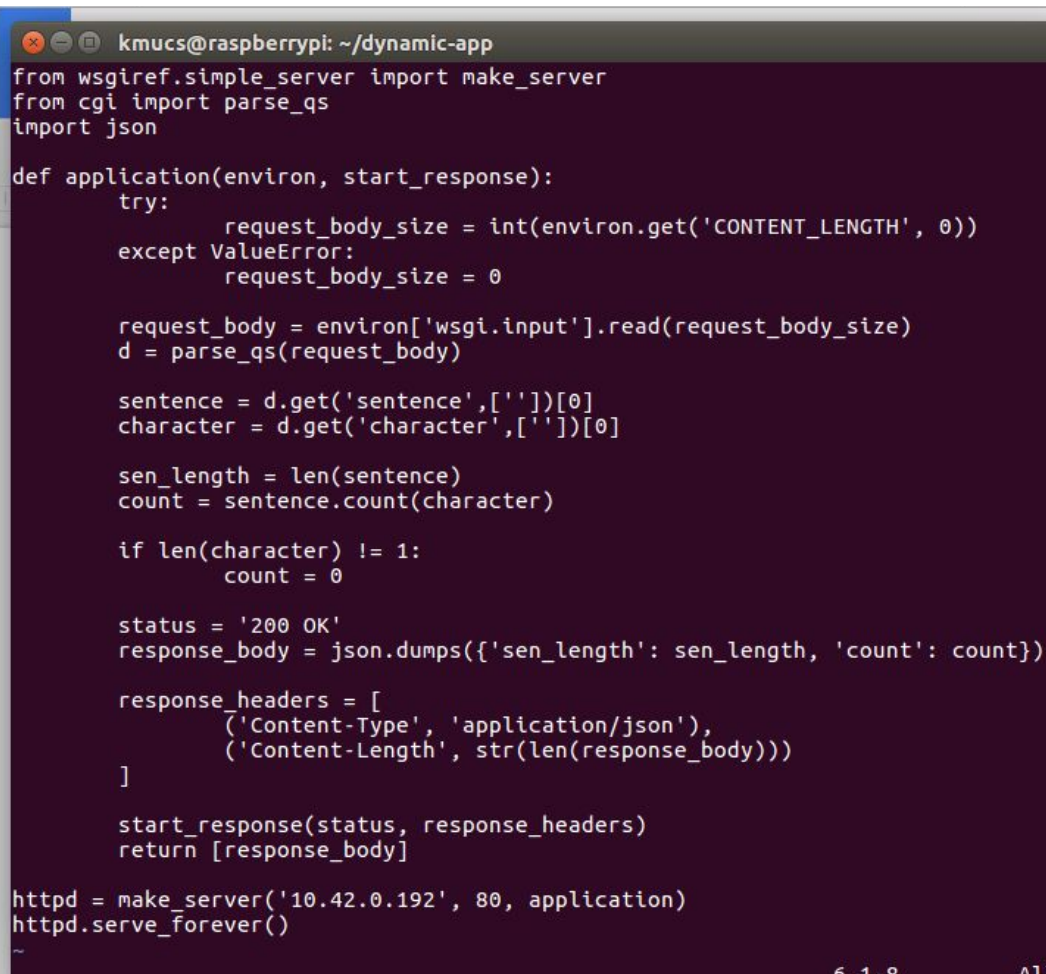


# 소프트웨어 프로젝트 1 - 기말 과제

소프트웨어학부  
20142753 이효준

## 1. 테스트 스크립트 만들기



```
kmucs@raspberrypi: ~/dynamic-app
from wsgiref.simple_server import make_server
from cgi import parse_qs
import json

def application(environ, start_response):
    try:
        request_body_size = int(environ.get('CONTENT_LENGTH', 0))
    except ValueError:
        request_body_size = 0

    request_body = environ['wsgi.input'].read(request_body_size)
    d = parse_qs(request_body)

    sentence = d.get('sentence', [''])[0]
    character = d.get('character', [''])[0]

    sen_length = len(sentence)
    count = sentence.count(character)

    if len(character) != 1:
        count = 0

    status = '200 OK'
    response_body = json.dumps({'sen_length': sen_length, 'count': count})

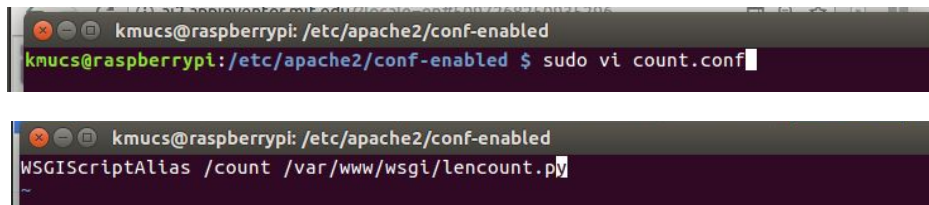
    response_headers = [
        ('Content-Type', 'application/json'),
        ('Content-Length', str(len(response_body)))
    ]

    start_response(status, response_headers)
    return [response_body]

httpd = make_server('10.42.0.192', 80, application)
httpd.serve_forever()
```

- 라이브러리, json import하기
  - 함수를 만들고 인자들을 받는다.
  - get 함수를 이용해 request\_body의 크기를 구해서 input값으로부터 request\_body를 구한다.
  - 앱을 통해 사용자에게 입력받은 문장과 character 정보가 있는 request\_body
  - String 타입 변수 d에 저장
  - get메소드를 이용해 문장과 character 값을 변수에 저장한다.
  - 문장의 길이를 구하고 count 함수를 이용해 count 값을 구한다. (길이가 1이 아닐경우 0)
  - json의 dumps 함수를 이용해 length, count 출력
  - response 헤더 구성
- (랜선을 사용해 스크린샷에 IP가 다르게 나왔습니다.)

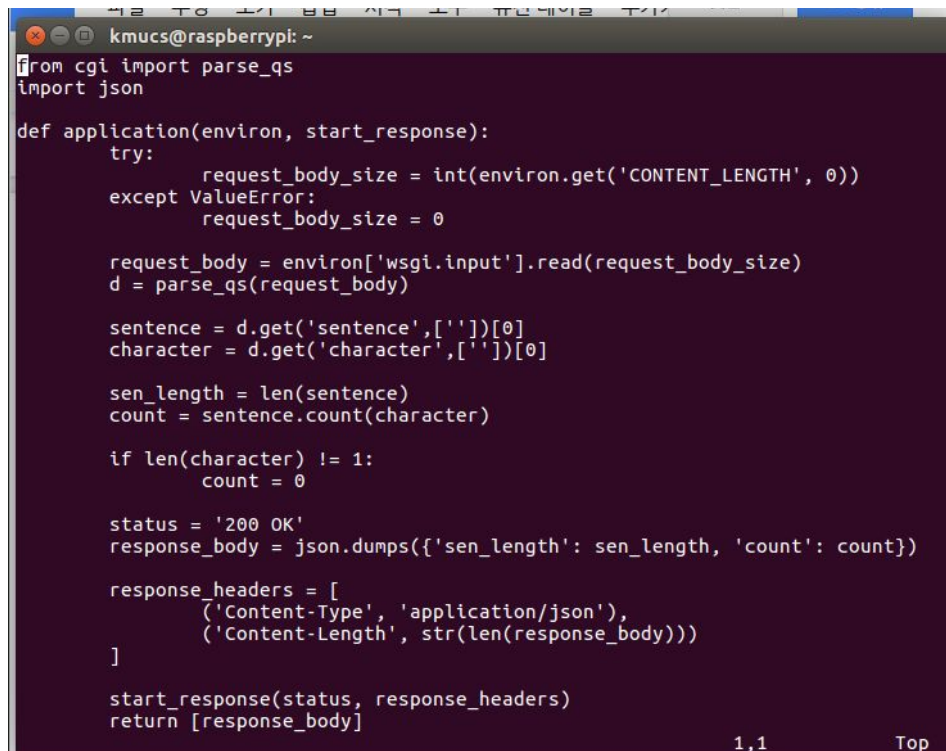
## 2.WSHI 스크립트 배치



```
kmucs@raspberrypi: /etc/apache2/conf-enabled
kmucs@raspberrypi:/etc/apache2/conf-enabled $ sudo vi count.conf

WSGIScriptAlias /count /var/www/wsgi/lencount.py
```

- Apache 웹서버가 시동할 경우 이 파일을 읽음
- /count에 대한 요청은 /var/www/wsgi/lencount.py를 실행하도록 함



```
from cgi import parse_qs
import json

def application(environ, start_response):
    try:
        request_body_size = int(environ.get('CONTENT_LENGTH', 0))
    except ValueError:
        request_body_size = 0

    request_body = environ['wsgi.input'].read(request_body_size)
    d = parse_qs(request_body)

    sentence = d.get('sentence', [''])[0]
    character = d.get('character', [''])[0]

    sen_length = len(sentence)
    count = sentence.count(character)

    if len(character) != 1:
        count = 0

    status = '200 OK'
    response_body = json.dumps({'sen_length': sen_length, 'count': count})

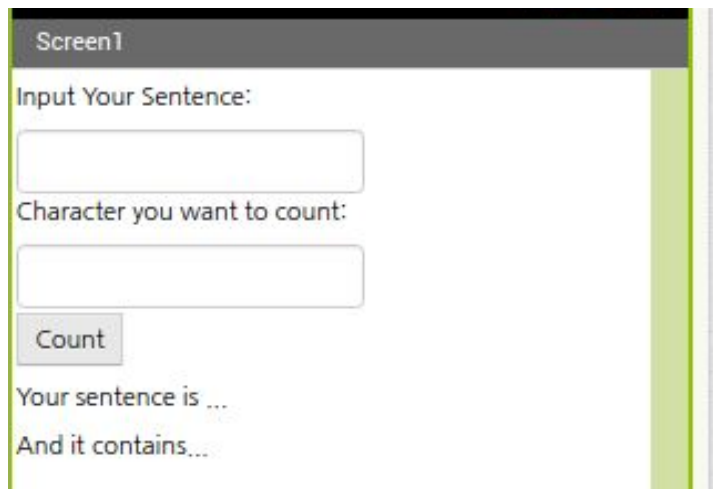
    response_headers = [
        ('Content-Type', 'application/json'),
        ('Content-Length', str(len(response_body)))
    ]

    start_response(status, response_headers)
    return [response_body]
```

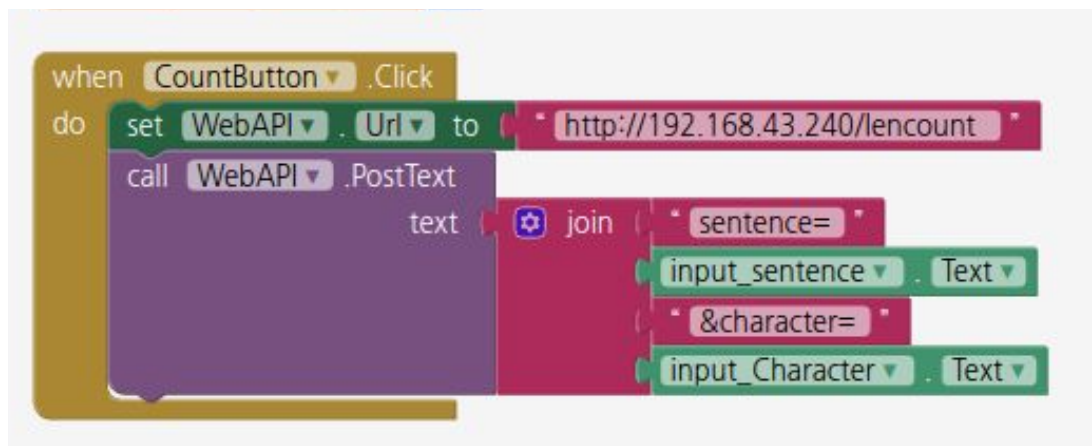
- 첫 테스트 스크립트에서 wsgi 서버를 import하는 문장과, 맨 아래 서버를 만드는 문장을 삭제
- 직접 실행하지 않고, Apache 웹 서버에서 스크립트를 실행하기 위해

- URL을 작성할 때 /lencount 를 통해 실행하도록 함께 작성해주어야 함

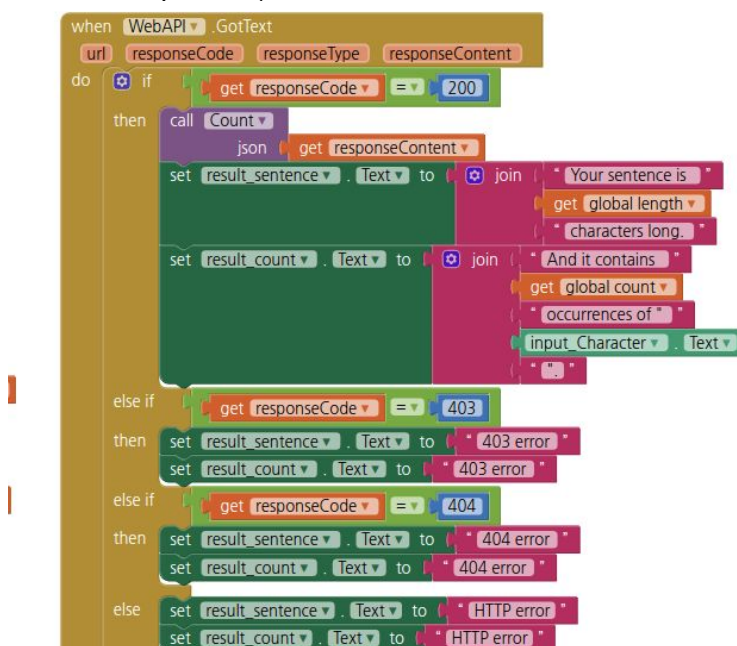
### 3. UI 디자인



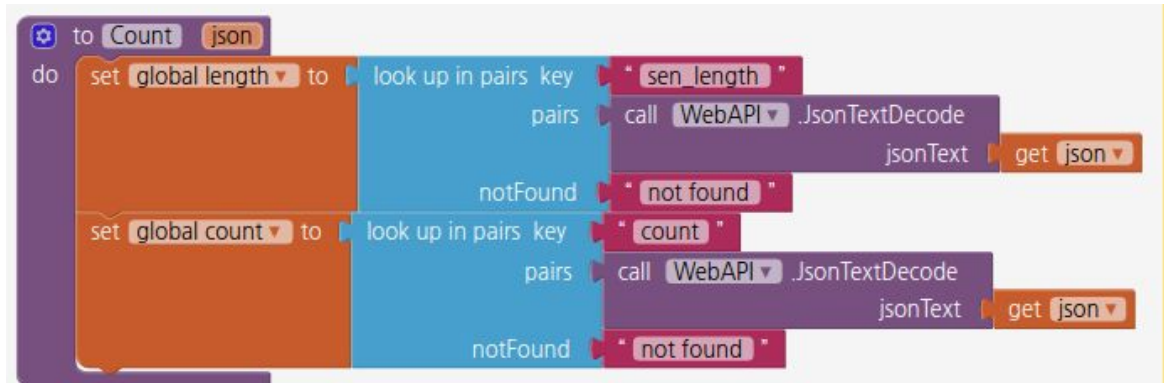
### 4. 블록코딩



- count 버튼을 클릭할 경우 먼저 URL 세팅 (주소는 테스트 스크립트에서와 같음)
- text를 web post함 (text는 "sentence= ...&character=..." 형태)



- 웹에서 Text를 받을 경우
- 요청이 성공한 경우 (responseCode ==200)
- responseContent 를 받아와 json 코드로 구성



- json 데이터를 받아 키에 해당하는 sen\_length, count와 그에 대한 값들로 구성된 리스트들에서 값을 가져와 전역변수인 length와 count에 저장
- 저장된 변수들을 결과 label로 출력
- (요청이 성공하지 않았을 경우의 error 메시지를 출력)