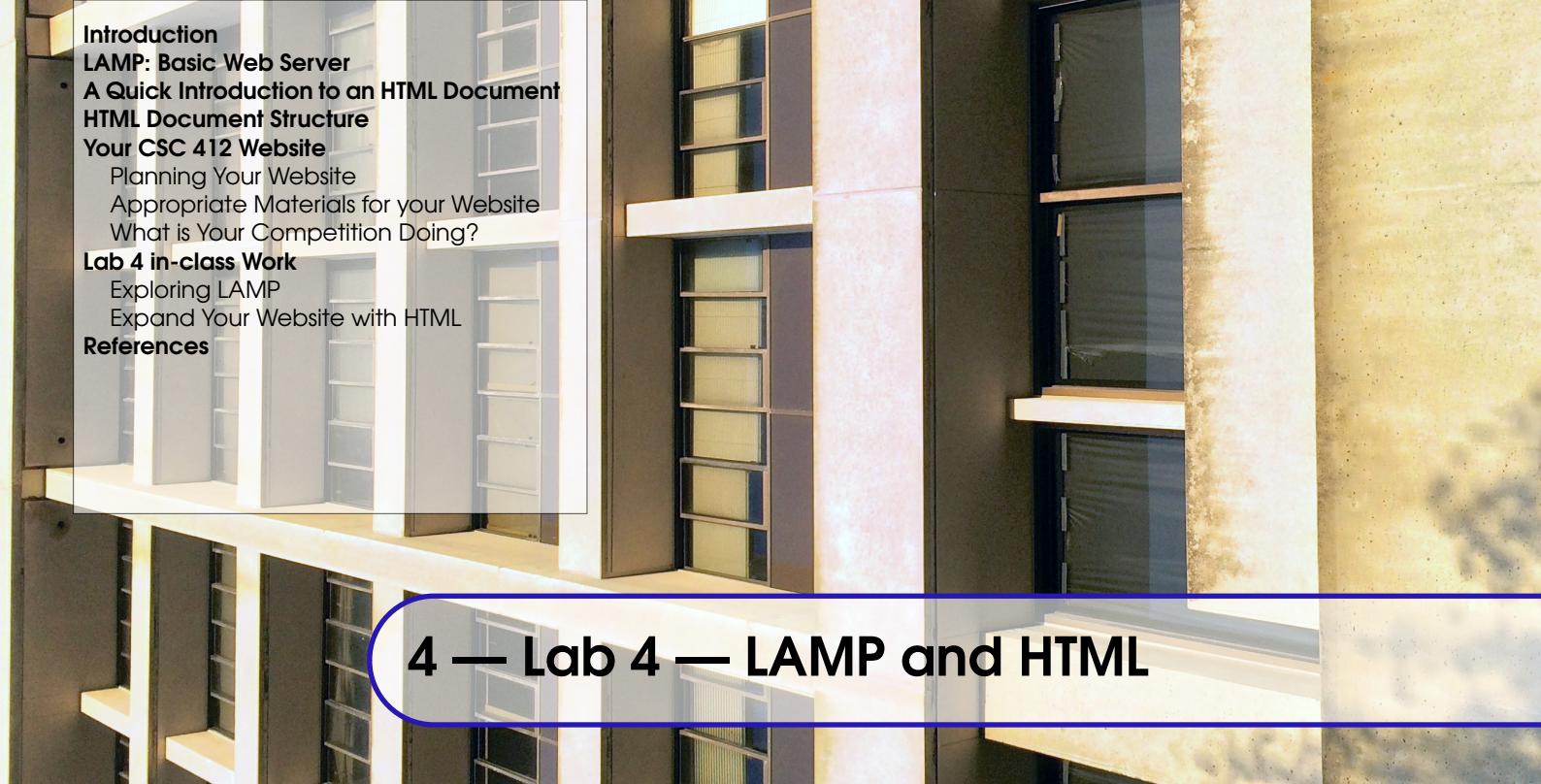


- 
- Introduction
  - LAMP: Basic Web Server
  - A Quick Introduction to an HTML Document
  - HTML Document Structure
  - Your CSC 412 Website
    - Planning Your Website
    - Appropriate Materials for your Website
    - What is Your Competition Doing?
  - Lab 4 in-class Work
    - Exploring LAMP
    - Expand Your Website with HTML
  - References

## 4 — Lab 4 — LAMP and HTML

### 4.1 Introduction

In this lab you will be introduced to the web server, begin learning about web scripting, and explore HTML.

### 4.2 LAMP: Basic Web Server

Technology	Description
Linux	The operating system that the stack operates in. It represents any UNIX based operating system such as flavors of Linux (e.g. RedHat or Ubuntu).
Apache	The web server application that coordinates requests for web page information from the Internet, and delivers web page content to the Internet.
MySQL	The database application from where information, such as data for dynamic web pages, is stored and retrieved. This can be any SQL-Based database system
PHP	An interpreted scripting language. The P may also stand for Perl or Python. In all cases the P represents a scripting language that generates dynamic page content.

Table 4.1: The components of a LAMP Stack.

LAMP is an acronym that stands for Linux, Apache, MySQL, and PHP. This acronym describes what is currently the most popular software solution stack that web servers run to provide users a way to deploy dynamic web pages.

A dynamic web page is a web page whose content changes based on underlying data. For

example, when you go to a site to shop, the contents of the page change based on what you are shopping for.

A software stack, usually just called a stack, is a collection of programs that work together to perform a task. In the LAMP stack, each component has a purpose in the process of web serving, which are described in Table 1. Some of the particular technologies may be replaced with others, and the acronym changes. For instance, a similar stack running on a windows server is called a WAMP stack.

### 4.3 A Quick Introduction to an HTML Document

HTML is an acronym for Hyper Text Mark-up Language. A mark-up language is a system for notating documents in a way that the mark-up can be distinguished from the text. In the case of HTML, tags are used to notate documents, and are contained in angle brackets <>. HTML documents are composed of text and tags.

HTML tags describe to the web browser how parts of the HTML document should be interpreted. Frequently, tags appear in pairs, made up of an opening (or start) tag and a closing (or end) tag. Closing tags begin with a forward slash. For example, the paragraph tag looks like:

```
<p> hello world !</p>
```

The text between the tags is part of the same paragraph element.

### 4.4 HTML Document Structure

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title></title>
5          <meta http-equiv="Content-Type"
6              content="text/html; charset=UTF-8">
7      </head>
8
9      <body>
10         <div>TODO write content</div>
11     </body>
12 </html>

```

Figure 4.1: Skeleton HTML

Figure 4.1 shows a skeleton of an HTML document. On line 1, the DOCTYPE declaration, denoted by the exclamation mark, should be the first line in the document. This tells the browser what version of the mark-up language the page contains. Make sure you have no blank lines above this line, as that can confuse some browsers into displaying blank pages.

Line 2 contains the opening html tag, and the rest of the document is contained within in this element. Line 3 through line 7 is the head element that contains information about the HTML document, or metadata. Information about styles and JavaScript can go here. On lines 5-6, information about the content of the document is stored, specifically how the document is encoded. Lines 9-11 contain the body element, which is what appears in the browser window when it renders the web page.

## 4.5 Your CSC 412 Website

### 4.5.1 Planning Your Website

Before you begin any software project, it is good practice to plan, at least a little bit. The goal of this class is for you to leave class with a personal website that introduces you and your work to the world, gives you a place to put your resume if you wish. This type of “business card” site could come in handy when you are applying for jobs.

Page	Filename	Description
index	index.html	Your "home" page. This is the page that the web browser looks for first when publishing your site to the internet.
about	about.html	This page will contain a brief synopsis about yourself, and perhaps an appropriate photo( if you wish).
current work	current.html	This page will contain links to your current work.

Table 4.2: Minimum pages for your CSC 412 website.

For this class, your site will have at minimum three pages as described in Table 4. You are encouraged to add other pages as you see fit. If you use PHP to create dynamic content, the filenames in Table 4 may have .php extensions instead of .html.

### 4.5.2 Appropriate Materials for your Website

Be careful to put only material on your site that you would want a potential employer to see. Your site should be considered a professional exercise. Good guidance about what is appropriate for a professional site can be found at

<http://www.salisbury.edu/careerservices/students/resumes/PersonalWebsite.html>  
It is strongly recommended that you take a look at this before you begin your project.

### 4.5.3 What is Your Competition Doing?

Take some time to browse around the web, looking at other people’s business card websites. Especially look for people in the industry you may be interested in working. By seeing what other people are putting on their pages, you can gauge what is appropriate for your page.

## 4.6 Lab 4 in-class Work

### 4.6.1 Exploring LAMP

In a LAMP stack, the components of the web server run in Linux. As you discovered in Lab 1, your account on the AWS-Server is running Linux. Recall that in Lab 2 you created a webpage on your AWS-Server account that is being served by this web server.

#### Apache Server

1. Using a terminal app, log into your account on the AWS-Server.
2. From the command prompt type:

`ps -Af` 

Quite a few lines should scroll by. Each line describes a current process running in the background of the operating system. The first column is the user id (UID), the second column is the process id (PID), and the last column is the last command run.

```
csc412@thecity:~$ ps -Af | grep apache2
www-data 4789 6905 0 02:31 ? 00:00:01 /usr/sbin/apache2 -k start
www-data 4976 6905 0 03:08 ? 00:00:01 /usr/sbin/apache2 -k start
root 6905 1 0 Jan09 ? 00:05:58 /usr/sbin/apache2 -k start
www-data 7312 6905 0 09:21 ? 00:00:01 /usr/sbin/apache2 -k start
www-data 7313 6905 0 09:21 ? 00:00:01 /usr/sbin/apache2 -k start
www-data 7375 6905 0 09:33 ? 00:00:01 /usr/sbin/apache2 -k start
www-data 8723 6905 0 13:57 ? 00:00:00 /usr/sbin/apache2 -k start
csc412 11658 11461 0 22:22 pts/3 00:00:00 grep apache2
www-data 12163 6905 0 Sep23 ? 00:00:04 /usr/sbin/apache2 -k start
www-data 18190 6905 0 Sep23 ? 00:00:03 /usr/sbin/apache2 -k start
www-data 19845 6905 0 Sep24 ? 00:00:03 /usr/sbin/apache2 -k start
www-data 30868 6905 0 Sep24 ? 00:00:02 /usr/sbin/apache2 -k start
csc412@thecity:~$
```

Figure 4.2: apach2 processes currently running.

3. From the command prompt type:

`ps -Af | grep apache`

The unix command grep apache searches for the string "apache" in each line that is outputted by the ps -Af command. The output of the ps -Af command is piped into the grep apache command using the vertical bar. Each line that contains the search string is output by grep onto the system console. The final result is a list of only the current processes that involve apache. You should even see that one of the lines is your user ID with the grep apache command, since you used the string "apache" in your search.

Notice that there are a number of processes running apache. Each one of the processes is running the apache server, waiting to serve web pages from the AWS-Server to a browser that requested it over the Internet.

4. From the command prompt, type:

`apachectl fullstatus | less`

The apachectl program controls the Apache web server on the system. You can also use apachectl to show you the current status of the apache web server on the system. By piping the output of the apachectl fullstatus into the less, you can scroll through the output using your arrow keys. To get out of the file, type q.

5. For more information about less, look at the man page entry for it by typing:

`man less`

## MySQL

Like Apache, the MySQL database also runs in its own process, waiting to service requests for access to the database. The d suffix to mysql indicates that this is a daemon running. A daemon is a process that runs in the background, providing a particular service.

6. Using the technique from above, see what processes are running the MySQL daemon, mysqld( use lower case only for the name).

**PHP**

PHP is a little different than Apache or MySQL in that it is not always running in the background. Rather, PHP is an interpreted scripting language. This language is interpreted through a special module in the Apache server, mod\_php. As the web server parses an HTML file to begin outputting it to a browser via the Internet, any PHP code in that file is sent to the PHP interpreter. The interpreter's output is then output to the Internet instead of the PHP code itself.

PHP can also be run from the command line using a command line interpreter.

7. From the command line, start the PHP command line interpreter by typing:

php

you will not see any prompt, the cursor will just be sitting on the next line.

8. Next, type:

<?php echo "\n\nHello World!\n\n"; ?>

Next you will see how the web server interprets PHP.

9. Go to your public\_html directory

10. Using your favorite editor create a file called demo.php. Put the following lines in the file:

```
<?php  
    phpinfo();  
?>
```

Save the file and exit the editor.

11. Go to your web browser and browse to <http://csc412sfsu.com/~username/demo.php> where username is your username.

Notice how much has been displayed on the page, and all from a single php command. The *phpinfo()* function was called by the web server's PHP interpreter, and the output of that function produced all this text on the web page.

#### **4.6.2 Expand Your Website with HTML**

In Lab 2 you created the basic skeleton for your website using NetBeans. In this part of the lesson, you will expand your personal web site.

12. Open NetBeans and the project that you created in Lab 2.

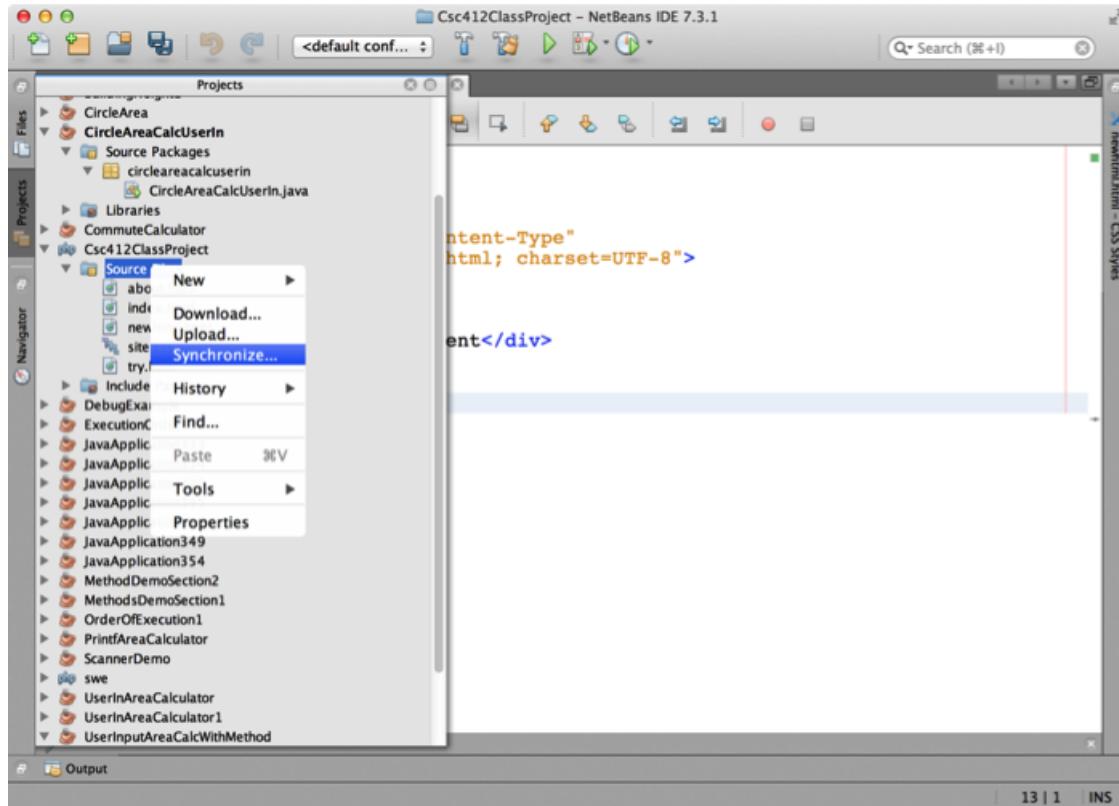


Figure 4.3: Synchronize your files

13. Before you start, you should be sure your remote files are synchronized to the files you have on your computer. Right click on the source files tab within your project, and select Synchronize, Figure 4.3

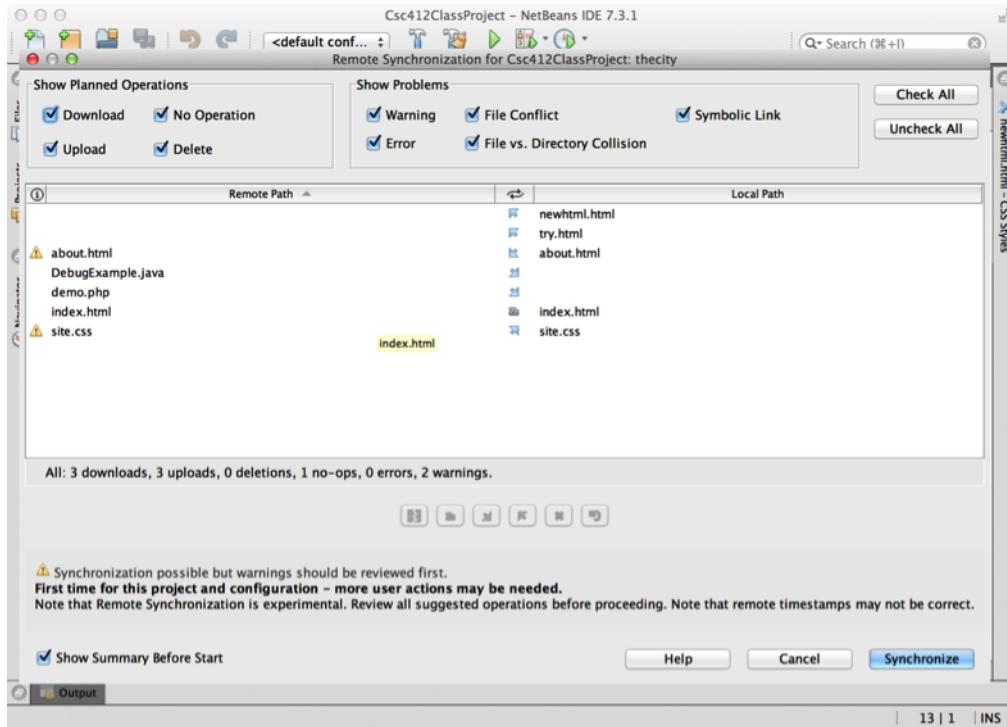


Figure 4.4: Make sure your synchronization is correct

14. Your computer and the AWS-Server will communicate, and NetBeans will figure out which files are newest between the two sites. Newer files on your computer will be uploaded to the AWS-Server, and newer files on the AWS-Server will be downloaded to your computer. Make sure that everything looks correct (which files are being synchronized, and to where). When you are satisfied, click the Synchronize button (Figure 4.4). Follow the prompts until your files are synchronized.
15. Using the example in Figure 4.1, put any missing tags into your index.html and about.html files. Do not include the paragraph element containing the TO DO text.
16. Using NetBeans, create a new html file in your project named current.html by selecting `File New File other HTML`.
17. Using the about.html file as a template, add content to the current.html file.
18. Edit your index.html file, and add a link to your new current.html page. Use the existing link as an example.
19. Explore <http://www.w3schools.com/html/default.asp> and try adding some elements to your web pages. Be sure to add somewhere on your home page that you are a student at SFSU, and this site is a practice site for CSC412 in Spring 2018.

#### **4.7 References**

Server Rankings

PHP on the command line

PHP

HTML Intro

[http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

[http://en.wikipedia.org/wiki/Markup\\_language](http://en.wikipedia.org/wiki/Markup_language)

<http://www.w3.org/TR/REC-html40/struct/global.html>

HTML Doctypes

Document Structure

What to put on your website

netbeans with HTML 5 chrome browser

Daemons