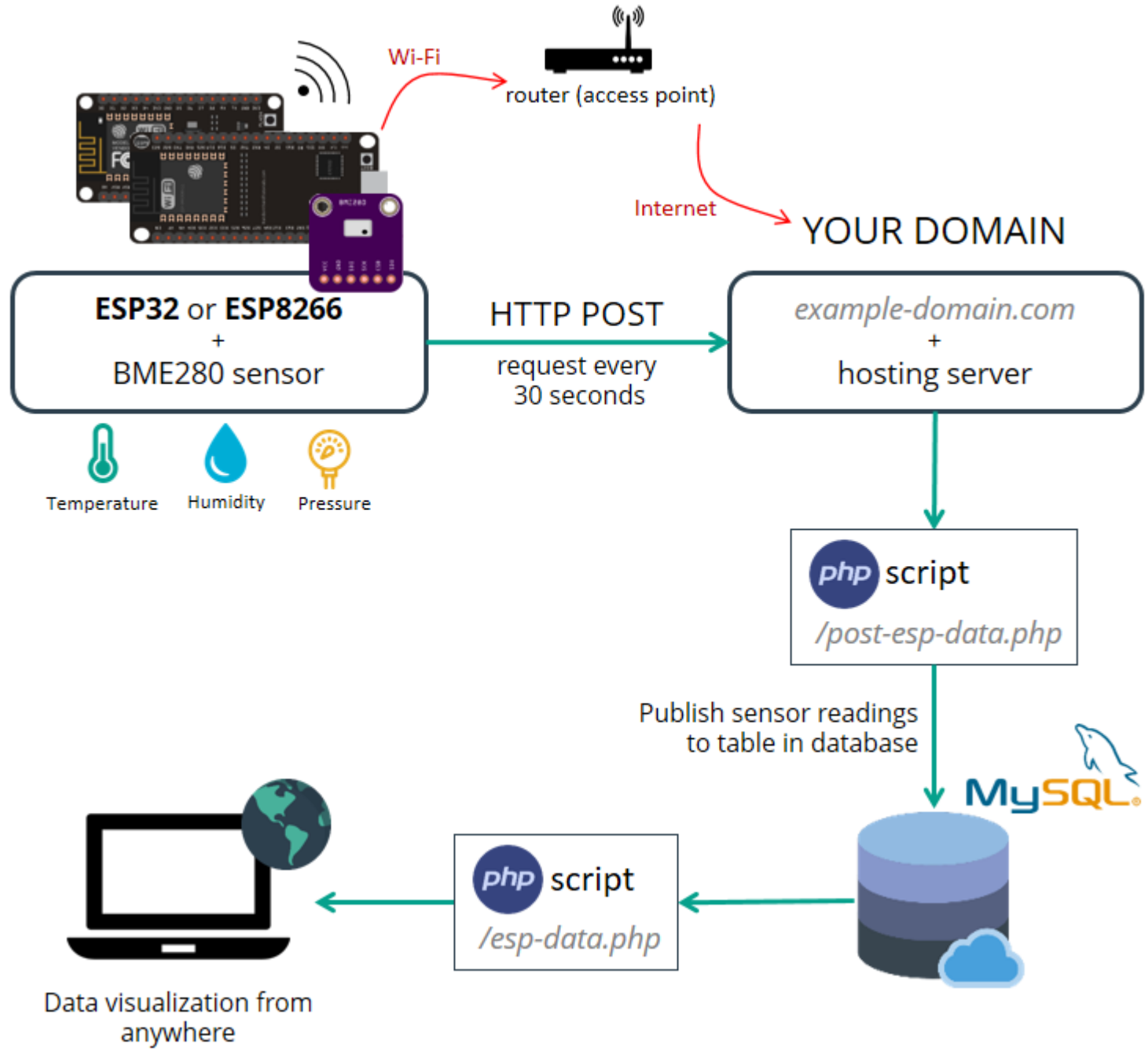
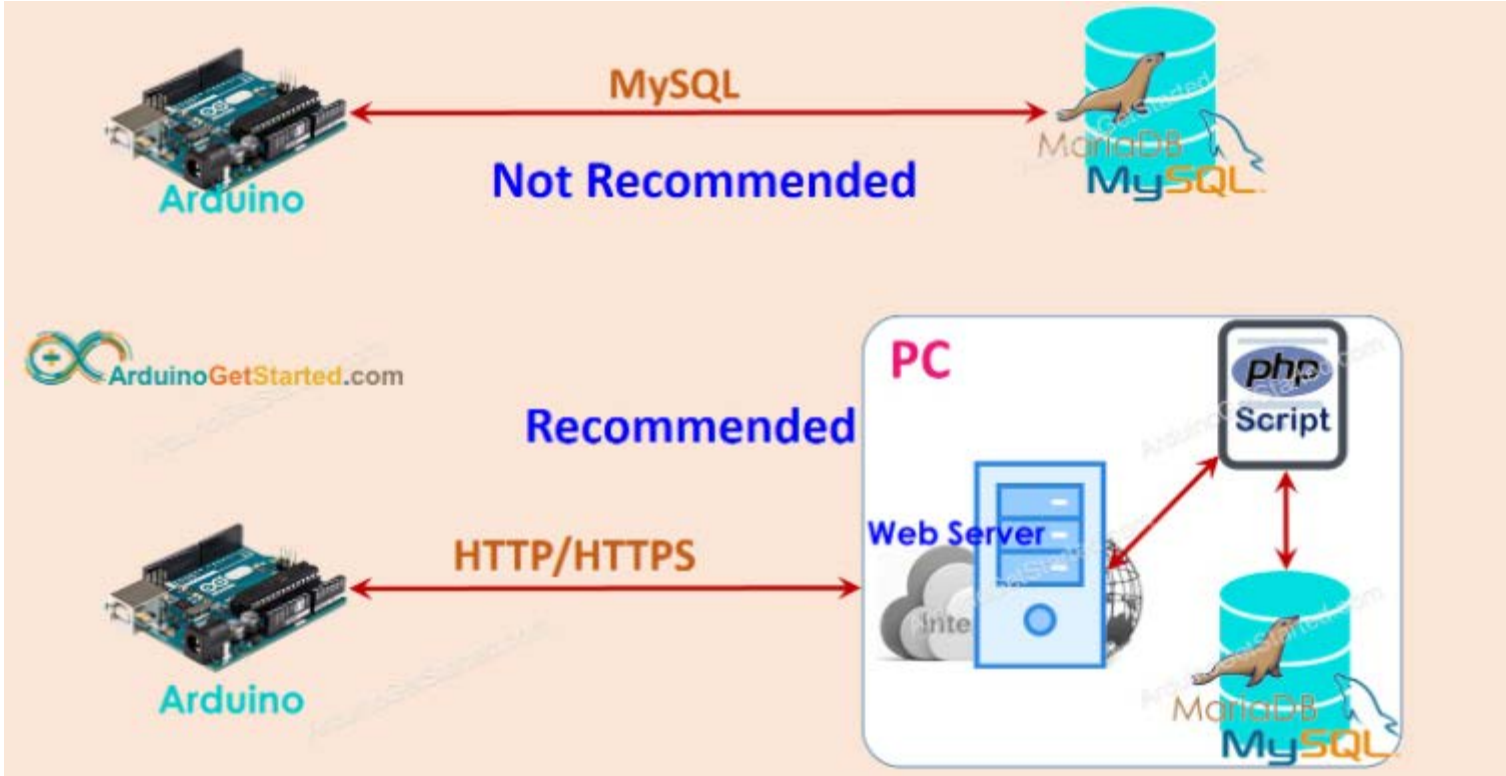
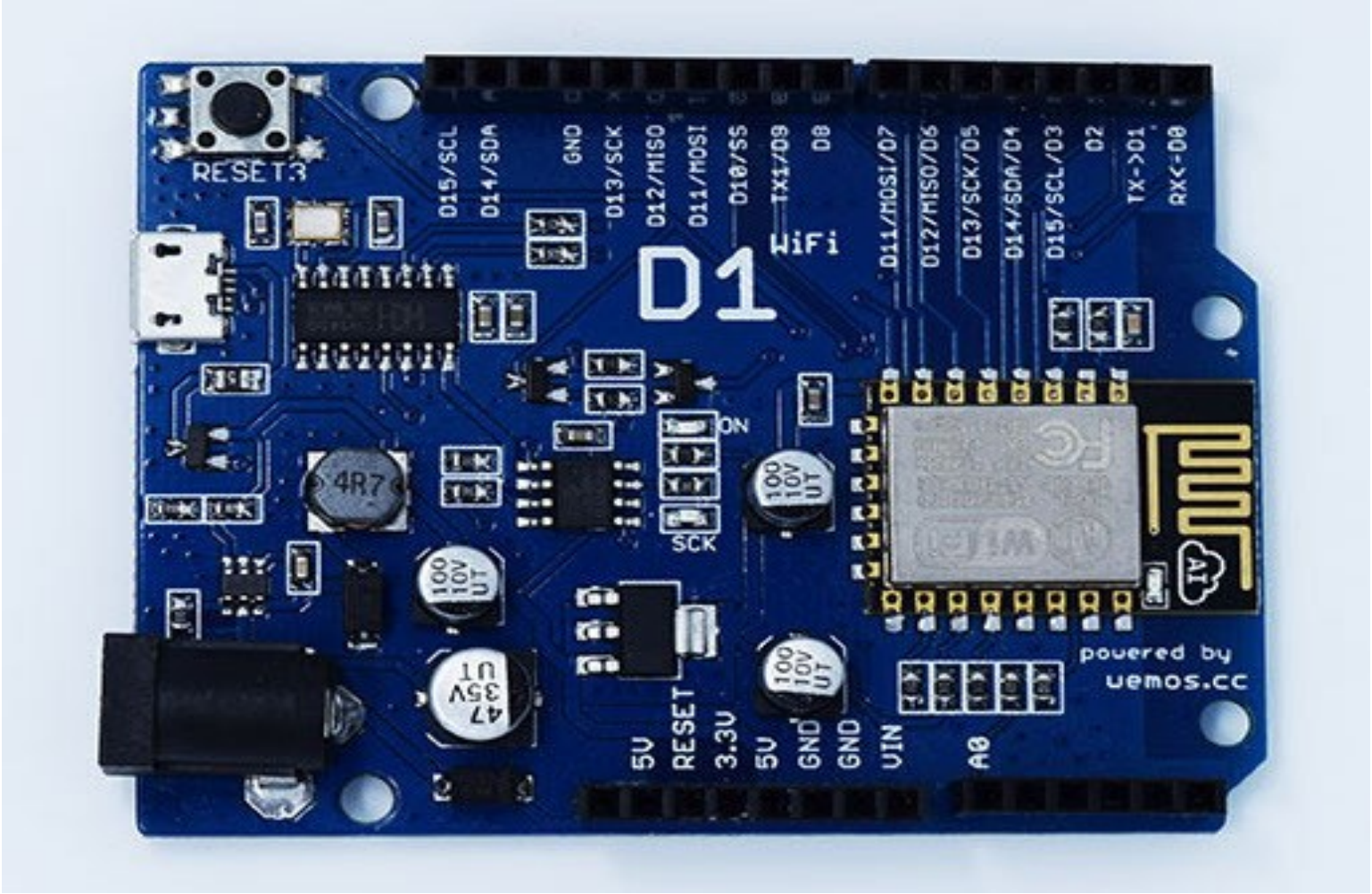
SESSION 4: I.O.T. Wifi

Advanced Arduino IDE and Esp8266


Wi-Fi
router (access point)
Internet
YOUR DOMAIN
ESP32 or ESP8266
+
BME280 sensor
HTTP POST
request every
30 seconds
example-domain.com
+
hosting server
Temperature
Humidity
Pressure
php script
/post-esp-data.php
Publish sensor readings
to table in database
MySQL
php script
/esp-data.php
Data visualization from
anywhere


MySQL
Arduino
Not Recommended
MariaDB
MySQL
ArduinoGetStarted.com
Recommended
PC
php
Script
Web Server
HTTP/HTTPS
Arduino
MariaDB
MySQL

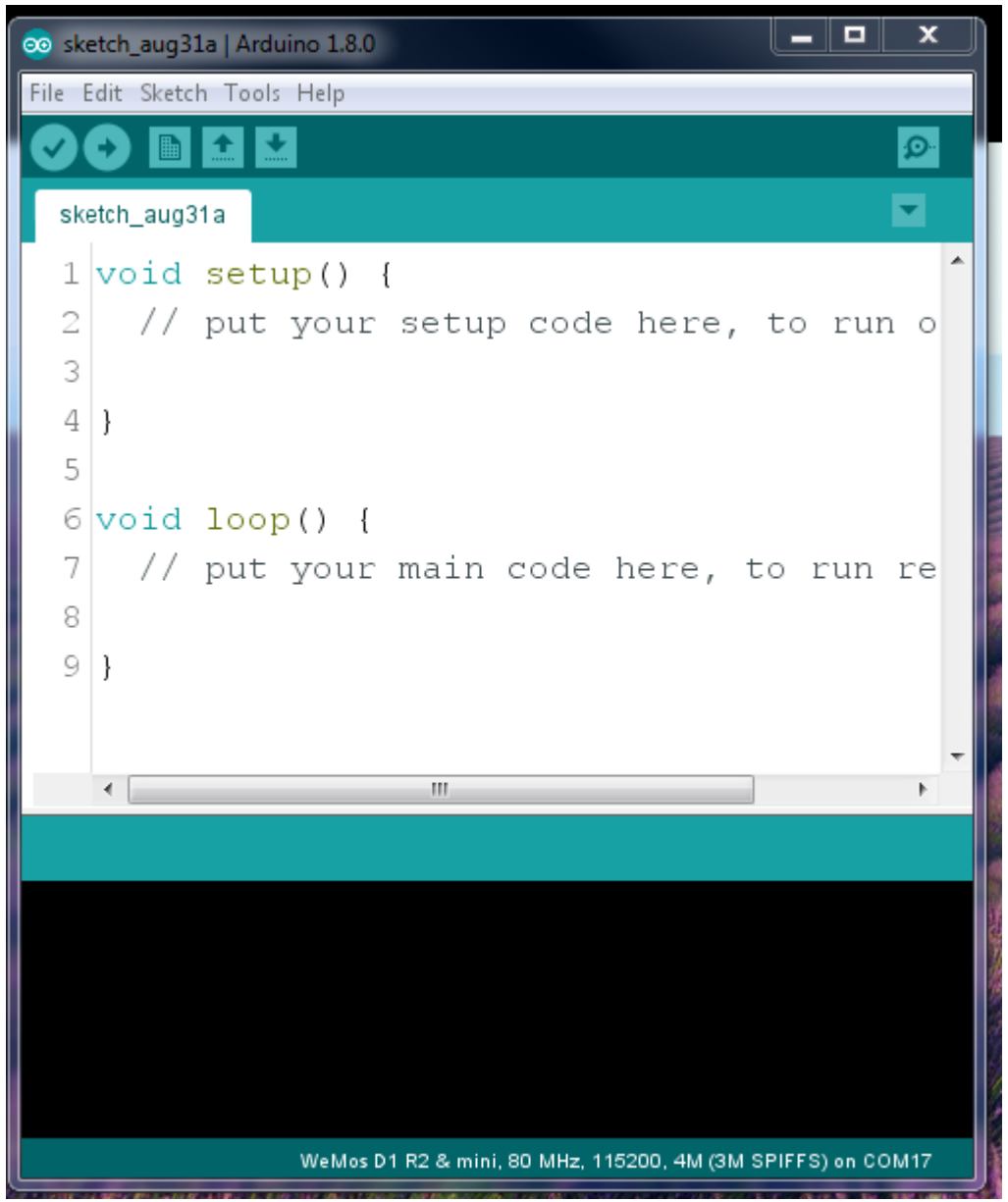
Wemos D1 ESP8266 based board



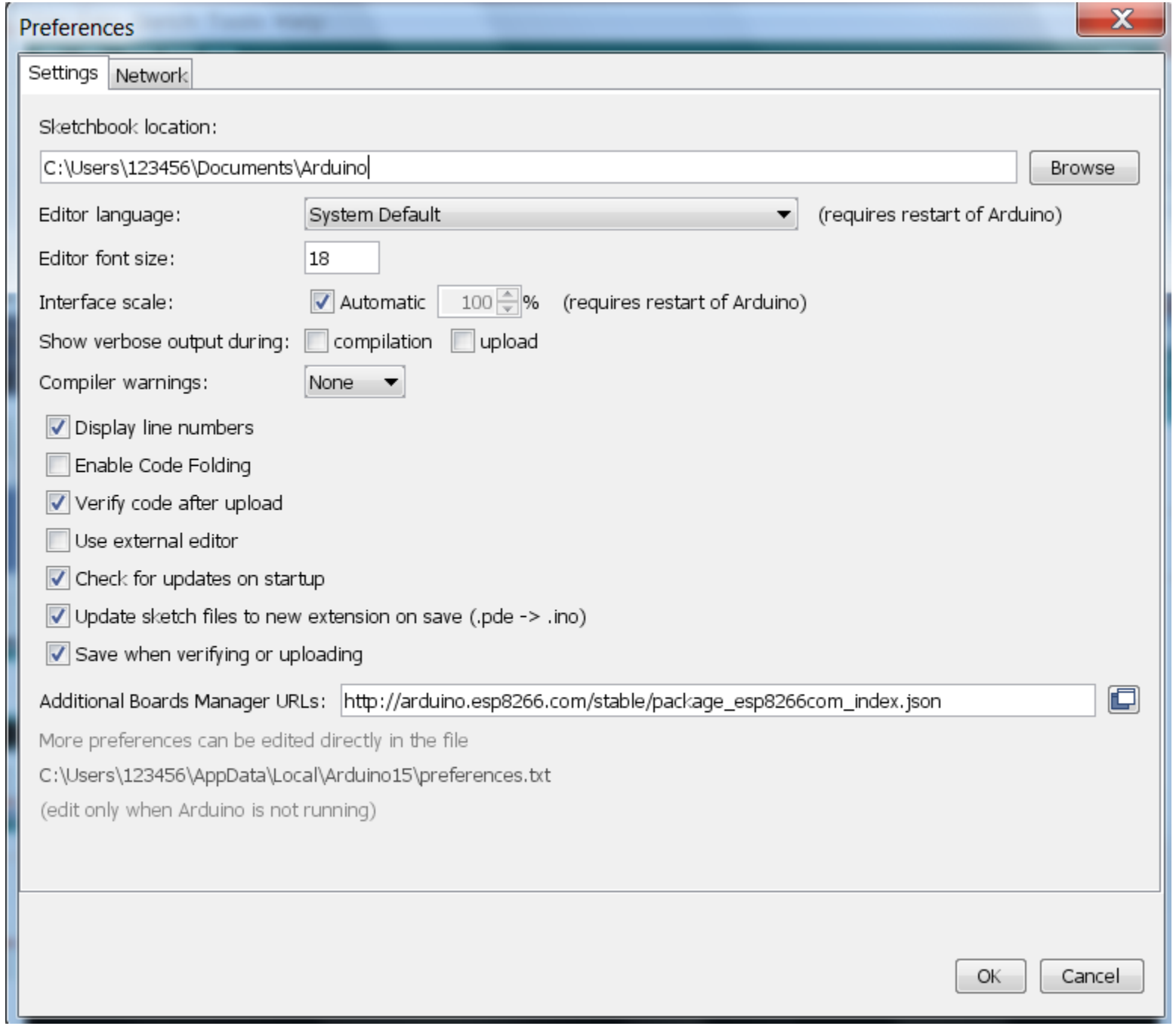
All GPIO work with 3.3 voltages

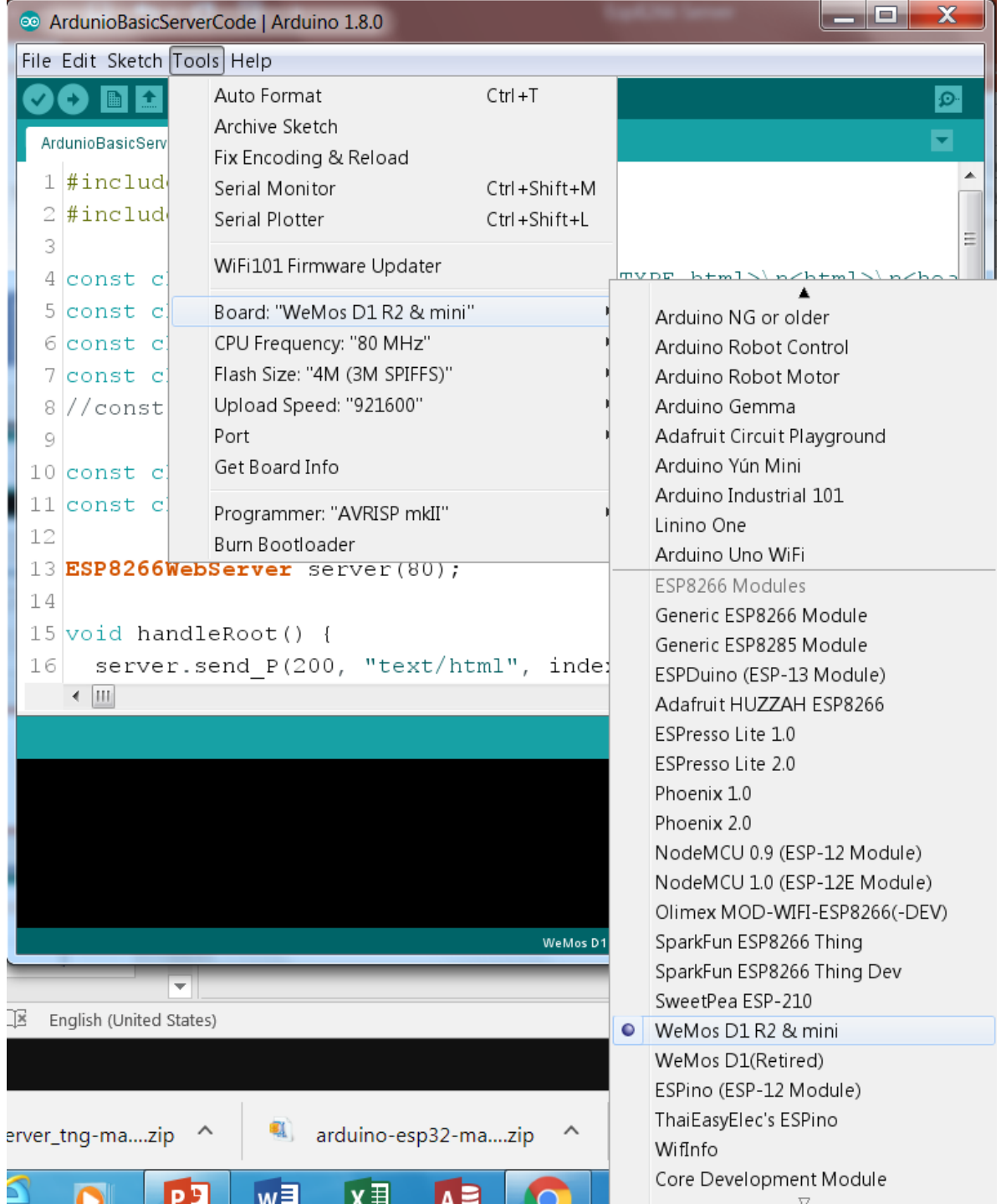
Pin (On Board)	Function	ESP-8266 Pin (Coding)
D0	RX	GPIO3
D1	TX	GPIO1
D2	IO	GPIO16
D3(D15)	IO,SCL	GPIO5
D4(D14)	IO,SDA	GPIO4
D5(D13)	IO,SCK	GPIO14
D6(D12)	IO,MISO	GPIO12
D7(D11)	IO,MOSI	GPIO13
D8	IO,Pull-up	GPIO0
D9	IO,pull-up, BUILTIN_LED	GPIO2
D10	IO,pull-down,SS	GPIO15
A0	Analog Input	A0

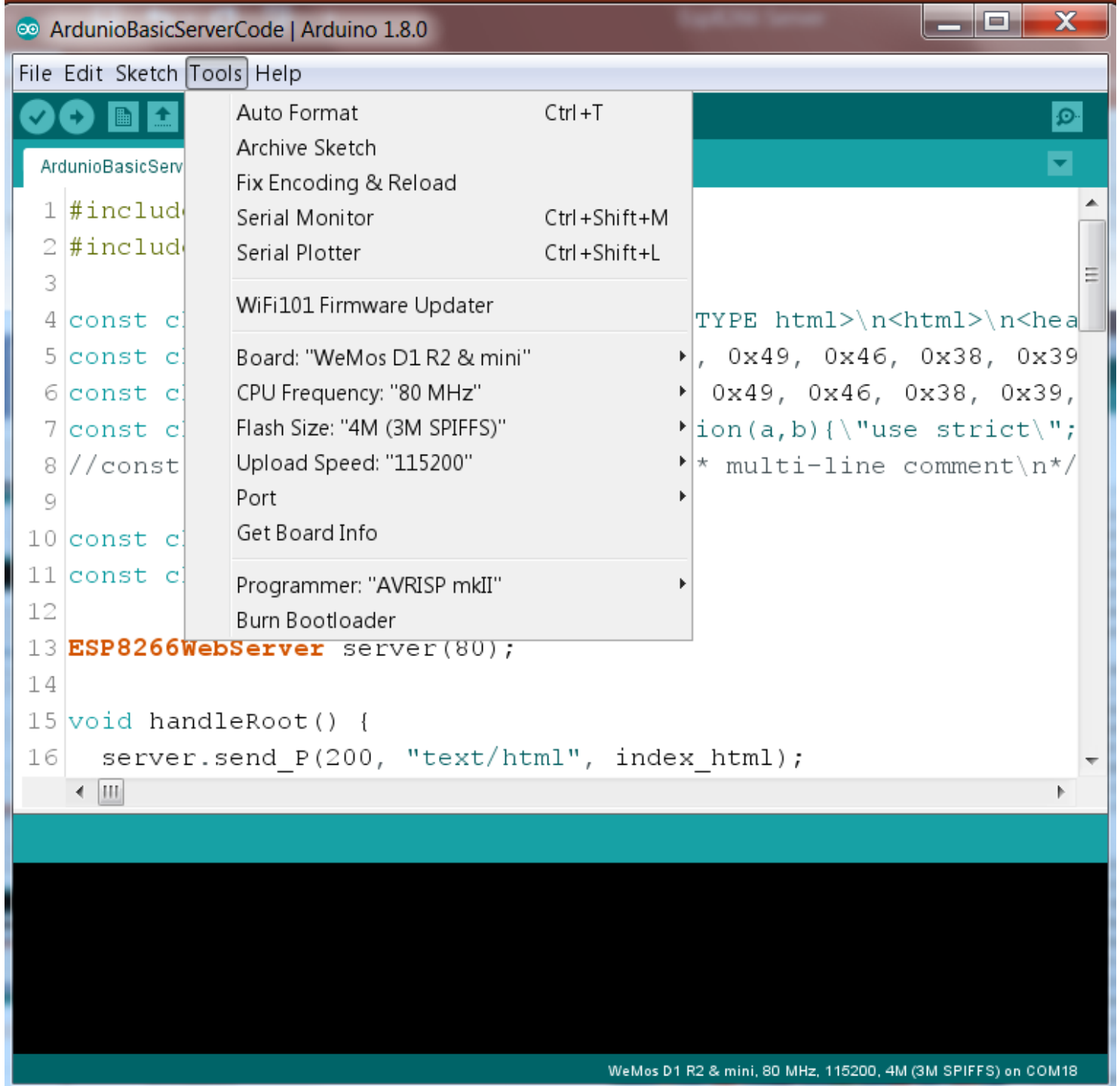
* All digital i/o can performance PWM output

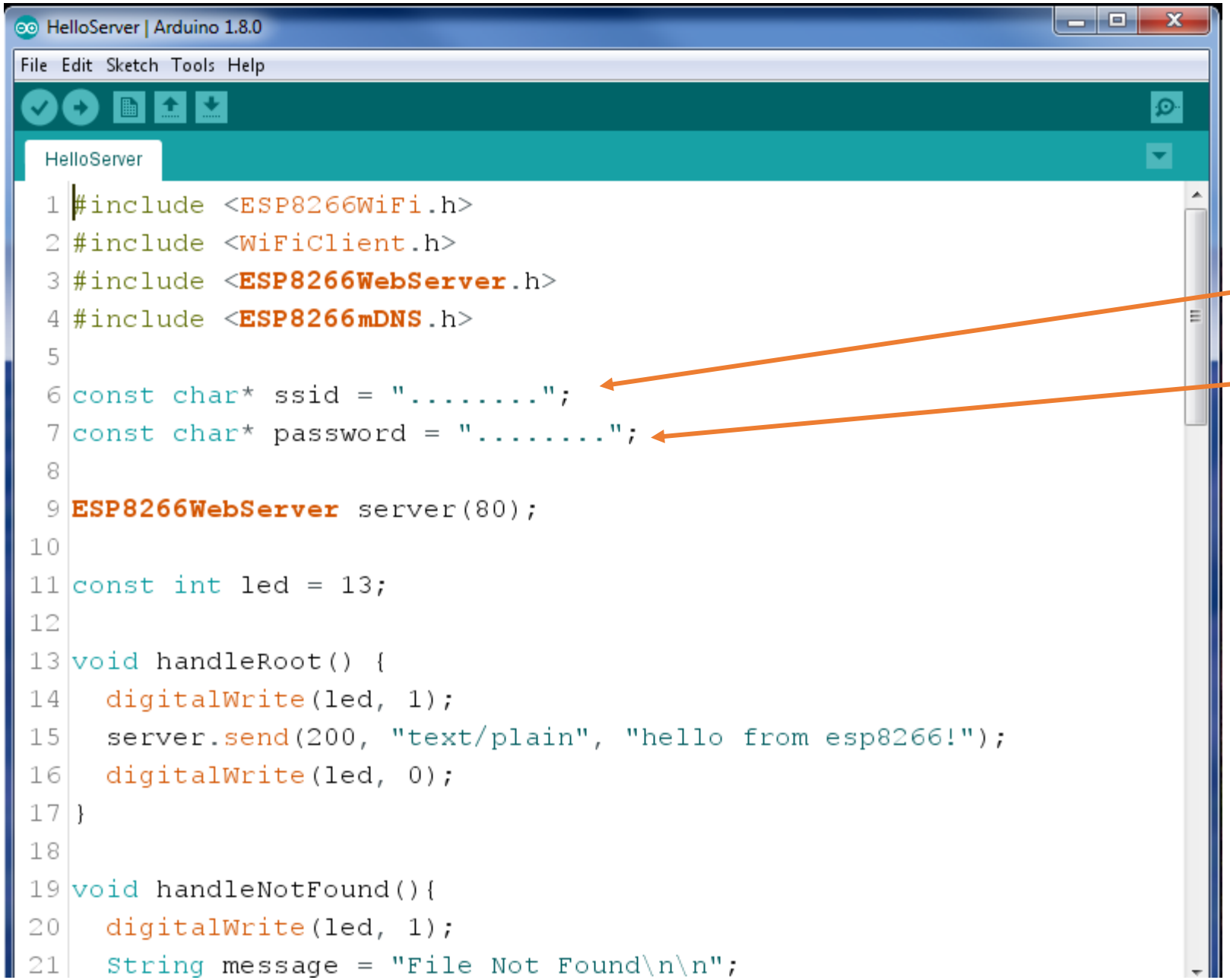


File → Examples → ESP8266WebServer → HelloServer


Preferences
Settings
Network
Sketchbook location:
C:\Users\123456\Documents\Arduino
Browse
Editor language:
System Default
(requires restart of Arduino)
Editor font size:
18
Interface scale:
Automatic
100
%
(requires restart of Arduino)
Show verbose output during:
compilation
upload
Compiler warnings:
None
Display line numbers
Enable Code Folding
Verify code after upload
Use external editor
Check for updates on startup
Update sketch files to new extension on save (.pde -> .ino)
Save when verifying or uploading
Additional Boards Manager URLs:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
More preferences can be edited directly in the file
C:\Users\123456\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)
OK
Cancel


ArdunioBasicServerCode | Arduino 1.8.0
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "WeMos D1 R2 & mini"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "921600"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
ESP8266WebServer server(80);
void handleRoot() {
server.send_P(200, "text/html", inde
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino Yún Mini
Arduino Industrial 101
Linino One
Arduino Uno WiFi
ESP8266 Modules
Generic ESP8266 Module
Generic ESP8285 Module
ESPDuino (ESP-13 Module)
Adafruit HUZZAH ESP8266
ESPresso Lite 1.0
ESPresso Lite 2.0
Phoenix 1.0
Phoenix 2.0
NodeMCU 0.9 (ESP-12 Module)
NodeMCU 1.0 (ESP-12E Module)
Olimex MOD-WIFI-ESP8266(-DEV)
SparkFun ESP8266 Thing
SparkFun ESP8266 Thing Dev
SweetPea ESP-210
WeMos D1 R2 & mini
WeMos D1(Retired)
ESPino (ESP-12 Module)
ThaiEasyElec's ESPino
WifInfo
Core Development Module
WeMos D1
English (United States)
erver_tng-ma....zip
arduino-esp32-ma....zip

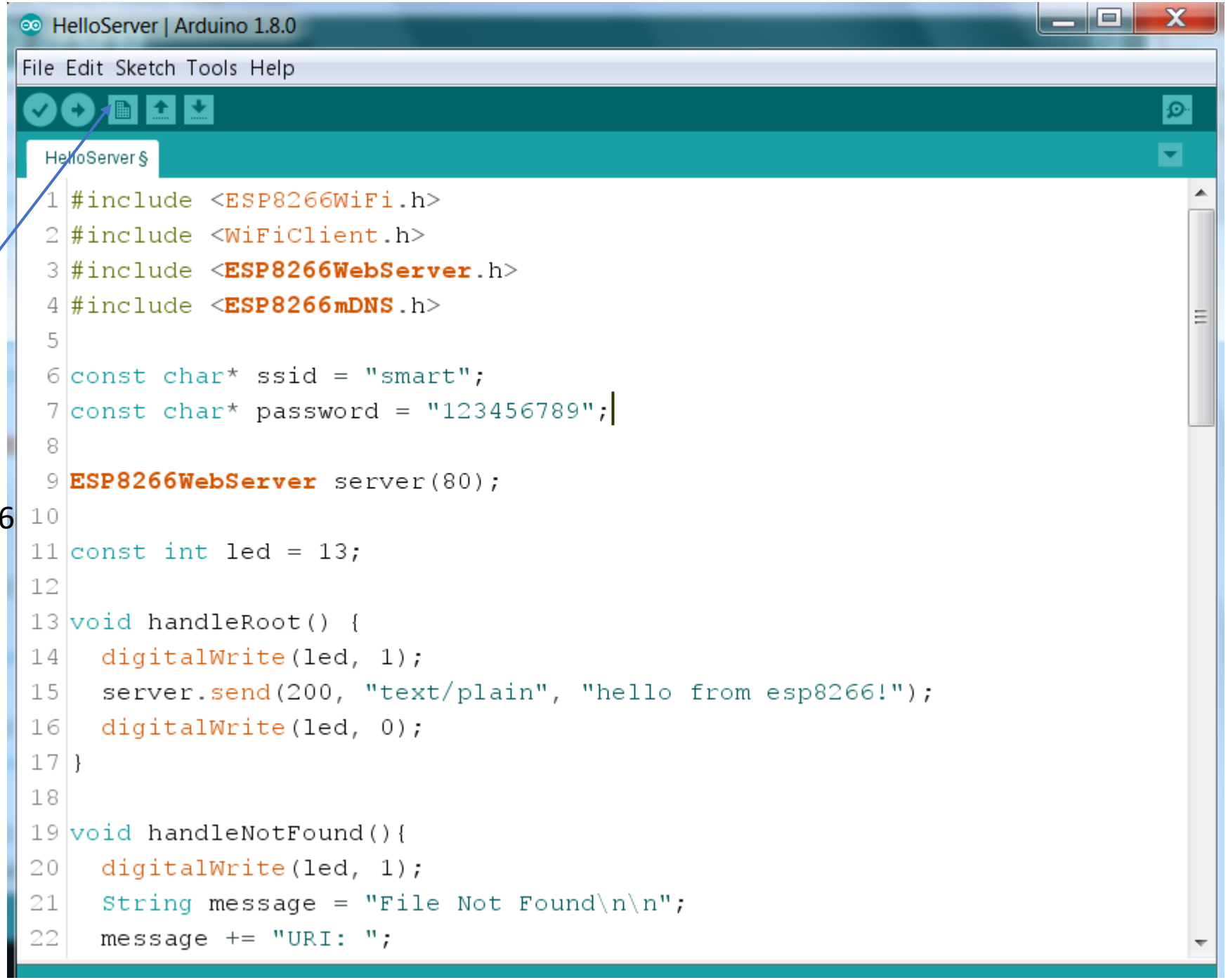

ArdunioBasicServerCode | Arduino 1.8.0
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "WeMos D1 R2 & mini"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
ArdunioBasicServ
1 #includ
2 #includ
3
4 const c
5 const c
6 const c
7 const c
8 //const
9
10 const c
11 const c
12
13 ESP8266WebServer server(80);
14
15 void handleRoot() {
16 server.send_P(200, "text/html", index_html);
TYPE html>\n<html>\n<hea
, 0x49, 0x46, 0x38, 0x39
0x49, 0x46, 0x38, 0x39,
ion(a,b){\"use strict\";
* multi-line comment\n*/
WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS) on COM18



ใส่ชื่อเครือข่าย เช่น “truemove”

ใส่รหัสผ่าน เช่น “123456”

Upload to esp8266



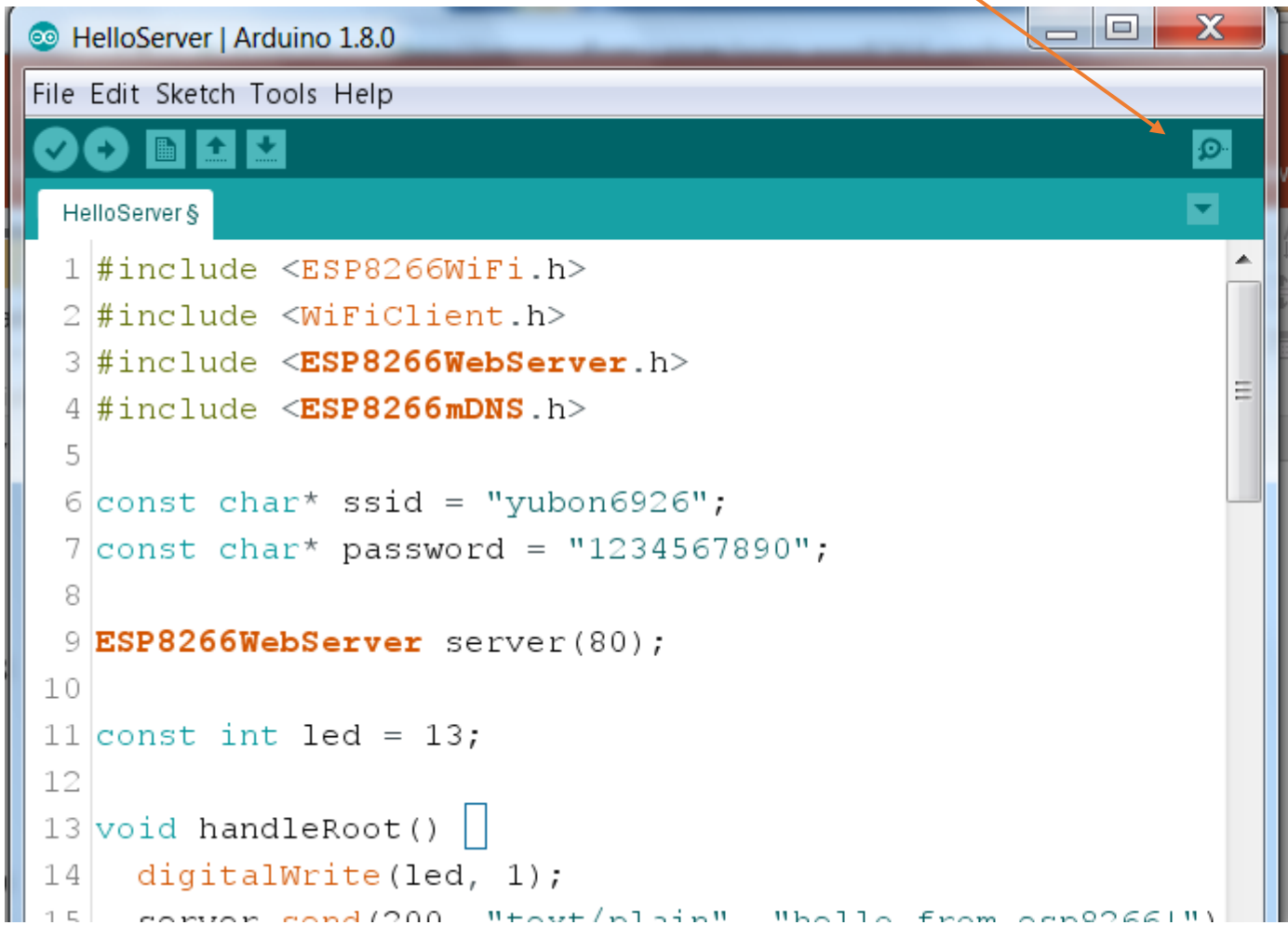
```
20  digitalWrite(led, 1);
```

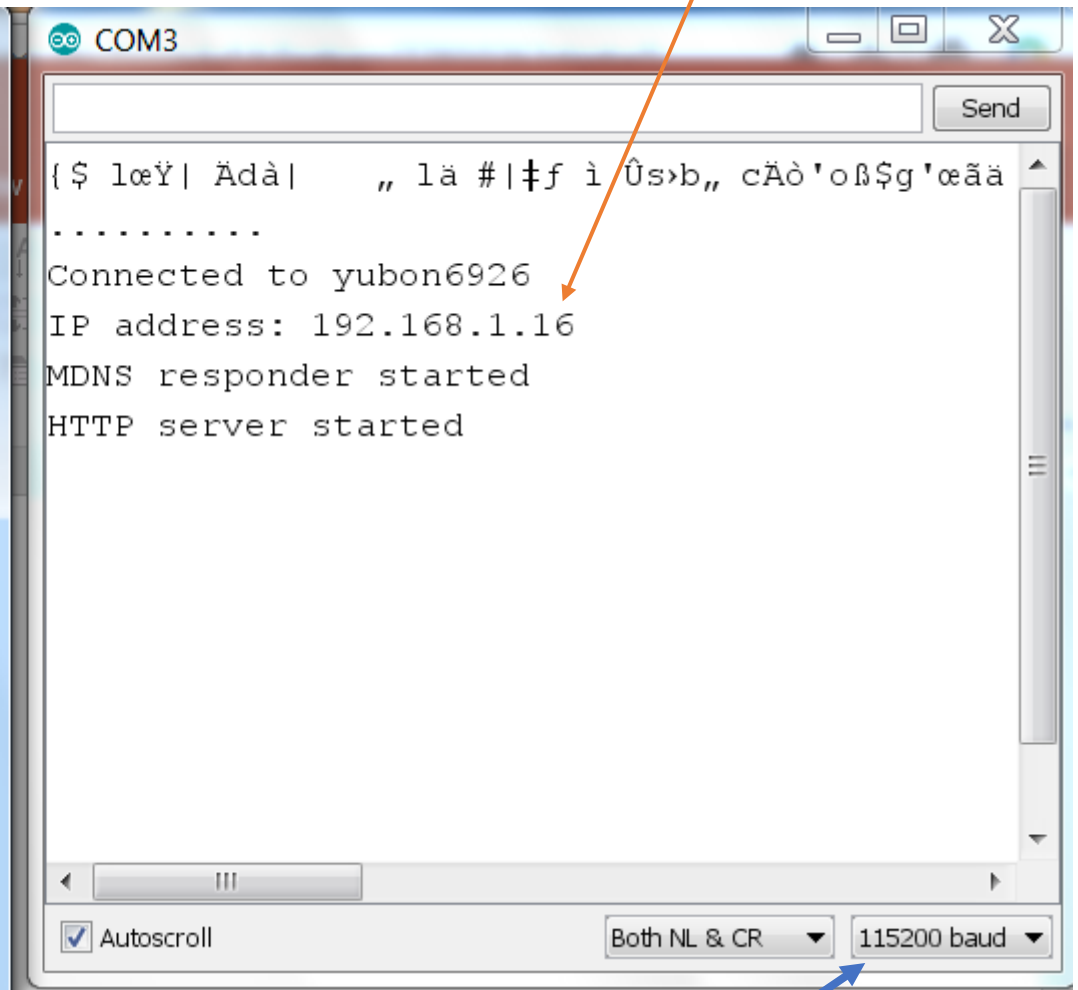
```
Uploading...
Build options changed, rebuilding all
Sketch uses 249279 bytes (23%) of program storage space. Maximum is 1044464 b
Global variables use 35772 bytes (43%) of dynamic memory, leaving 46148 bytes
Uploading 253424 bytes from C:\Users\123456\AppData\Local\Temp\58e9973bb6e2e9
.....
.....
7                                   WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS) on COM3
```

```
Done uploading.
Sketch uses 249279 bytes (23%) of program storage space. Maximum is 1044464
Global variables use 35772 bytes (43%) of dynamic memory, leaving 46148 byte
Uploading 253424 bytes from C:\Users\123456\AppData\Local\Temp\58e9973bb6e2e
.....
.....
.....
........
```

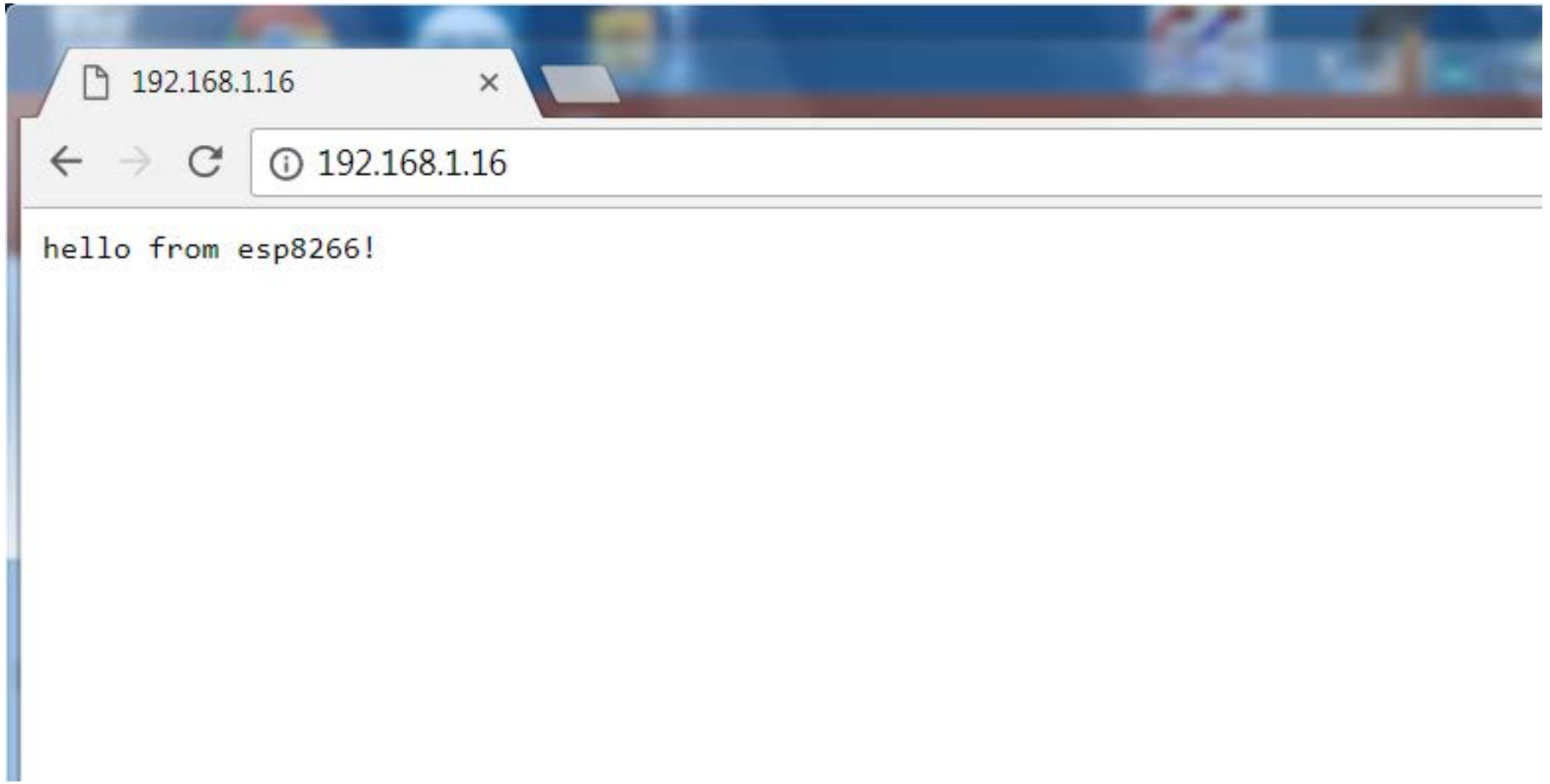
Open serial monitor

Got ip address





Select baud rate 115200


192.168.1.16
192.168.1.16
hello from esp8266!

HelloServer §

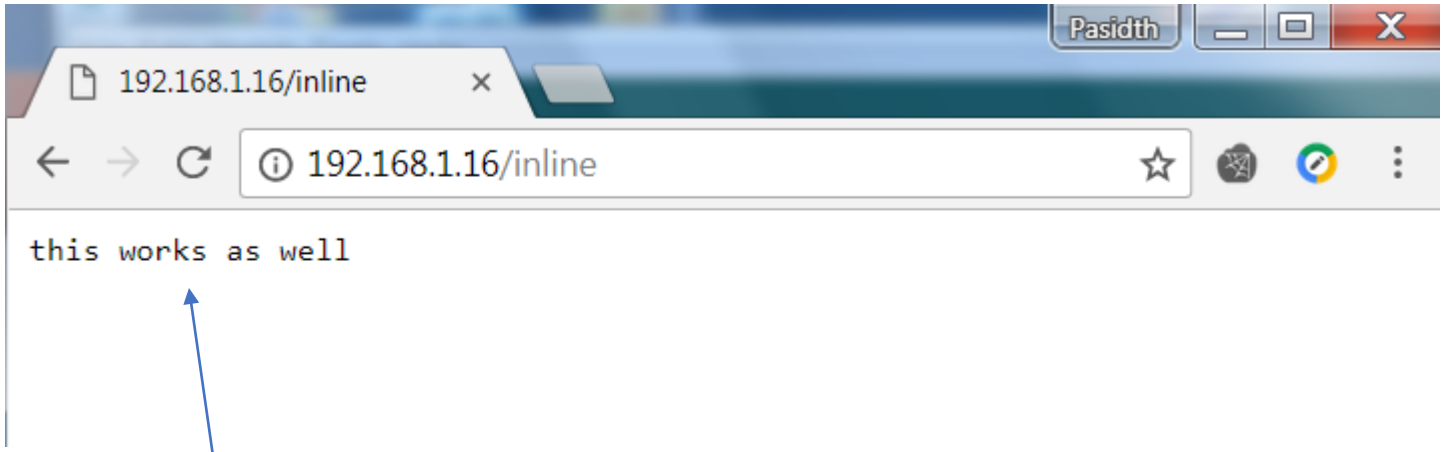
```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "smart";
const char* password = "123456789";

ESP8266WebServer server(80);

const int led = 2;

void handleRoot() {
  digitalWrite(led, 1);
  server.send(200, "text/plain", "hello from esp8266!");
  digitalWrite(led, 0);
}
```

Change to 2 for builtin LED

Text to display on browser

Led on and off each time sending to browser

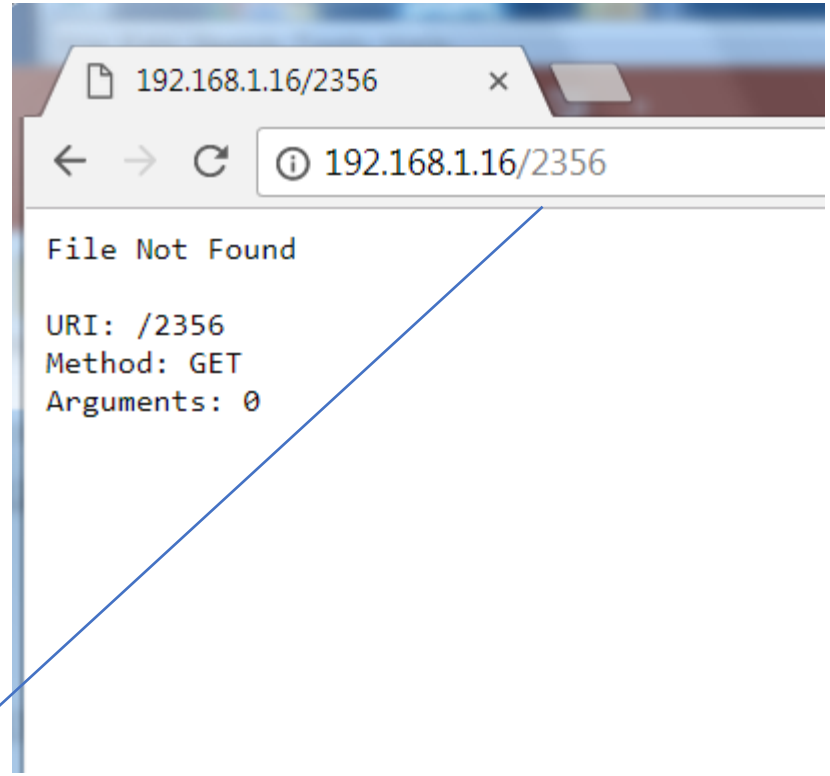
Request command




```
server.on("/", handleRoot);

server.on("/inline", [](){
  server.send(200, "text/plain", "this works as well");
});

server.onNotFound(handleNotFound);

server.begin();
Serial.println("HTTP server started");
```

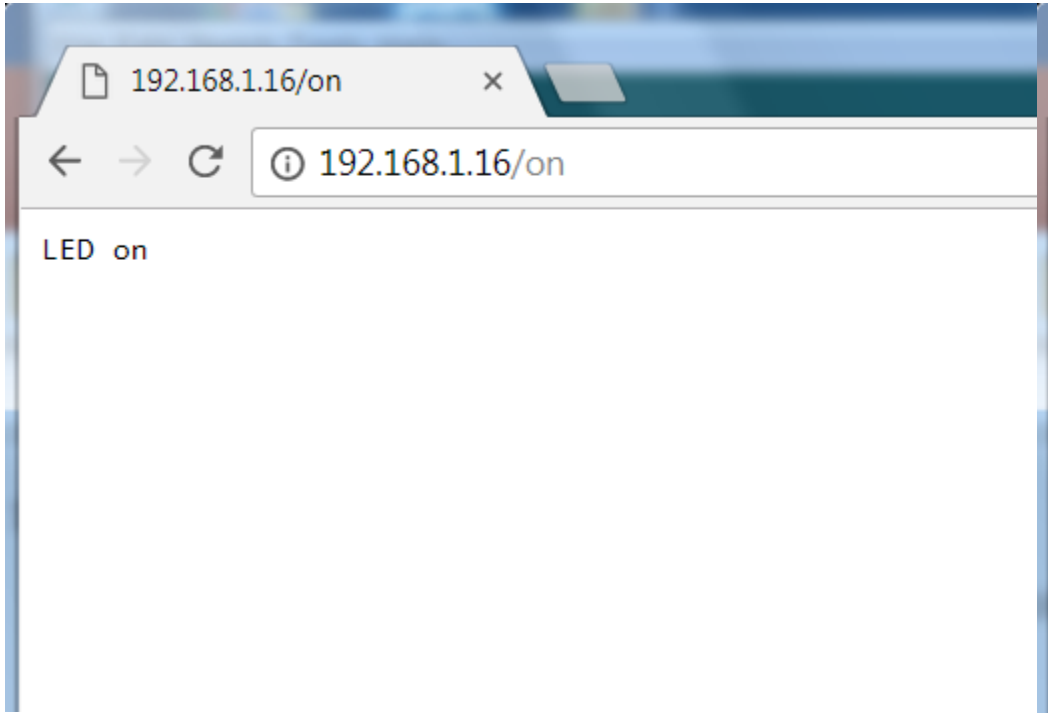
Send text

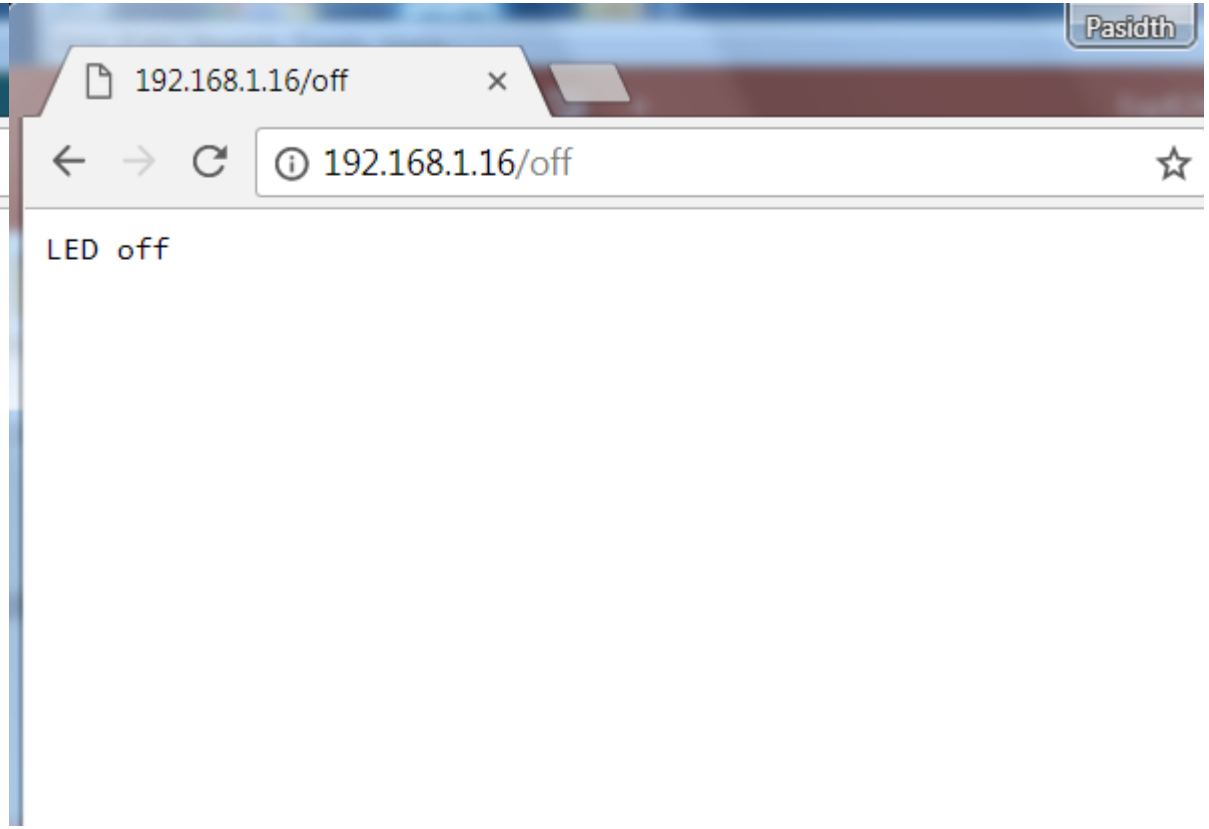
```
void handleNotFound(){
  digitalWrite(led, 1);
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET)?"GET":"POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i=0; i<server.args(); i++){
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
  digitalWrite(led, 0);
}


  server.on("/", handleRoot);

  server.on("/inline", [](){
    server.send(200, "text/plain", "this works as well");
  });

  server.onNotFound(handleNotFound);
```



Exercise 1-1

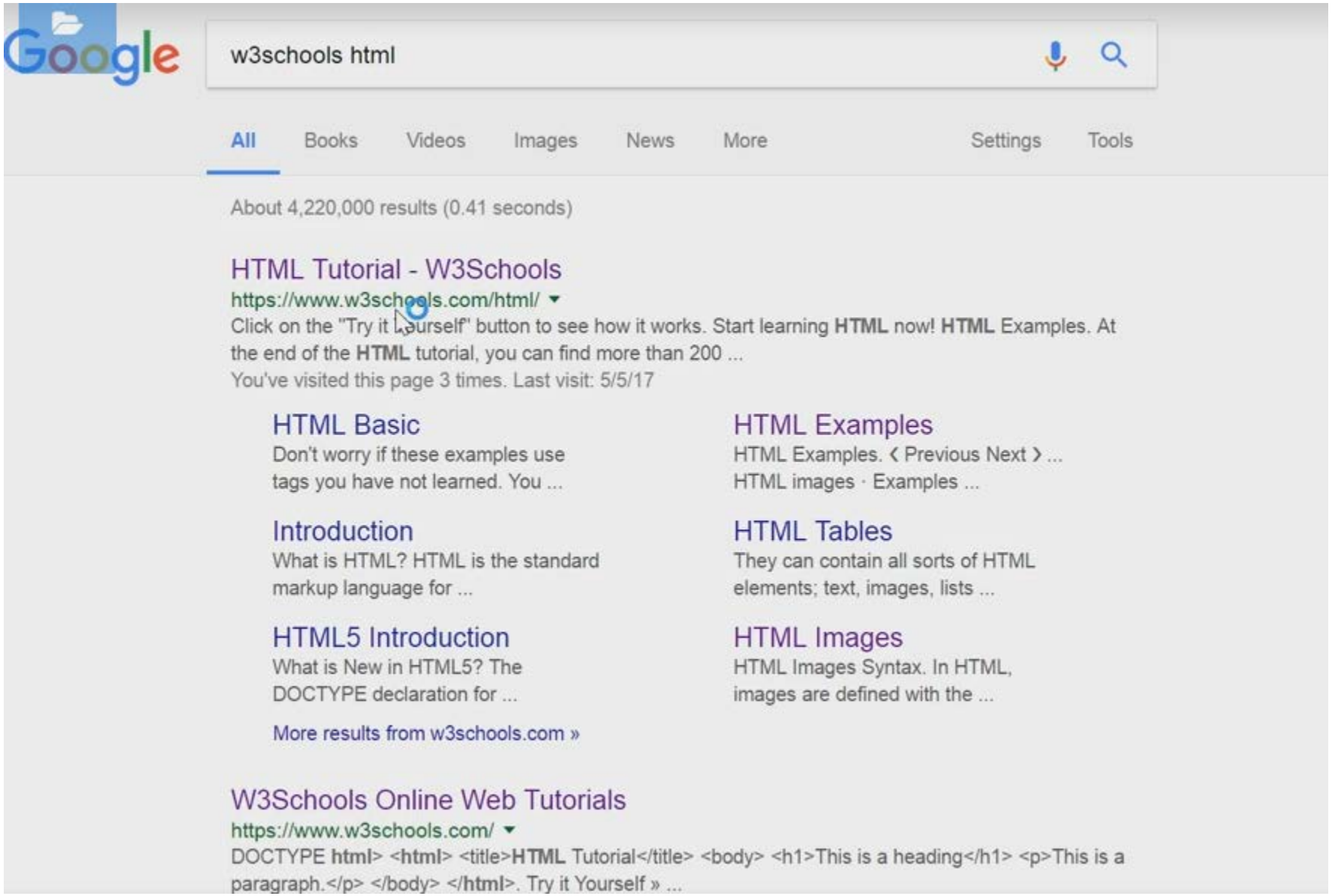
- เพิ่มคำสั่ง /on สั่งให้ LED on, /off สั่งให้ LED off และส่งข้อความกลับมาที่ browser

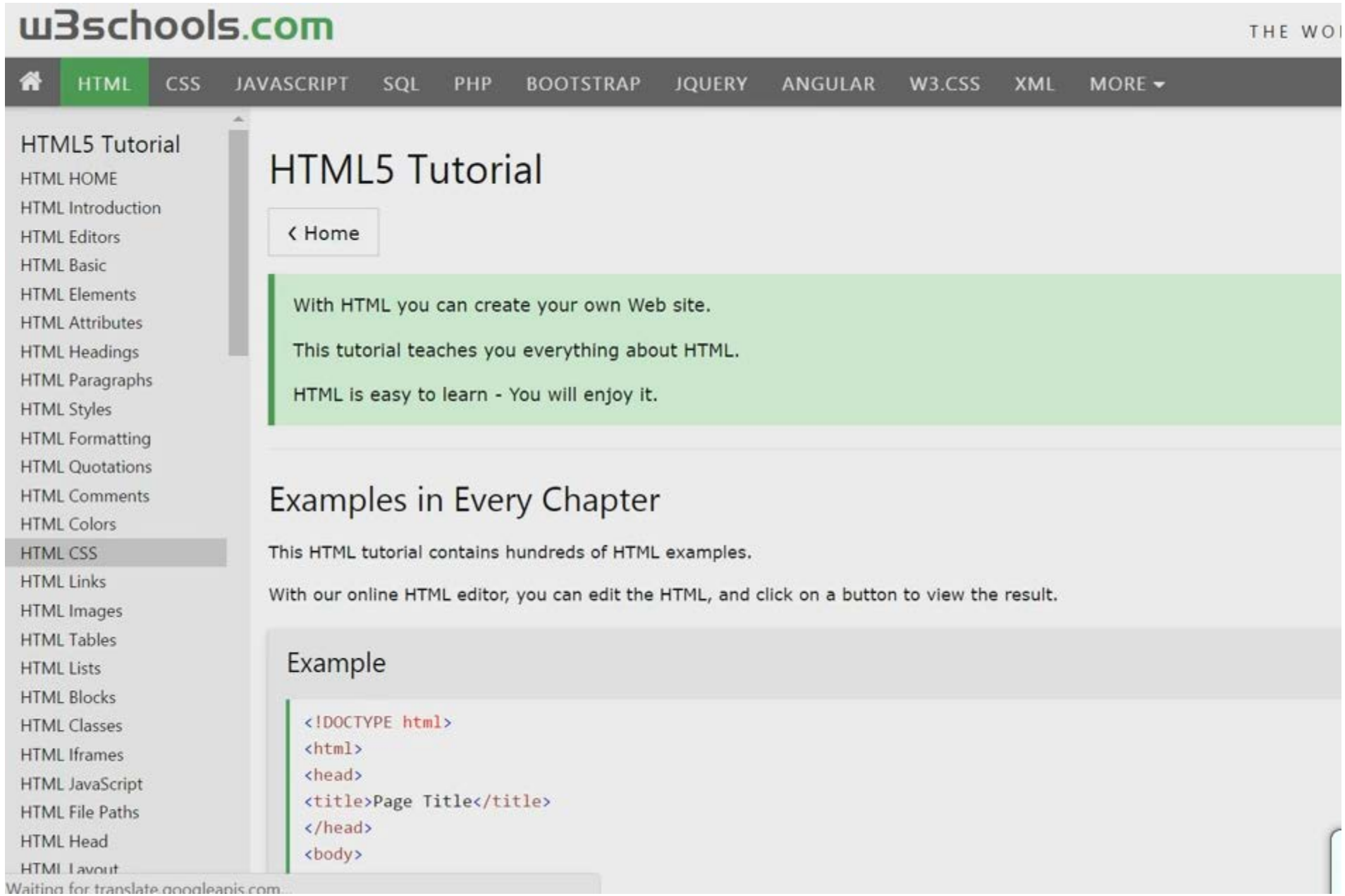

192.168.1.16/on
192.168.1.16/on
LED on


Pasidth
192.168.1.16/off
192.168.1.16/off
LED off

```
60
61  server.on("/", handleRoot);
62
63  server.on("/inline", [](){
64    server.send(200, "text/plain", "this works as well");
65  });
66
67  server.on("/on", [](){
68    server.send(200, "text/plain", "LED on");
69    digitalWrite(LED, 1);
70  });
71
72    server.on("/off", [](){
73    server.send(200, "text/plain", "LED off");
74    digitalWrite(LED, 0);
75  });
76
77
```

Put HTML to Arduino Sketch


Google
w3schools html
All
Books
Videos
Images
News
More
Settings
Tools
About 4,220,000 results (0.41 seconds)
HTML Tutorial - W3Schools
https://www.w3schools.com/html/
Click on the "Try it Yourself" button to see how it works. Start learning HTML now! HTML Examples. At the end of the HTML tutorial, you can find more than 200 ...
You've visited this page 3 times. Last visit: 5/5/17
HTML Basic
Don't worry if these examples use tags you have not learned. You ...
HTML Examples
HTML Examples. ❮ Previous Next ❯ ... HTML images · Examples ...
Introduction
What is HTML? HTML is the standard markup language for ...
HTML Tables
They can contain all sorts of HTML elements; text, images, lists ...
HTML5 Introduction
What is New in HTML5? The DOCTYPE declaration for ...
HTML Images
HTML Images Syntax. In HTML, images are defined with the ...
More results from w3schools.com »
W3Schools Online Web Tutorials
https://www.w3schools.com/
DOCTYPE html> <html> <title>HTML Tutorial</title> <body> <h1>This is a heading</h1> This is a paragraph. </body> </html>. Try it Yourself » ...


w3schools.com
THE WO
HTML
CSS
JAVASCRIPT
SQL
PHP
BOOTSTRAP
JQUERY
ANGULAR
W3.CSS
XML
MORE
HTML5 Tutorial
HTML HOME
HTML Introduction
HTML Editors
HTML Basic
HTML Elements
HTML Attributes
HTML Headings
HTML Paragraphs
HTML Styles
HTML Formatting
HTML Quotations
HTML Comments
HTML Colors
HTML CSS
HTML Links
HTML Images
HTML Tables
HTML Lists
HTML Blocks
HTML Classes
HTML Iframes
HTML JavaScript
HTML File Paths
HTML Head
HTML Layout
HTML5 Tutorial
‹ Home
With HTML you can create your own Web site.
This tutorial teaches you everything about HTML.
HTML is easy to learn - You will enjoy it.
Examples in Every Chapter
This HTML tutorial contains hundreds of HTML examples.
With our online HTML editor, you can edit the HTML, and click on a button to view the result.
Example
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
Waiting for translate.googleapis.com...

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> section of an HTML page, within a <style> element:

Example

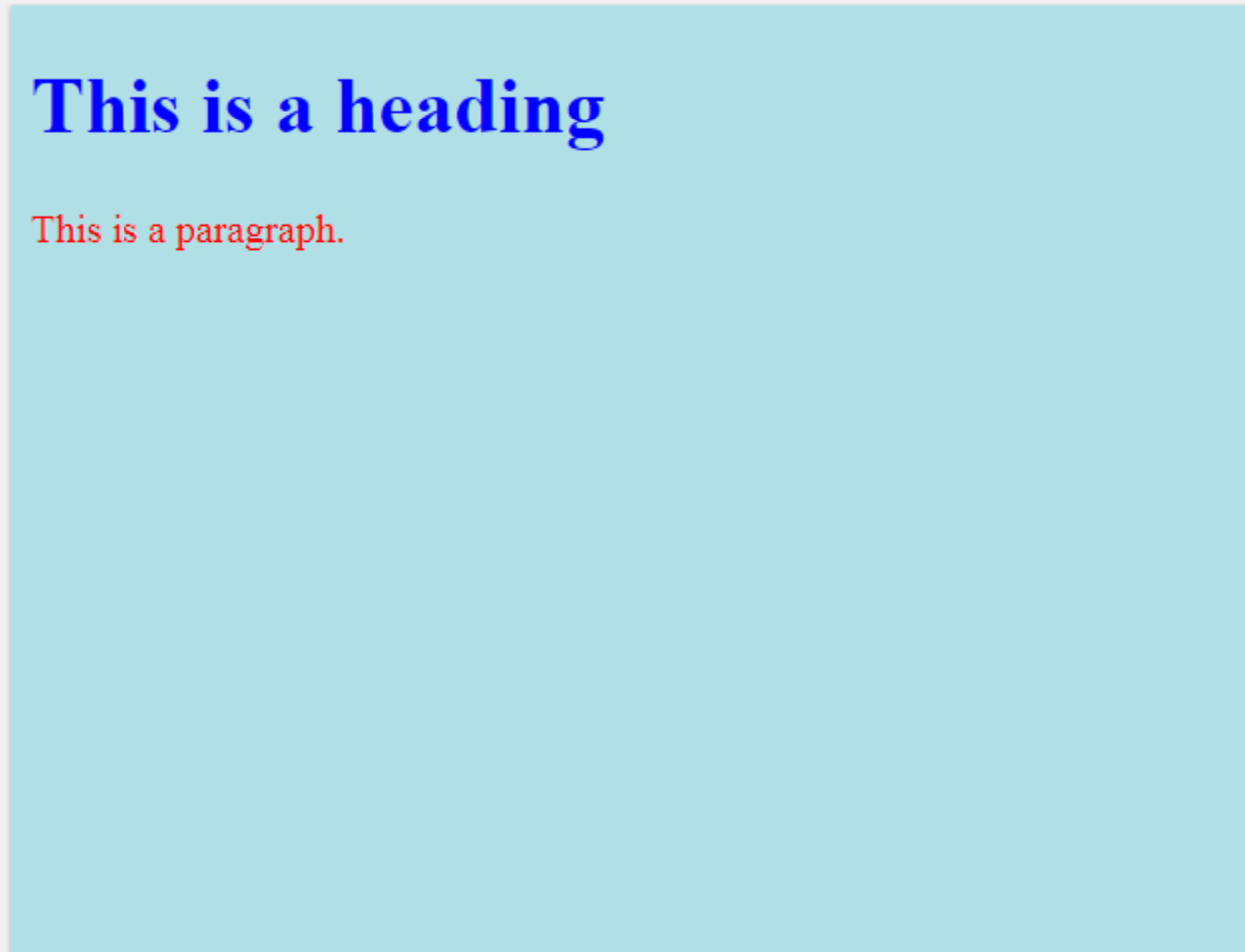
```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

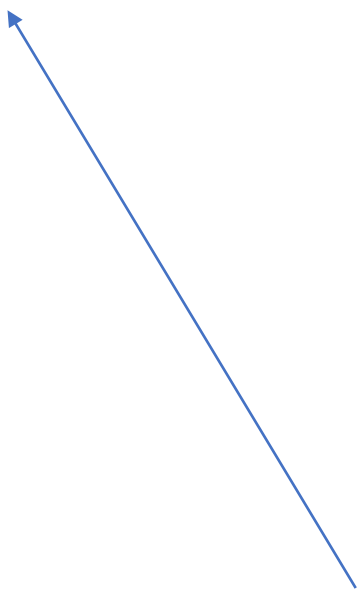
Run »

Result Size: 497 x 517

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



```html
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>ESP 8266 Server</h1>
<p>This is a test program.</p>

</body>
</html>
```



Convert to string format for Arduino sketch

Welcome

These days it seems like any idiot with a laptop computer can churn out a business book and make a few bucks. That's certainly what I'm hoping. - Scott Adams, *TDP* foreword.

This site contains some of my electronics related projects. You can also check here weather in Gdansk, Poland or use some simple javascript based online tools.

Special thanks to Ultra Trade.

Tomasz Ostrowski

Text -> C/C++ string converter

Converting text into C-like literal, escaping newlines, tab, double quotes, backslash.

Source text:

Options:

split output into multiple lines

Convert

C/C++ string

Sitemap

- PROJECTS
- SOFTWARE
- ONLINE TOOLS
  - Base64 -> HEX
  - Base32 -> HEX
  - Base32hex -> HEX
  - ASCII -> HEX
  - HEX -> Base64
  - HEX -> Base32
  - HEX -> Base32hex
  - HEX -> ASCII
  - HEX -> DEC
  - DEC -> ASCII
  - MD5 calculator
  - SHA1 calculator
  - Regular exp.
  - **Text -> Cpp**
  - Cpp -> text
  - Text -> Pascal
  - HEX -> file
  - File -> HEX
  - Text -> HTML ul
  - CRC8
  - WAVE generator
  - Bin decoder
  - RC4
  - XOR
- OTHER
- Links
- What's new?
- Contact

tomeko.net

Text -> C/C++ string converter

Converting text into C-like literal, escaping newlines, tab, double quotes, backslash.

Source text:

```
p    {color: red;}
</style>
</head>
<body>

<h1>Hello ESP 8266 Server</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Options:

☑ split output into multiple lines

Convert

C/C++ string

```
"<!DOCTYPE html>\n"
"<html>\n"
"<head>\n"
"<style>\n"
"body {background-color: powderblue;}\n"
"h1   {color: blue;}\n"
"p    {color: red;}\n"
"</style>\n"
"</head>\n"
"<body>\n"
```

HelloServer §

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char index_html[] PROGMEM = {};// put html into Program Memory
const char* ssid = "smart";
const char* password = "123456789";

ESP8266WebServer server(80);

const int led = 2; //builtin led
#define LED 0 //GPIO0 = D8

void handleRoot() {
  digitalWrite(led, 1);
  server.send(200, "text/plain", "hello from esp8266!");
  digitalWrite(led, 0);
}
```

```
const char index_html[] PROGMEM = {"<!DOCTYPE html>\n"
"<html>\n"
"<head>\n"
"<style>\n"
"body {background-color: powderblue;}\n"
"h1   {color: blue;}\n"
"p    {color: red;}\n"
"</style>\n"
"</head>\n"
"<body>\n"
"\n"
"<h1>Hello ESP 8266 Server</h1>\n"
"<p>This is a paragraph.</p>\n"
"\n"
"</body>\n"
"</html>"};// put html into Program Memory
const char* ssid = "smart";
const char* password = "123456789";
```

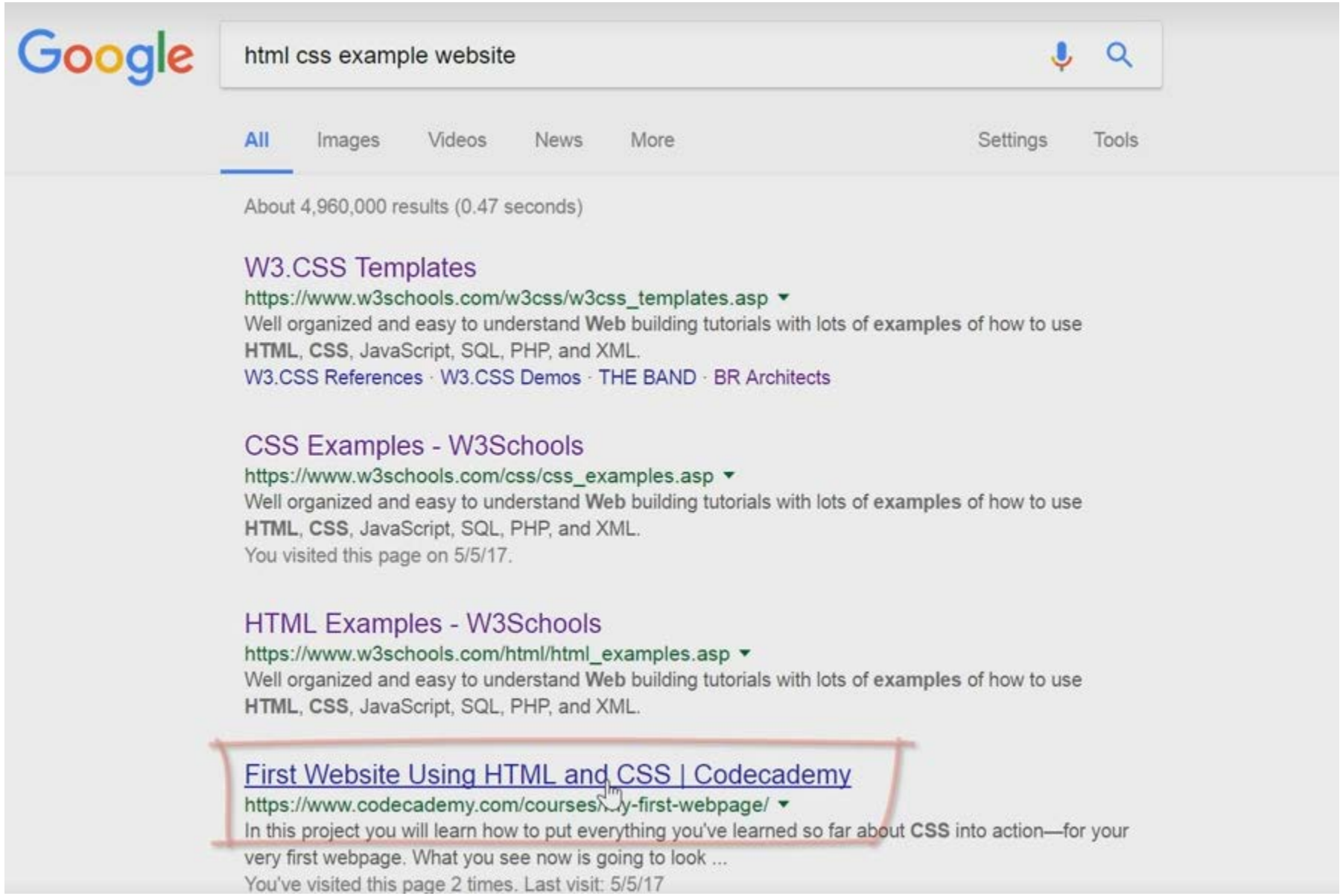
```
const int led = 2; //builtin led
#define LED 0 //GPIO0 = D8

void handleRoot() {
  digitalWrite(led, 1);
  server.send_P(200, "text/html", index_html);//send from Program memory html
  digitalWrite(led, 0);
}
```

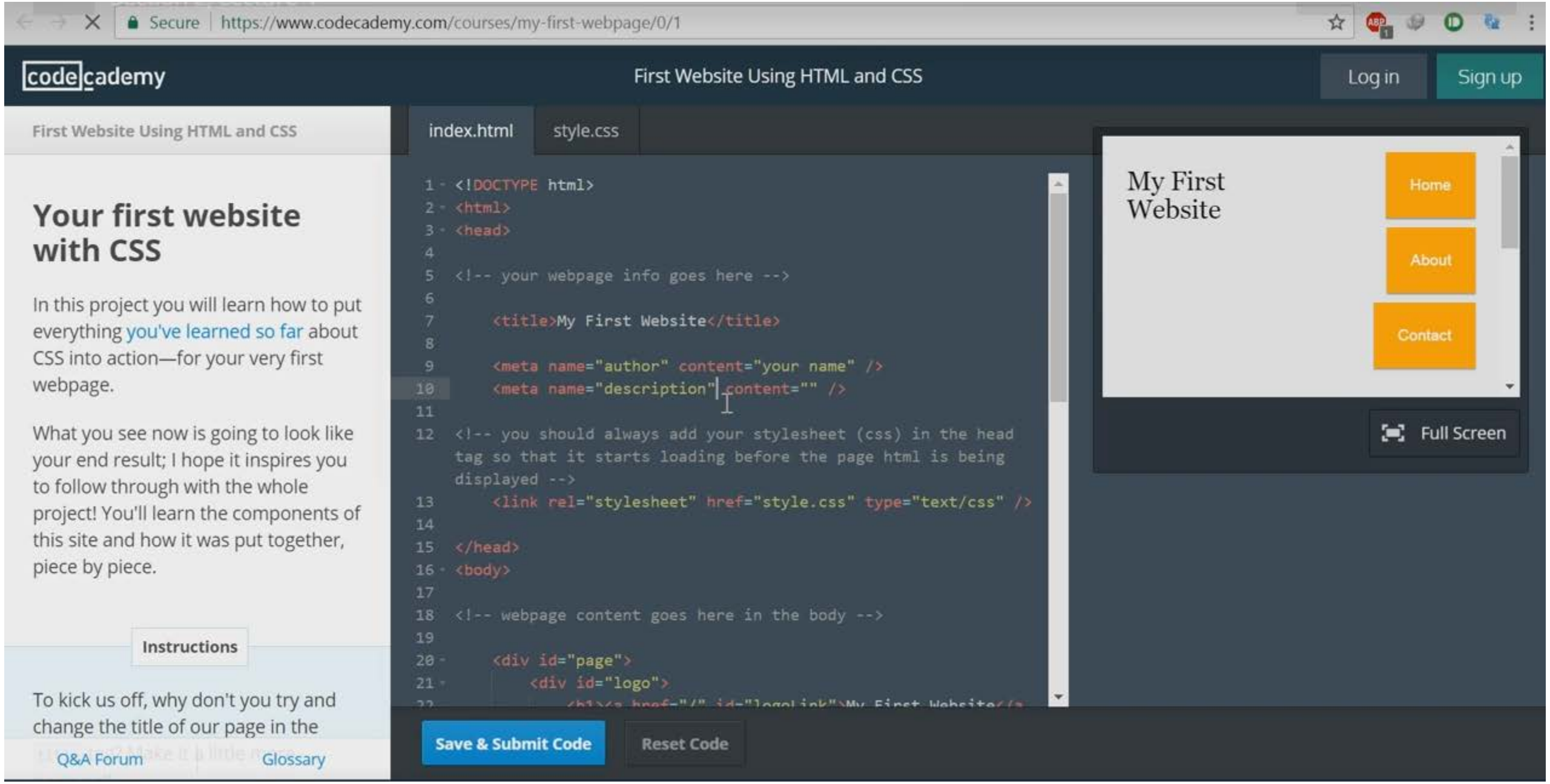
Change to index_html

Then comply and upload

Exercise 2

CSS Format


Google
html css example website
All
Images
Videos
News
More
Settings
Tools
About 4,960,000 results (0.47 seconds)
W3.CSS Templates
https://www.w3schools.com/w3css/w3css_templates.asp
Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, PHP, and XML.
W3.CSS References · W3.CSS Demos · THE BAND · BR Architects
CSS Examples - W3Schools
https://www.w3schools.com/css/css_examples.asp
Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, PHP, and XML.
You visited this page on 5/5/17.
HTML Examples - W3Schools
https://www.w3schools.com/html/html_examples.asp
Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, PHP, and XML.
First Website Using HTML and CSS | Codecademy
https://www.codecademy.com/courses/my-first-webpage/
In this project you will learn how to put everything you've learned so far about CSS into action—for your very first webpage. What you see now is going to look ...
You've visited this page 2 times. Last visit: 5/5/17


Secure
https://www.codecademy.com/courses/my-first-webpage/0/1
codecademy
First Website Using HTML and CSS
Log in
Sign up
First Website Using HTML and CSS
Your first website with CSS
In this project you will learn how to put everything you've learned so far about CSS into action—for your very first webpage.
What you see now is going to look like your end result; I hope it inspires you to follow through with the whole project! You'll learn the components of this site and how it was put together, piece by piece.
Instructions
To kick us off, why don't you try and change the title of our page in the
Q&A Forum
Glossary
index.html
style.css
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5 <!-- your webpage info goes here -->
6
7 <title>My First Website</title>
8
9 <meta name="author" content="your name" />
10 <meta name="description" content="" />
11
12 <!-- you should always add your stylesheet (css) in the head tag so that it starts loading before the page html is being displayed -->
13 <link rel="stylesheet" href="style.css" type="text/css" />
14
15 </head>
16 <body>
17
18 <!-- webpage content goes here in the body -->
19
20
21
My First Website
Home
About
Contact
Full Screen
Save & Submit Code
Reset Code

tomeko.net

Text -> C/C++ string converter

Converting text into C-like literal, escaping newlines, tab, double quotes, backslash.

Source text:

```
<!DOCTYPE html>
<html>
<head>

<!-- your webpage info goes here -->

    <title>My First Website</title>

        <meta name="author" content="your name" />
        <meta name="description" content="" />
```

Options:

☑ split output into multiple lines

Convert

C/C++ string

```
"<!DOCTYPE html>\n"
"<html>\n"
"<head>\n"
"\n"
"<!-- your webpage info goes here -->\n"
"\n"
"    <title>My First Website</title>\n"
"\t\n"
"\t<meta name=\"author\" content=\"your name\" />\n"
"\t<meta name=\"description\" content=\"\" />\n"
```

Can choose split output or in one line

Convert html to c++

tomeko.net

Text -> C/C++ string converter

Converting text into C-like literal, escaping newlines, tab, double quotes, backslash.

Source text:

```
/*
* multi-line comment
*/
p{ line-height: 1em; }
h1, h2, h3, h4{
    color: orange;
        font-weight: normal;
        line-height: 1.1em;
        margin: 0 0 .5em 0;
}
```

Options:

☑ split output into multiple lines

Convert

C/C++ string

```
"/*\n"
"* multi-line comment\n"
"*/\n"
"p{ line-height: 1em; }\n"
"h1, h2, h3, h4{\n"
"    color: orange;\n"
"\tfont-weight: normal;\n"
"\tline-height: 1.1em;\n"
"\tmargin: 0 0 .5em 0;\n"
"}\n"
```

Convert css to c++

Can choose split output or in one line

```
5
6  const char index_html[] PROGMEM = {"<!DOCTYPE html>\n<html>\n<head>\n\
7  const char style_css[] PROGMEM={"/*\n* multi-line comment\n*/\np{ line
8  const char* ssid = "smart";
9  const char* password = "123456789";
10
```

Use convert to one single line
Put converted c++ into { }

```
16 void handleRoot() {
17   digitalWrite(led, 1);
18   server.send_P(200, "text/html", index_html);//send from Pr
19   digitalWrite(led, 0);
20 }
21 void handleCss(){
22   server.send_P(200,"text/css",style_css);
23 }
```

Add void handleCss()

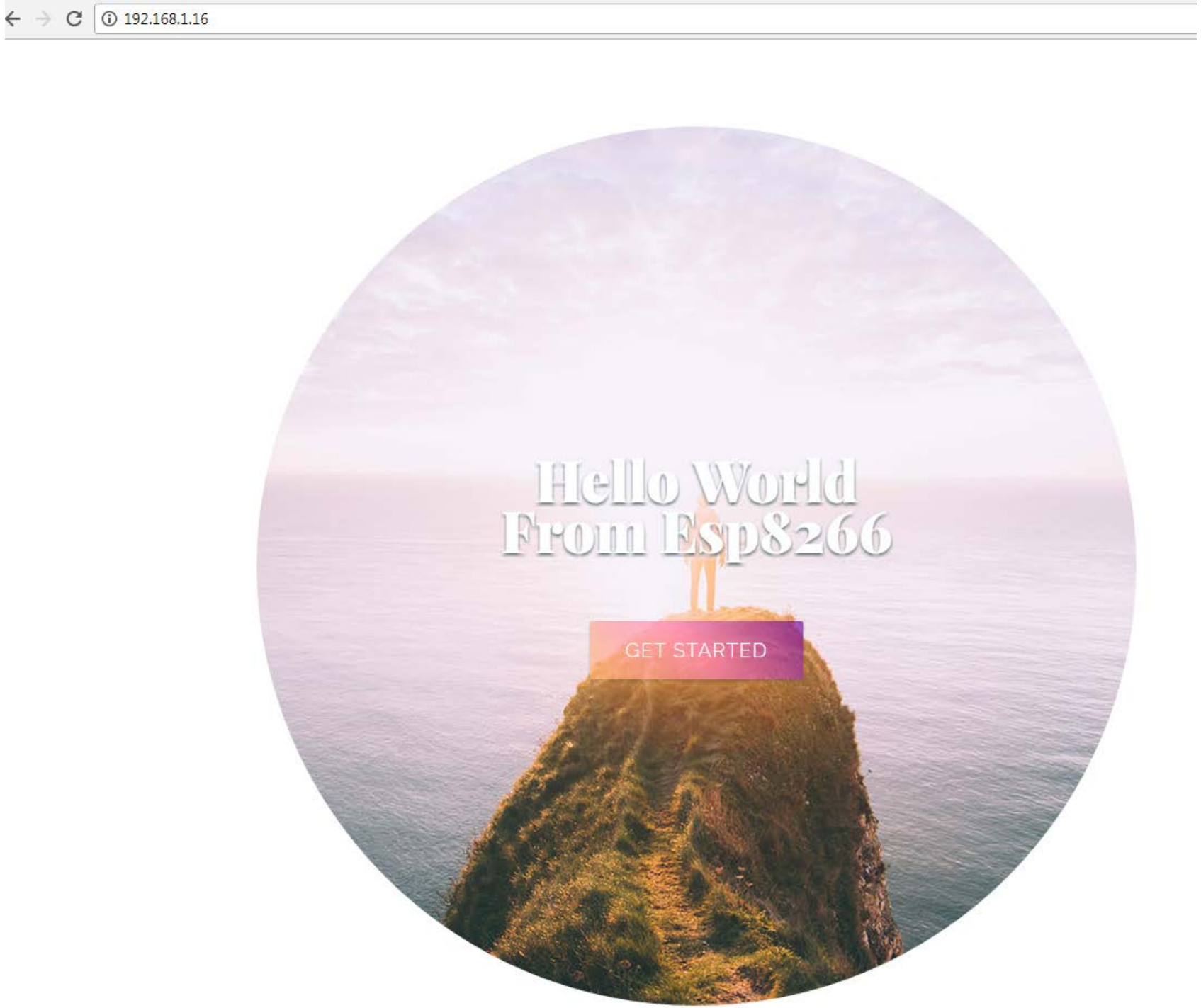
```
66
67  server.on("/style.css",handleCss);
68
69  server.on("/inline", [](){
70    server.send(200, "text/plain", "this works as well");
71  });
```

Add server.on for css

Then upload to sketch


192.168.1.16
My First Website
Home
About
Contact
Home
This is my first webpage! I was able to code all the HTML and CSS in order to make it. Watch out world of web design here I come!
I can use my skills here to create websites for my business, my friends and family, my C.V, blog or articles. As well as any games or more experiment stuff (which is what the web is really all about).
WEBPAGE MADE BY [YOUR NAME]

Exercise 3



Exercise 3-1

Learn How to Put Photo in Html

Run »

Result Size: 497 x 566

```
<!DOCTYPE html>
<html>
<body>

<img id="myImage" onclick="changeImage()"
src="pic_bulboff.gif" width="100" height="180">

<p>Click the light bulb to turn on/off the light.</p>

<script>
function changeImage() {
    var image = document.getElementById('myImage');
    if (image.src.match("bulbon")) {
        image.src = "pic_bulboff.gif";
    } else {
        image.src = "pic_bulbon.gif";
    }
}
</script>

</body>
</html>
```

Click the light bulb to turn on/off the light.

tomeko.net

Text -> C/C++ string converter

Converting text into C-like literal, escaping newlines, tab, double quotes, backslash.

Source text:

```
<!DOCTYPE html>
<html>
<body>

<img id="myImage" onclick="changeImage()" src="pic_bulboff.gif"
width="100" height="180">

<p>Click the light bulb to turn on/off the light.</p>

<script>
```

Options:

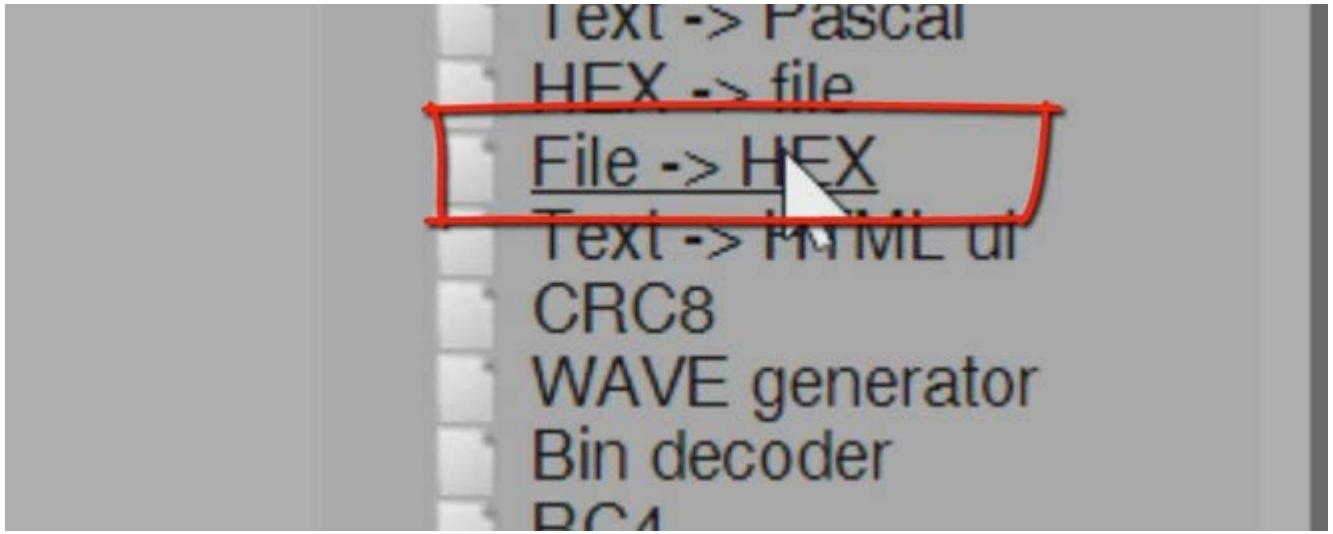
☐ split output into multiple lines

Convert

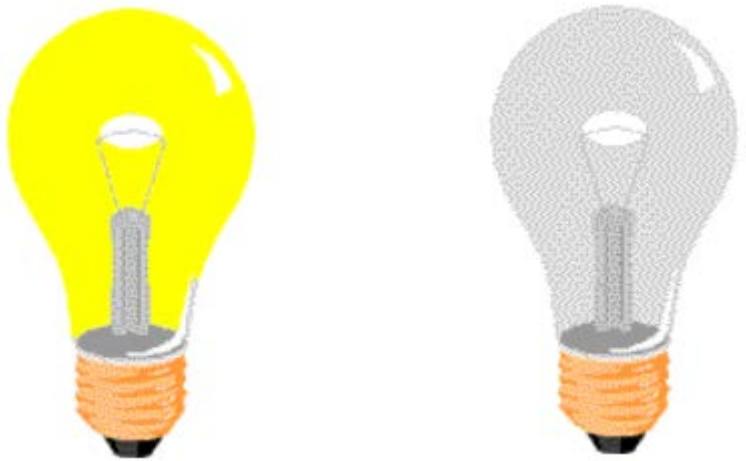
C/C++ string

```
"<!DOCTYPE html>\n<html>\n<body>\n\n<img id=\"myImage\"
onclick=\"changeImage()\" src=\"pic_bulboff.gif\" width=\"100\"
height=\"180\">\n\n<p>Click the light bulb to turn on/off the light.
</p>\n\n<script>\nfunction changeImage() {\n    var image =
document.getElementById('myImage');\n    if (image.src.match(\"bulbon\"))
{\n        image.src = \"pic_bulboff.gif\";\n    } else {\n
image.src = \"pic_bulbon.gif\";\n    }\n}\n</script>\n\n</body>\n</html>"
```

Convert html to C++



Convert fil.gif to Hex file and put into sketch



File to hexadecimal converter

Client-side (javascript, no data is sent to server) file to hexadecimal code conversion. Be careful with files > 1 MB (possible high resource consumption, e.g. Chromium 46 has serious problems when loading few MB of text into textarea, offline tools might be better for large files).

File: Choose File pic_bulbon.gif or **drop file here**

Options:

- [x] Use 0x and comma as separator (C-like)
- [ ] Insert newlines after each 16B

Convert

Output:

```
0x47, 0x49, 0x46, 0x38, 0x39, 0x61, 0x64, 0x00, 0xB4, 0x00, 0xD5, 0xFF, 0x00, 0xC0, 0xC0,
0xC0, 0xFF, 0xFF, 0xCC, 0xFF, 0xFF, 0x99, 0xFF, 0xFF, 0x66, 0xFF, 0xFF, 0x33, 0xFF, 0xFF,
0x00, 0xFF, 0xCC, 0xFF, 0xFF, 0xCC, 0xCC, 0xFF, 0xCC, 0x99, 0xFF, 0xCC, 0x66, 0xFF, 0xCC,
0x33, 0xFF, 0x99, 0x99, 0xFF, 0x99, 0x66, 0xFF, 0x99, 0x33, 0xCC, 0xFF, 0xFF, 0xCC, 0xFF,
0xCC, 0xCC, 0xFF, 0x66, 0xCC, 0xFF, 0x33, 0xCC, 0xCC, 0xFF, 0xCC, 0xCC, 0xCC, 0xCC, 0xCC,
0x99, 0xCC, 0xCC, 0x66, 0xCC, 0x99, 0xCC, 0xCC, 0x99, 0x99, 0xCC, 0x99, 0x66, 0xCC, 0x99,
0x33, 0x99, 0xCC, 0xCC, 0x99, 0xCC, 0x99, 0x99, 0x99, 0xCC, 0x99, 0x99, 0x99, 0x99, 0x99,
0x66, 0x99, 0x66, 0x66, 0x99, 0x66, 0x33, 0x66, 0x66, 0x99, 0x66, 0x66, 0x66, 0x66, 0x33,
0x66, 0x66, 0x33, 0x33, 0x33, 0x66, 0x66, 0x33, 0x66, 0x33, 0x33, 0x33, 0x66, 0x33, 0x33,
0x33, 0x33, 0x33, 0x00, 0x00, 0x33, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

See also: bin2hex.exe for Windows or bin2hex for Linux (source).

Sitemap

- PROJECTS
- SOFTWARE
- ONLINE TOOLS
  - Base64 -> HEX
  - Base32 -> HEX
  - Base32hex -> HEX
  - ASCII -> HEX
  - HEX -> Base64
  - HEX -> Base32
  - HEX -> Base32hex
  - HEX -> ASCII
  - HEX -> DEC
  - DEC -> ASCII
  - MD5 calculator
  - SHA1 calculator
  - Regular exp.
  - Text -> Cpp
  - Cpp -> text
  - Text -> Pascal
  - HEX -> file
  - **File -> HEX**
  - Text -> HTML ul
  - CRC8
  - WAVE generator
  - Bin decoder
  - RC4
  - XOR
- OTHER
- Links
- What's new?
- Contact

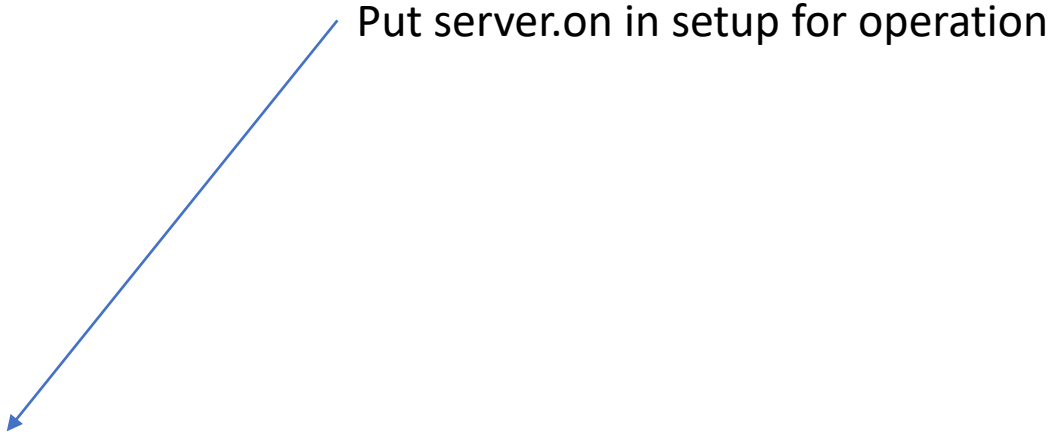
```
5
6  const char index_html[] PROGMEM = {"<!DOCTYPE html>\n<html>\n<body>\r
7  //const char style_css[] PROGMEM={"/*\n* multi-line comment\n*/\np{ 
8  const char bulb_on_gif[] PROGMEM={0x47, 0x49, 0x46, 0x38, 0x39, 0x61,
9  const char bulb_off_gif[] PROGMEM={0x47, 0x49, 0x46, 0x38, 0x39, 0x6
10
11
```

Convert photo to hex file and put in Arduino sketch

```
26 void handleBulbOn(){
27   server.send_P(200,"image/gif",bulb_on_gif,sizeof(bulb_on_gif));
28 }
29 void handleBulbOff(){
30   server.send_P(200,"image/gif",bulb_off_gif,sizeof(bulb_off_gif));
31 }
32 void handleNotFound(){
33   digitalWrite(led, 1);
34   String message = "File Not Found\n\n";
```

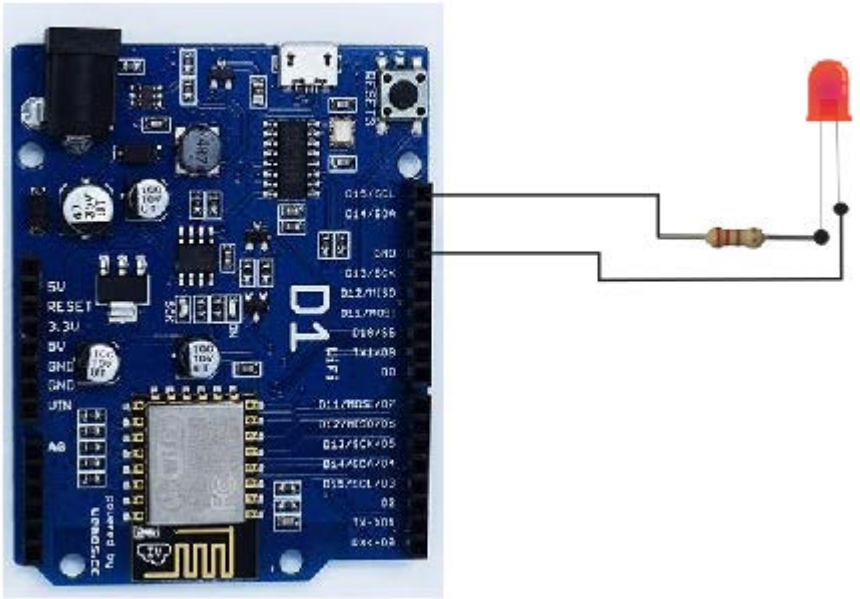
Add void handle

```
81  server.on("/on", [](){
82    server.send(200, "text/plain", "LED on");
83    digitalWrite(LED, 1);
84  });
85
86    server.on("/off", [](){
87    server.send(200, "text/plain", "LED off");
88    digitalWrite(LED, 0);
89  });
90  server.on("/pic_bulloff.gif",handleBulbOff);
91  server.on("/pic_bullon.gif",handleBulbOn);
92
```

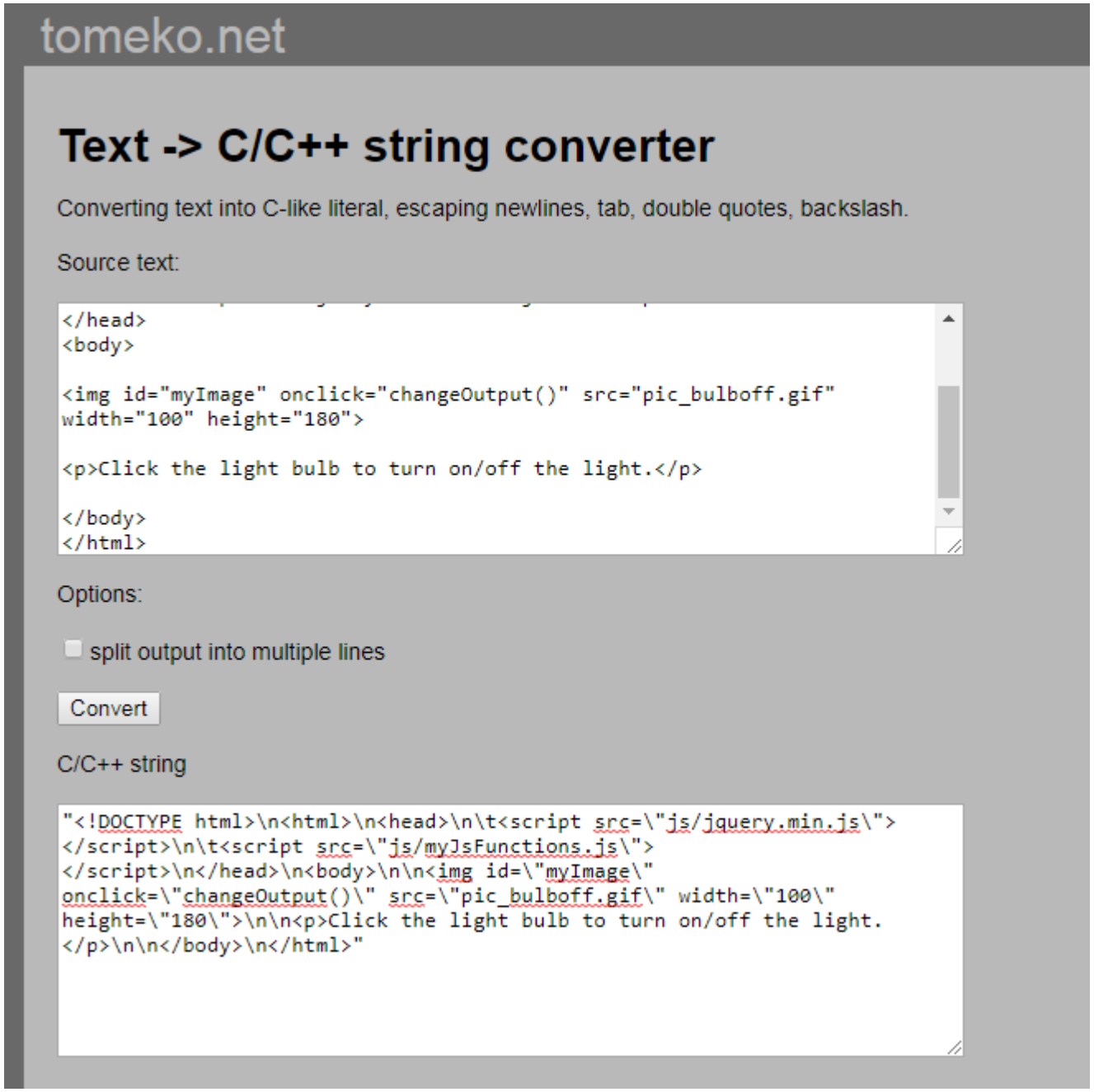


Example 4

With Javascript



Exercise 5



1. Convert files
1.1 Convert index_html to C++
1.2 Convert jquery.min to C++
1.3 Convert myJsFunctions to C++
1.4 Put into sketch

Esp8266JSLightBulb

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

int ledOutput = 5;//gpio5 = D3,D15
const char index_html[] PROGMEM={"<!DOCTYPE html>\n<html>\n<head>\n\t<script
const char bulb_off_gif[] PROGMEM={0x47, 0x49, 0x46, 0x38, 0x39, 0x61, 0x64,
const char bulb_on_gif[] PROGMEM={0x47, 0x49, 0x46, 0x38, 0x39, 0x61, 0x64,
const char jquery_js[] PROGMEM={"!function(a,b){\"use strict\";\"object\"==t
```

```
26 void handlejQuery(){
27   server.send_P(200,"application/js",jquery_js);
28 }
29 void handleMyfunctions(){
30     server.send_P(200,"application/js",myFunctions_js);
31 }
32 void handleDigitalOutputToggle(){
33   String response="";
34   digitalWrite(ledOutput, !digitalRead(ledOutput));
35   response+=!digitalRead(ledOutput);
36   server.send(200,"text",response);
37   Serial.println(response);
38 //  digitalWrite(ledOutput,response);
39 }
40
```

2. Add void function

```
  server.on("/", handleRoot);
  server.on("/pic_bulboff.gif",handleBulbOff);
  server.on("/pic_bulbon.gif",handleBulbOn);
  server.on("/js/jquery.min.js",handlejQuery);
  server.on("/digital_output/toggle",handleDigitalOutputToggle);

  server.onNotFound(handleNotFound);

  server.begin();
  Serial.println("HTTP server started");
}
```

3. Add server.on
4. Complie and upload to sketch

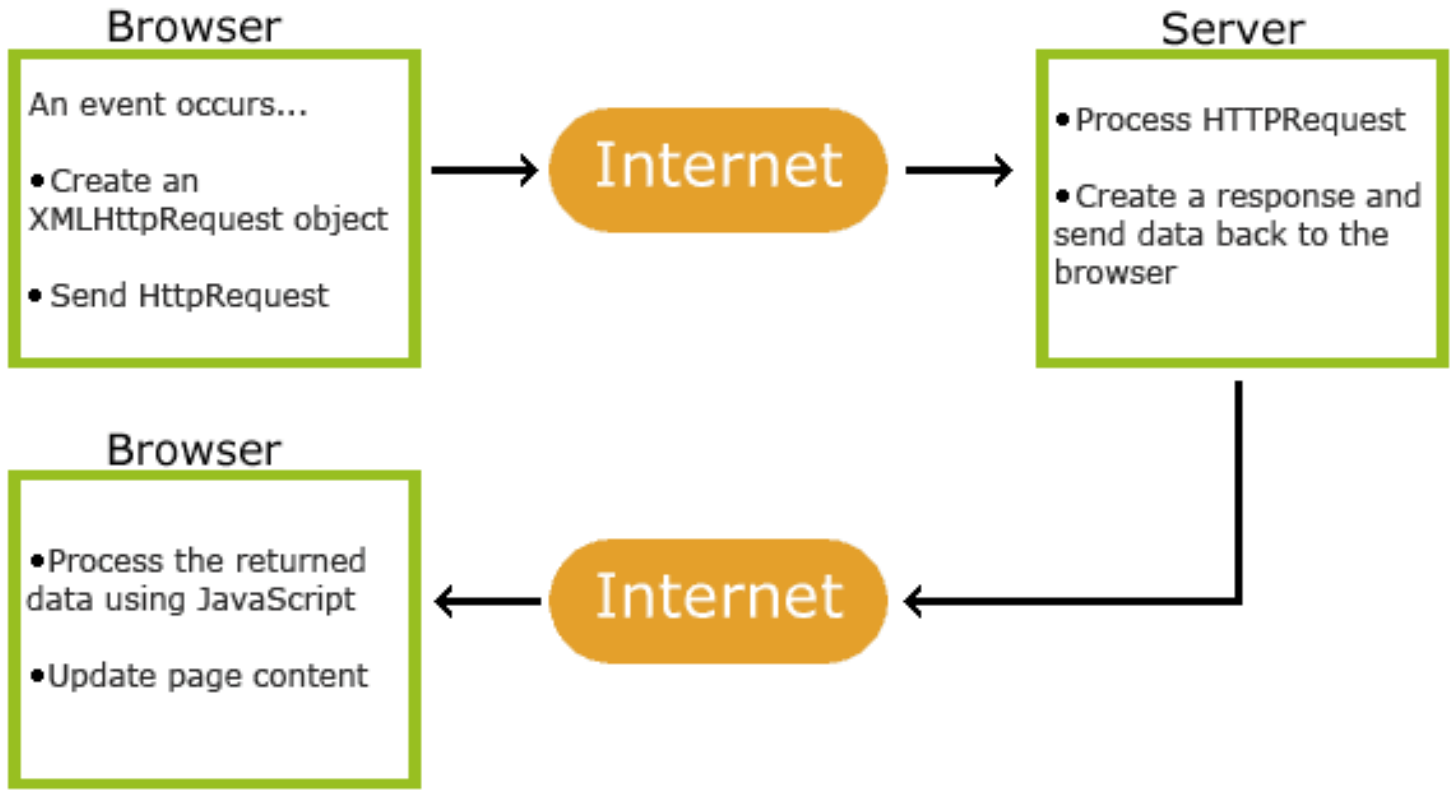
Multiple inputs and Outputs

Example 6

AJAX = Asychronous Javascriprt And XML

- AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.
- AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

How AJAX Works



1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

AJAX - Send a Request To a Server

The XMLHttpRequest object is used to exchange data with a server.

Send a Request To a Server

To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

```
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

Method	Description
open(*method, url, async*)	Specifies the type of request *method*: the type of request: GET or POST *url*: the server (file) location *async*: true (asynchronous) or false (synchronous)
send()	Sends the request to the server (used for GET)
send(*string*)	Sends the request to the server (used for POST)

GET or POST?

GET is simpler and faster than POST, and can be used in most cases.

However, always use POST requests when:

- A cached file is not an option (update a file or database on the server).
- Sending a large amount of data to the server (POST has no size limitations).
- Sending user input (which can contain unknown characters), POST is more robust and secure than GET.

GET Requests

A simple GET request:

Example

```
xhttp.open("GET", "demo_get.asp", true);
xhttp.send();
```

The url - A File On a Server

The url parameter of the open() method, is an address to a file on a server:

```
xhttp.open("GET", "ajax_test.asp", true);
```

The file can be any kind of file, like .txt and .xml, or server scripting files like .asp and .php (which can perform actions on the server before sending the response back).

Asynchronous - True or False?

Server requests should be sent asynchronously.

The async parameter of the open() method should be set to true:

```
xhttp.open("GET", "ajax_test.asp", true);
```

By sending asynchronously, the JavaScript does not have to wait for the server response, but can instead:

- execute other scripts while waiting for server response
- deal with the response after the response is ready

```
<!DOCTYPE html>
<html>
<body>
<h2>The XMLHttpRequest Object</h2>
<p id="demo">Let AJAX change this text.</p>
<button type="button" onclick="loadDoc()">Change Content</button>
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText; }  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();}
</script>
</body>
</html>
```

ตย. XMLHttpRequest Object AJAX

1. Create “ajax_info.txt”
2. Save html file
3. Run in browser (firefox)

Java script object notation (JSON)

JSON is a syntax for storing and exchanging data.
JSON is text, written with JavaScript object notation.

Exchanging Data

When exchanging data between a browser and a server, the data can only be text.
JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server.
We can also convert any JSON received from the server into JavaScript objects.
This way we can work with the data as JavaScript objects, with no complicated parsing and translations.

Sending Data

If you have data stored in a JavaScript object, you can convert the object into JSON, and send it to a server:

การใช้ ajax คู่กับ json

```
<!DOCTYPE html>
<html>
<head>
   <title>Esp8266 I/O</title>  //title of browser
   <link rel="stylesheet" type="text/css" href="css/style.css">
   <script src="js/jquery.min.js"></script>
   <script>
      $(document).ready(function() {
         setInterval("get_digital_output_status()", 2000);
      });
      function get_digital_output_status() { var someUrl = "/digital_outputs";
         $.ajax({url: someUrl,dataType: "json",success: function(response) {  //send data to a web server
               if (response.digital_outputs["dout1"] == 1)  $("#dout1").html("HIGH")
               else   $("#dout1").html("LOW")
               if (response.digital_outputs["dout2"] == 1)  $("#dout2").html("HIGH")
               else   $("#dout2").html("LOW")
               if (response.digital_outputs["dout3"] == 1)  $("#dout3").html("HIGH")
               else  $("#dout3").html("LOW")
               if (response.digital_outputs["dout4"] == 1) $("#dout4").html("HIGH")
               else   $("#dout4").html("LOW")}})}
```