

[Home](#) | [Software](#)

[Beginners](#)

[Projects](#)

[Tutorials](#)

[Articles](#)

[Reviews](#)

STARTING ELECTRONICS

Electronics for Beginners and Beyond

[Blog](#)

[YouTube](#)

[Donate](#)

[Home](#) ▶ [Tutorials](#) ▶ [Arduino](#) ▶ [Ethernet Shield Web Server Tutorial](#) ▶ Ethernet Shield Tutorial Summary

Arduino Ethernet Shield Web Server Tutorial Summary and Conclusion

Created on: 11 April 2013

Final part of the Arduino Ethernet Shield Web Server Tutorial

In this final part of the multi-part Arduino Ethernet shield web server tutorial, we summarise the technologies covered by the tutorial and provide some extra information. We also look at where to get further information on the various technologies and how to progress from here.

HTTP

The first technology to get familiar with when writing code for a web server that hosts HTML web pages is the Hypertext Transfer Protocol (HTTP). HTTP is normally invisible to the user of a web browser, except for **http://** in the browser address bar. HTTP is happening behind the scenes for the web surfer.

Donate to Starting Electronics

[Donate](#)



Arduino Ethernet Shield Tutorial

[Tutorials](#)

[Arduino](#)

[Part 1: Ethernet Shield Tutorial Introduction and Hardware](#)

[Part 2: Basic Arduino Web Server](#)

[Part 3: HTML Web Page Structure](#)

Because we are writing web server code on our Arduino, we need to know how HTTP works. When a web browser connects to an Arduino web server by a user entering the IP address in the address bar of the web browser (the IP address is set in the Arduino sketch), the web browser sends an HTTP request for the default or home page of the web server.

The Arduino web server must answer the HTTP request with the text that the protocol requires and then send the actual web page.

A good resource for learning more about HTTP is [HTTP Made Really Easy](#) by James Marshall.

[Part 2](#) of this Arduino web server tutorial shows how HTTP is handled in the first Arduino web server sketch.

HTML and XHTML

Every web page consists of a text file written in Hypertext Markup Language (HTML). HTML structures the web page, marking which portion of the text are paragraphs, headers, checkboxes, buttons, etc.

There are actually several versions of HTML and also several versions of XHTML. XHTML is also used for creating web pages, but has a stricter syntax than HTML. For example, all (X)HTML tags must use lower-case letters in XHTML, but HTML can use upper or lower-case letters.

A web page can be written in any of the standards and could be HTML or XHTML – the person browsing a website would normally not know which standard the website is using unless he looked at the source code of a web page from the site.

HTML 5

We have actually been using HTML 5 in this tutorial. The version of HTML or XHTML used in a web page is specified by the very first line of markup in the web page. All the

Part 4:
Arduino SD
Card Web
Server

Part 5:
Arduino Web
Server LED
Control

Part 6:
Reading a
Switch

Part 7:
Reading a
Switch using
AJAX

Part 8:
Reading a
Switch
Automatically
using AJAX

Part 9:
Reading an
Analog Input
and Switches
using AJAX

Part 10:
Linking Web
Pages

Part 11: Web
Page Images

Part 12: CSS
Introduction

examples in this tutorial have used the following line at the top of the HTML page:

```
<!DOCTYPE html>
```

This lets the browser know that the web page that it is loading is written using HTML 5.

XHTML 1.0

Other HTML and XHTML standards use different markup for the first line of the web page. For example, strict and transitional XHTML use the following lines respectively (these are shown on two lines each here to fit on the page, the two lines are normally written on one line and separated by a space):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

< >

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

< >

HTML 4.01

The first line of markup for two of the HTML standards for HTML 4.01 are shown here (strict and transitional):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

< >

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

< >

CSS

Part 13:
Reading a Switch with SD Card Web Server and Ajax

Part 14:
Reading Inputs with Ajax and XML

Part 15:
Analog Value Displayed on Gauge

Part 16: Inputs and Outputs (I/O)

Part 17:
Accessing HTML Tags with CSS and JavaScript

Part 18: CSS for Positioning, Sizing and Spacing

Summary and Conclusion

Cascading Style Sheets (CSS) are used to style the HTML elements on a web page. This means that CSS can be used to change the font style, size and colour. The size position and colour of various HTML elements can be changed using CSS.

In our examples in this tutorial, the CSS was included as part of the HTML file and put between opening and closing **style** tags.

Normally a website will have a separate style sheet file (e.g. style.css) containing all the CSS. This allows the same style sheet to be used to apply styles to all the pages on the website.

Because we have mostly been using single web pages on the Arduino web server, it was easier to include the style in the web page. If more than one web page is needed using this method, then the same styles would need to be copied to the top of each page. The advantage of using a separate style sheet means that the style sheet only needs to be included in the top of each HTML page (the style sheet is included using a single line of markup).

In the relatively slow Arduino, a faster response will be obtained by including the CSS styles in the top of the web page when hosting a single web page. This is because only one HTTP request is then needed to get the page and CSS.

JavaScript and Ajax

We used JavaScript to implement Ajax for sending data to the Arduino from the web page and getting data back from the Arduino behind the scenes.

Ajax enables parts of the web page to be updated. This reduces the amount of data that needs be sent from the Arduino making the updates faster because the entire web page does not need to be reloaded every time new data is to be displayed on it.

Another advantage of using Ajax is that the web page does not flicker when data is updated as occurs when the entire page is refreshed.

HTML 5 Canvas

HTML 5 introduces the **canvas** tag. The canvas HTML element allows JavaScript to draw to the canvas area of the web page.

We saw the HTML canvas with JavaScript being used by the [Gauge component written by Mykhailo Stadnyk](#) in [part 15](#) of this tutorial.

Only modern web browsers that support the relatively new HTML 5 standard will be able to run the gauge example.

Conclusion

Most of the technologies used to create a web page are rather large topics and have entire books dedicated to each one.

We have used a small subset of each of these technologies that have hopefully given you a start in creating your own Arduino web server and web pages.

How far you wish to pursue each of these subjects depends on what you want to implement on your web server and how good you want to get at each technology.

There are many resources available for web page design covering HTML, CSS, JavaScript and Ajax such as books and web sites – just do some searching on the Internet to find more detailed tutorials and references.

Related Resources and Projects

These projects and resources are related to this Arduino web server tutorial:

- [How to add a second gauge to an HTML web page](#) – part 15 of the web server tutorial shows how to add a single gauge to a web page, this tutorial shows how to add a second gauge to a web page.
- [Web server with two temperature gauges](#) – uses two of the gauges from part 15 of this tutorial to display two temperatures measured by the Arduino.
- [Arduino MEGA web server that controls 24 outputs](#) – an Arduino MEGA 2560 is used to control 24 Arduino outputs from a web page.
- [How to connect an Arduino to the Internet](#) – the web server tutorial demonstrates how to set up a web server on a local network. This tutorial shows how to connect an Arduino to the Internet so that it can be accessed from any device that has a web browser, such as a smart phone, tablet or PC.
- [Programming and using an Arduino Ethernet board](#) – an Arduino Ethernet board is like an Arduino Uno, but has built in Ethernet and no USB port.
- [How to control an LED from a web page and push button](#) – controlling the same LED from a web page checkbox or button and a physical push-button switch connected to the Arduino.
- [Connecting and testing an Arduino Ethernet shield](#) – how to test an Arduino Ethernet shield or other Arduino Ethernet enabled board.
- [Testing an Arduino Ethernet shield SD card](#) – testing the SD card socket on an Arduino Ethernet shield or other Arduino board that has an SD card.
- [Arduino programming course](#) – learn more about the Arduino programming language to help you write better sketches.

[← Go back to Part 18](#)

Login

Comments

There are no comments posted yet. [Be the first one!](#)

Post a new comment

Enter text right here!

Comment as a Guest, or login:

Name

Displayed next to your comments.

Email

Not displayed publicly.

Website (optional)

If you have a website, link to it here.

Subscribe to

Submit Comment

[Arduino](#)

[Pinout](#)

[About](#)

[Contact](#)

[Donate](#)

[Privacy Policy](#)

[Amazon Adverts](#)

© 2012 – 2021, Starting Electronics