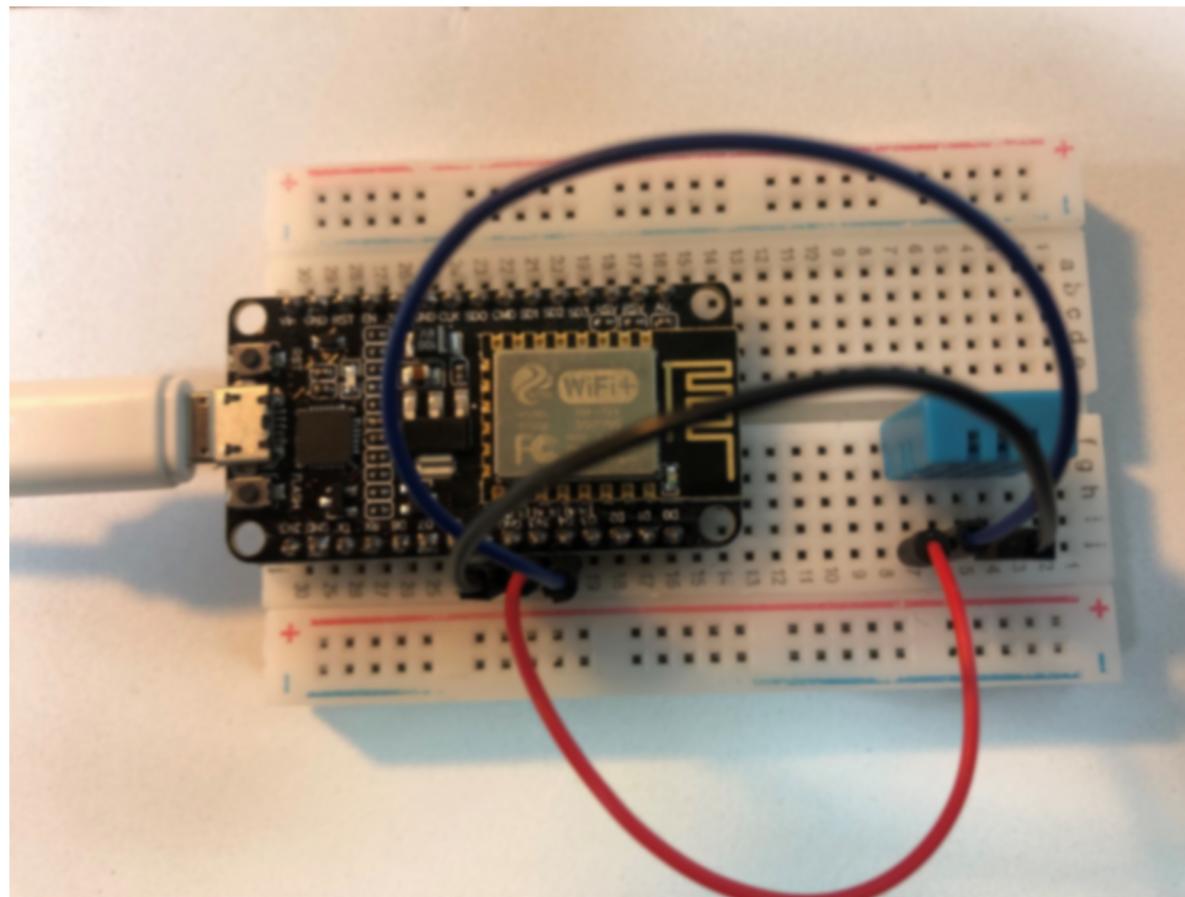


# NodeMCU를 이용한 간단한 웹서버 구현 :

웹 기반 온습도 측정 장치 만들기

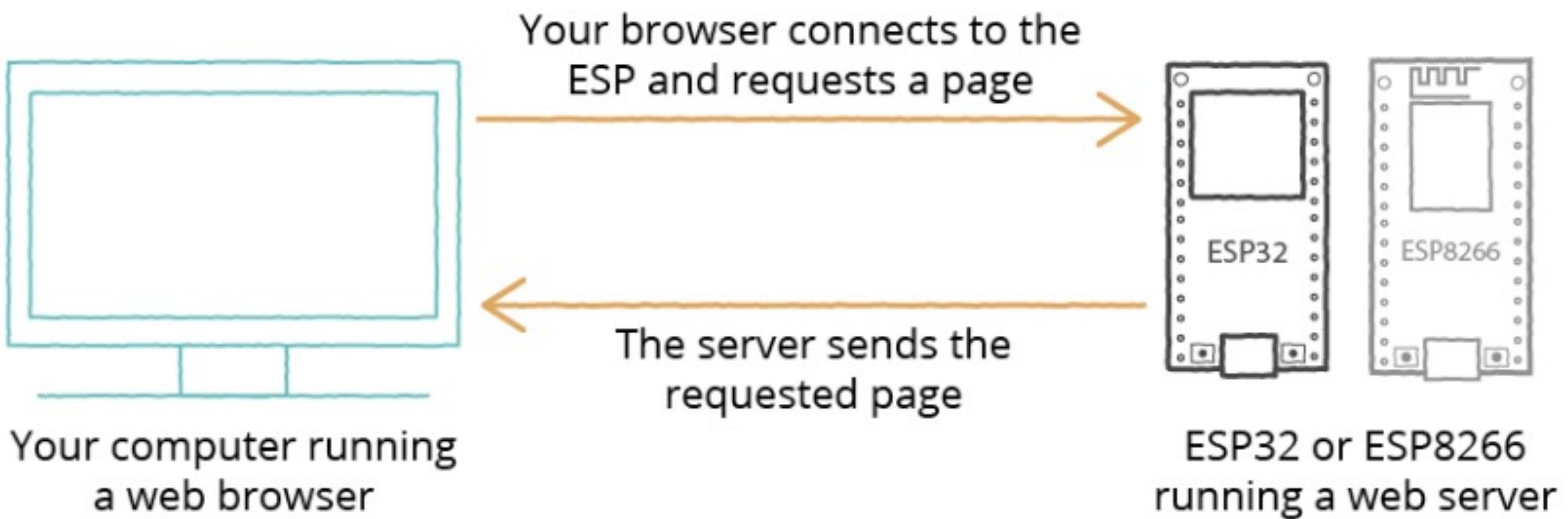
# 회로 구성



# 작동 구조 : Web server - client



# 작동 구조 : Web server - client



# 아두이노 WiFi - Server

```
File Edit Sketch Tools Help
Server_basic §
1
2
3
4
5 void setup() {
6
7
8
9
10 }
11
12 void loop() {
13
14
15
16
17 }
```

- 1번에는 헤더 파일 포함, 네트워크 접속 정보 외 각종 선언들이 들어가고,
- 2번은 setup() 함수로 WiFi 연결을 설정.
- 3번은 loop() 함수에서 Client 접속이 들어올 때마다 웹페이지를 전송.

# WiFi 라이브러리, 클래스

- 라이브러리 사용 : #include <WiFi.h>
- WiFi 클래스
  - begin(), disconnect(), config(), status(), getSocket()....
- IPAddress 클래스
  - localIP(), subnetMask()...
- Server 클래스
  - Server, WiFiServer(), begin(), available(), write(), print(), println()
- Client 클래스
  - Client, WiFiClient(), connected(), connect(), available(), read(), flush()...
- 참고 문서 : <https://www.arduino.cc/en/Reference/WiFi>

# void setup() 구현

```
1 #include <ESP8266WiFi.h>
2 #include "DHT.h"
3 const char* ssid = "melon"; // 와이파이 공유기의 SSID
4 const char* password = "deitcs3217"; // 와이파이 공유기
5 #define DHTPIN D4      // what pin we're connected to
6 #define DHTTYPE DHT11    // DHT 11
7 DHT dht(DHTPIN, DHTTYPE);
8
```

```
11 void setup()
12 {
13     Serial.begin(9600);
14
15     // 와이파이 네트워크에 접속
16     Serial.print("Connecting to ");
17     Serial.println(ssid);
18     WiFi.begin(ssid, password);
19     dht.begin();
20
21     while (WiFi.status() != WL_CONNECTED) {
22         delay(500);
23         Serial.print(".");
24     }
25     Serial.println("WiFi connected");
26
27     // 웹서버 개시
28     server.begin();
29     Serial.println("HTTP server started");
30
31     // 접속할 IP 주소 표시
32     Serial.print("Use this URL : ");
33     Serial.print("http://");
34     Serial.print(WiFi.localIP());
35     Serial.println("/");
36 }
```

# WiFi.status()

- WL\_CONNECTED : 공유기에 성공적으로 연결이 된 후
  - WL\_IDLE\_STATUS : WiFi 상태가 변동되는 중간 과정
  - WL\_NO\_SSID\_AVAIL : 사용자명 SSID 에 연결 될 수 없을 때
  - WL\_CONNECT\_FAILED : 패스워드가 부정확할 경우
- WL\_DISCONNECTED : 와이파이 모듈이 스테이션 모드로 배열되지 못한 경우

```

38 void loop()
39 {
40     // 클라이언트가 접속되었는지 확인
41     WiFiClient client = server.available();
42
43     if (!client) {
44         return; // 함수 실행 종료
45     }
46
47     // 클라이언트가 데이터를 보낼 때까지 기다림
48     Serial.println("new client");
49     while(!client.available()){
50         delay(1);
51     }
52
53     // 클라이언트 리퀘스트의 첫번째 줄을 읽어들임. http 요청
54     String request = client.readStringUntil('\n');
55     Serial.println(request);
56     client.flush();                                Client request 읽기
57
58     // 클라이언트 리퀘스트에 대한 응답 메시지를 시리얼통신으로 클라이언트에 전송
59     client.println("HTTP/1.1 200 OK");
60     client.println("Content-Type: text/html");
61     client.println("Connection: close");
62     client.println();                               Http response Header 보내기
63     client.println("<!DOCTYPE HTML>");           Message 내용 보내기
64     client.println("<html>");
65     client.println("<meta http-equiv=\"refresh\" content=\"5\">" );
66     client.println("<br />");                     Message 내용 보내기
67     client.println(""); // do not forget this one
68

```

# void main() 구현

Message 내용 보내기

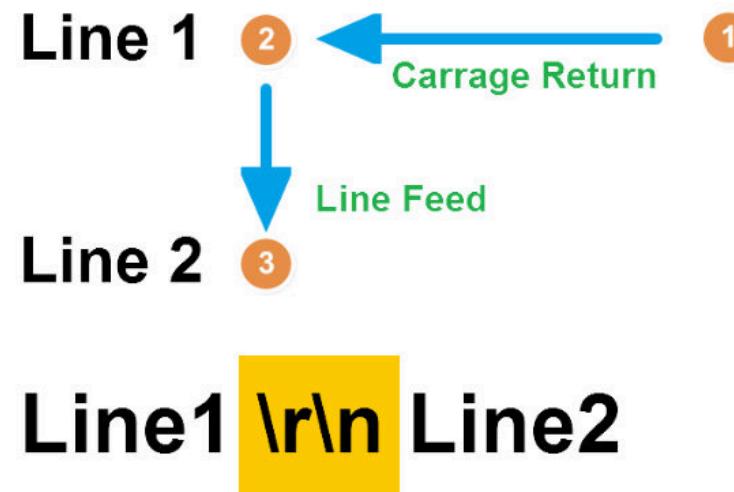
```

69     float h = dht.readHumidity();
70     float t = dht.readTemperature();
71     float f = dht.readTemperature(true);
72
73     client.print("<h2>Weather Station</h2>");
74     client.print("Temperature (C) : ");
75     client.println(t);
76     client.println("<p />");
77
78     client.print("Humidity (%) : ");
79     client.println(h);
80     client.println("<p />");
81
82     client.println("</html>");                   Message 내용 보내기
83     delay(1);
84     Serial.println("Client disconnected");
85 }

```

# 클라이언트 Request

- 웹서버에 http 요청 : 예) http://10.0.1.15
- String request = client.readStringUntil('\n');
  - 클라이언트 Request 문자열이 빈줄(\r 혹은 \n)일 때까지 읽음
- CR, \r, Carriage Return : 그 줄의 처음으로 커저를 옮김
- LF, \n, Line Feed : 다음 줄로 커저를 내림



# HTTP Response 형식

HTTP/1.1 200 OK

Status Line

Content-type:text/html

Header

.....

empty line

빈 줄

```
<!DOCTYPE html>
<html>
    <head> </head>
    <body> </body>
</html>
```

Message Body

# 시리얼 모니터

```
COM4
[전송]
↓↓↓
Connecting to melon
..WiFi connected
Server started
Use this URL : http://10.0.1.15/

```

```
COM4 - □ × 전송  
[§]sConnecting to melon  
.WiFi connected  
Server started  
Use this URL : http://10.0.1.15/  
new client  
GET / HTTP/1.1  
Client disconnected  
new client  
GET /favicon.ico HTTP/1.1  
Client disconnected  
new client  
GET / HTTP/1.1  
Client disconnected  
new client  
GET /favicon.ico HTTP/1.1  
Client disconnected  
new client  
GET / HTTP/1.1  
Client disconnected  
new client  
GET /favicon.ico HTTP/1.1  
Client disconnected  
new client  
GET / HTTP/1.1  
Client disconnected  
new client  
GET / HTTP/1.1  
Client disconnected  
[§]sClient disconnected  
 자동 스크롤  타이머스탬프 표시 새 줄 9600 보드레이트 출력 지우기
```

< WiFi 접속 성공, 서버 동작 시작 >

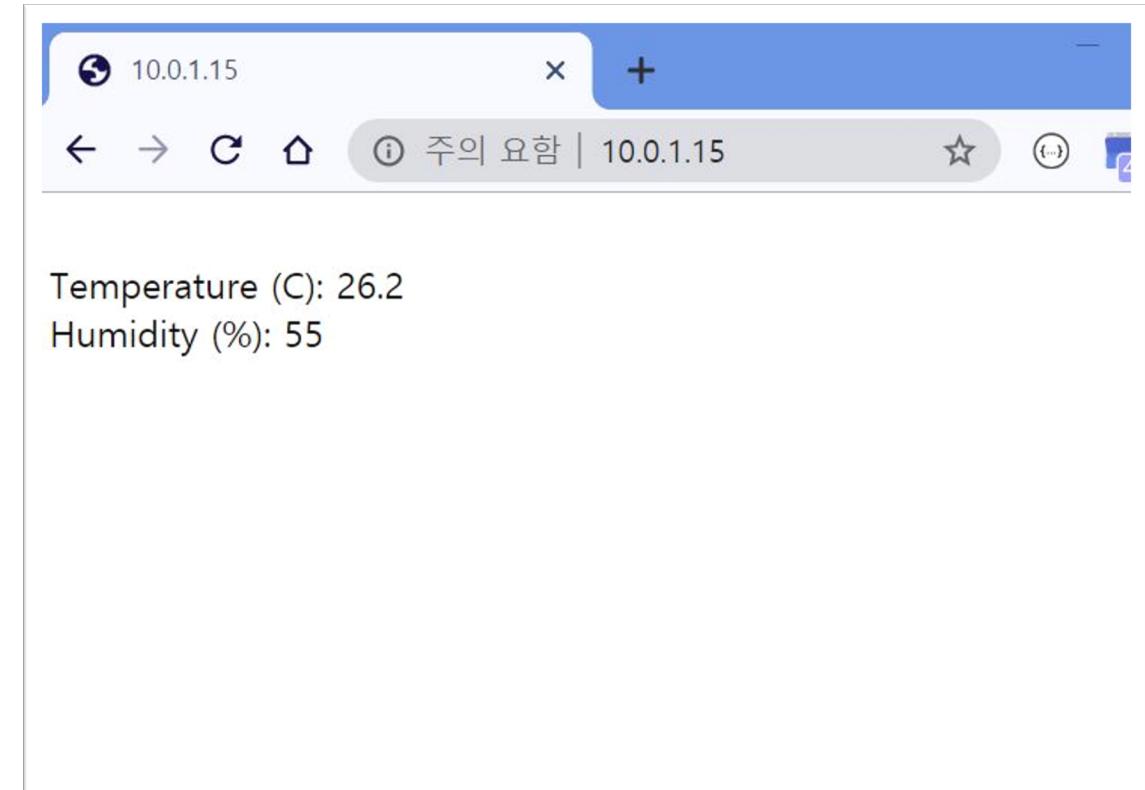
< 클라이언트 request, 서버 response>

COM4

```
Connecting to melon
..WiFi connected
Server started
Use this URL : http://10.0.1.15/
new client
GET / HTTP/1.1
Client disconnected
new client
GET /favicon.ico HTTP/1.1
Client disconnected
new client
GET / HTTP/1.1
Client disconnected
new client
GET /favicon.ico HTTP/1.1
Client disconnected
new client
GET /favicon.ico HTTP/1.1
Client disconnected
new client
GET / HTTP/1.1
Client disconnected
new client
GET /favicon.ico HTTP/1.1
Client disconnected
new client
GET / HTTP/1.1
Client disconnected
new client
GET /favicon.ico HTTP/1.1
Client disconnected
new client
GET / HTTP/1.1
Client disconnected
new client
GET / HTTP/1.1
Client disconnected
```

자동 스크롤  타임스탬프 표시 새 콜 9600 보드레이트 출력 지우기

URL: http://10.0.1.15



< 클라이언트 request, 결과 >

< WiFi 접속 성공, 웹 서버 response >

# 참고 소스 코드

- <https://github.com/IoT-Lab-02/Week06-Lab>

# 과제

- 수업 내용을 참고하여 자신의 웹 기반 온습도 측정기를 제작하시오.
  - 참고 소스 코드를 이해하고 공부하여, modify 혹은 추가하면 좋다
- W-Fi는 공유기 혹은 스마트폰 핫스팟을 이용
- 제출물
  - 소스 코드
  - 결과물의 스마트폰 촬영 동영상
    - 본인 확인용 얼굴도 일부 포함
    - 작동 영상(웹 브라우저, 회로 등)

# 다음주 내용 : 웹 서버를 이용한 LED ON/OFF 하기

