi:/etc/nginx/sites-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 34 May 17 15:36 default -> /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root 40 May 24 10:02 draw.kmin.com -> /etc/nginx/sites-available/draw.kmin.com
lrwxrwxrwx 1 root root 35 May 20 10:31 kmin.com -> /etc/nginx/sites-available/kmin.com
```



그래서 방금 만든 파일을 이곳에 링크시켜준다.

```
root@Kminpi:/etc/nginx/sites-enabled# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@Kminpi:/etc/nginx/sites-enabled#
```

nginx를 테스트 해준다.

logout 명령을 통하여 라즈베리에서 빠져나온 후  
kjun유저로 다시 접속한다.

```
login as: kjun
kjun@192.168.137.112's password:
Linux Kminpi 4.19.97-v7+ #1294 SMP Thu Jan 30 13:15:58 GMT 2020 armv7l

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.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

kjun@Kminpi:~ $
```

```
kjun@Kminpi:~ $ ls
kjun@Kminpi:~ $
```

아직은 이 유저에게는 아무 디렉토리도 없다.

```
kjun@Kminpi:~ $ mkdir html
kjun@Kminpi:~ $ cd html
kjun@Kminpi:~/html $
```

그래서 html파일을 저장할 html디렉토리를 생성한다.

```
kjun@Kminpi:~/html $ cat > index.html
draw.kmin.com
```

먼저 이 서버를 시험하기 위하여 index.html을 간단한  
파일로 만들어 본다.

```
C:\Users\hwp>vim /etc/hosts
```

관리자 모드로 명령프롬프트에 들어가서 로컬pc의  
hosts파일 수정한다.



관리자: 명령 프롬프트 - vim /etc/hosts

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com          # source server
#       38.25.63.10       x.acme.com             # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1         localhost
#       ::1               localhost
192.168.137.69 kmin.com
192.168.137.69 draw.kmin.com
```

hosts파일에 ip주소와 도메인을 추가한다.

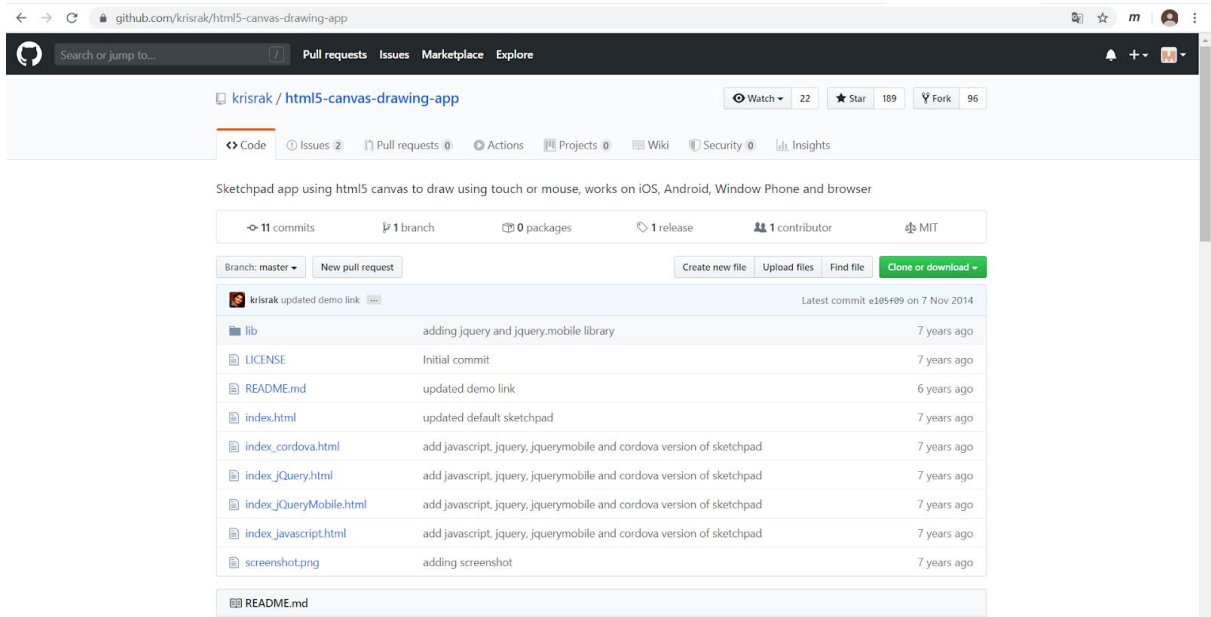
```
C:\Users\hnp>cat /etc/hosts
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com          # source server
#       38.25.63.10       x.acme.com             # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1         localhost
#       ::1               localhost
192.168.137.69 kmin.com
192.168.137.69 draw.kmin.com
```

hosts파일이 다음과 같이 바뀌었다.

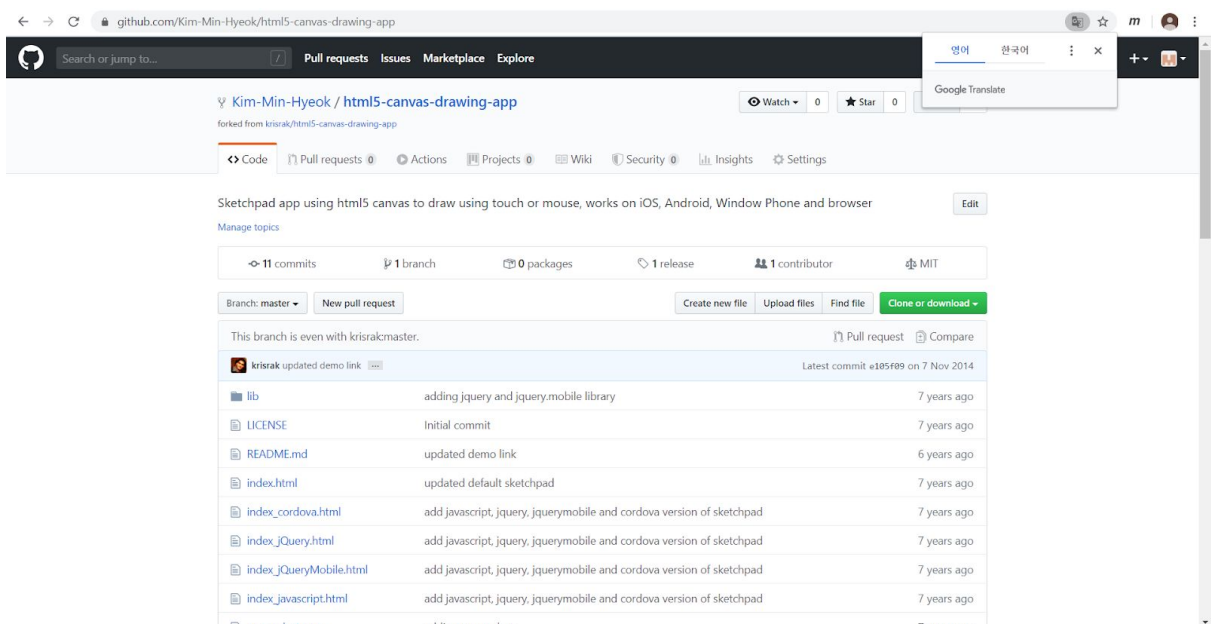
← → ↻ ⓘ 주의 요함 | draw.kmin.com

draw.kmin.com

그랬더니 도메인을 입력하니 아까 설정한 index.html파일의 결과가 출력된다.



이번에는 마음에 드는 html템플릿 하나를 깃허브에서 포크뜨기로 한다.



포크 뜯 그림판 기능의 html이다.

```
kjun@Kminpi:~ $ cd html
kjun@Kminpi:~/html $ pwd
/home/kjun/html
kjun@Kminpi:~/html $
```

```
kjun@Kminpi:~/html $ git clone https://github.com/Kim-Min-Hyeok/html5-canvas-drawing-app.git
Cloning into 'html5-canvas-drawing-app'...
```

그런 후 putty를 이용하여 git clone 명령을 실행하여 html 디렉토리에 다음과 같이 복제한다.

```
kjun@Kminpi:~/html $ ls -l
total 8
drwxr-xr-x 4 kjun kjun 4096 May 24 11:27 html5-canvas-drawing-app
-rw-r--r-- 1 kjun kjun  14 May 24 10:09 index.html
kjun@Kminpi:~/html $ mv html5-canvas-drawing-app draw
kjun@Kminpi:~/html $ ls -l
total 8
drwxr-xr-x 4 kjun kjun 4096 May 24 11:27 draw
-rw-r--r-- 1 kjun kjun  14 May 24 10:09 index.html
kjun@Kminpi:~/html $
```

다음과 같이 clone이 성공 하였다. 그러나 디렉토리 이름이 너무 길기에 mv 명령을 통하여 디렉토리명을 변경하여 주었다(draw)



/draw를 추가하여 알맞게 도메인을 입력하여 주면 다음과 같이 성공적으로 그림판이 뜬다.

```

};

    </script>
</head>
<body>
<div id="page">
    <div class="header">
        <a id="new" class="navbtn">New</a>
        <div class="title">Sketch Pad</div>
    </div>
<div id="content"><p style="text-align:center">Loading Canvas...</p></div>
<div class="footer">
    <div class="palette-case">
        <div class="palette-box">
            <div class="palette white"></div>
        </div>
        <div class="palette-box">
            <div class="palette red"></div>
        </div>
        <div class="palette-box">
            <div class="palette blue"></div>
        </div>
        <div class="palette-box">

```

이번에는 이 index.html파일을 수정하려고 한다.

```

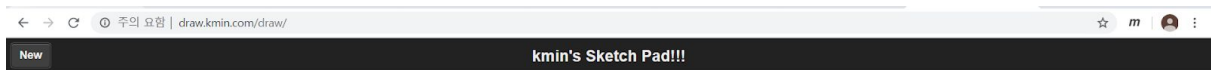
};

    </script>
</head>
<body>
<div id="page">
    <div class="header">
        <a id="new" class="navbtn">New</a>
        <div class="title">kmin's Sketch Pad!!!</div>
    </div>
<div id="content"><p style="text-align:center">Loading Canvas...</p></div>
<div class="footer">
    <div class="palette-case">
        <div class="palette-box">
            <div class="palette white"></div>
        </div>
        <div class="palette-box">
            <div class="palette red"></div>
        </div>
        <div class="palette-box">
            <div class="palette blue"></div>
        </div>
        <div class="palette-box">

```

나는 SketchPad라는 위의 제목을 변경하기 위하여 다음과 같이 수정하였다.





다시 새로고침을 해보면 보는 것과 같이 수정이 성공적으로 되었다.

```
kjun@Kminpi:~/html/draw $ git config --global user.email "21900102@handong.edu"
kjun@Kminpi:~/html/draw $ git config --global user.name "Kim-Min-Hyeok"
kjun@Kminpi:~/html/draw $ git add index.html
kjun@Kminpi:~/html/draw $ git commit -m "set title"
[master 8bd9b8a] set title
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
kjun@Kminpi:~/html/draw $ git push https://github.com/Kim-Min-Hyeok/html5-canvas
-drawing-app
Username for 'https://github.com': Kim-Min-Hyeok
Password for 'https://Kim-Min-Hyeok@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 314 bytes | 78.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Kim-Min-Hyeok/html5-canvas-drawing-app
e105f09..8bd9b8a master -> master
```

이렇게 수정한 내용을 내 github에 업데이트 하기위하여 git add, commit, push명령을 이용하여 업데이트 하였다.  
(커밋 메시지 = set title)

index.html set title 3 minutes ago

다음과 같이 github가 업데이트 되었다.

## Raspi초기세팅 + 웹서버세팅 + 가상호스트세팅까지 진행한 소감

처음에 라즈베리파이를 초기세팅하는 것이 제일 힘들었다. 아직 시작도 안했는데 초기세팅부터 이렇게 오류가 뜨면 어떻게 하지라는 생각이었다. 그러나 교수님께서 카톡방을 통해서 강의 외의 다양한 방법들을 제시해 주셨고, 그 중 나에게 가장 잘 맞는 것을 찾아서 세팅을 성공적으로 할 수 있었다. 나는 핫스팟을 이용하였는데, 이렇게 하면, 라즈베리의 ip주소를 잘 찾을 수 없었던 기숙사에서 효과적이었다. 그다음은 웹서버세팅이었다. 초기세팅에서 조금 애를 먹었던 탭인지 비교적 너무나도 수월하였다. 강의 그대로 차근차근 따라하면 오류없이 잘 되었다. 또한 그렇게 방법도 복잡하지 않아서 만들어지는 웹서버를 보며 쾌감을 느끼기도 하였다. 가상호스트도 세팅하였는데 물론 내가 여러 호스트들을 세팅하고 고객에게 서버를 제공하고 그 서버를 이용하여 고객의 입장에서 사이트를 만드는 것을 혼자 다 하였지만, 나중에 다른 사람들과 함께 이러한 서버를 운영한다면, 이라는 생각에 재미있었다. 또한 처음 웹사이트를 만드는 일은 그래도 처음하는 것이라 서툴기도 하였고, 그것을 포함한 모든 수업에서 그러했지만, 다행히 방에 같은 다른 분반의 같은 과목수업을 듣는 형이 있어 조금 진도가 빠른 내가 옆에서 도와주며 더 배우고 두번째 실습은 좀 더 빠르게 수행할 수 있었다.

### <잔디심기>

나는 계속해서 이 과목과 다른 과목의 과제들을 올리고 있다. (자바프로그래밍, 논리설계, 오픈소스소프트웨어 등)



물론 매일 하는 것이 바빠서 까먹을 때도 있지만, 그래도 내 잔디에 초록색이 조금씩 보이기 시작하니 더 생각하고 올릴 수 있다.

