

Creating a Web Site

Instructor: Daniel Chong

Email: dchong@mybpl.org

Course Objective: To gain a higher level of understanding of both HTML and CSS and learn how to deploy our web pages to the wider Internet.

Topical Outline:

1. Advanced HTML Elements and Attributes
2. Styling Pages with CSS
3. Getting a Domain & Web Hosting Services
4. File Transfers

In the first class offered here at the library, *Designing Web Pages with HTML*, we learned the basic frameworks for putting together a web page. We got a lot things to appear onto our web pages, but we didn't make a cohesive, web ready page. Tonight we will learn how to take what we learned in that first class, and build upon that foundation to make a web site that is able to be deployed to the web. We will add things like navigation links to other pages within our website, and sidebars with information about our website. We will also learn how to create a style sheet with CSS to make visual, elegant changes to our website. Finally we will cover different web hosting services, and also how to transfer our source files to those services using File Transfer techniques.

Advanced HTML Elements and Attributes

The first element we will cover is called the Navigation Element. This element is called with the `<nav></nav>` tags. What this does is it essentially creates us a special div element that contains links to **INTERNAL** web pages. This means that this is where we would keep the links that take us to pages within our website as a whole.

For example, consider you have a website to showcase your photography. Would we just throw every picture you have ever taken on one page, along with all the information about how to buy a print? Or would you create separate pages, one for each photo shoot and one for a store to purchase photos? From a design standpoint it would probably function better with multiple pages to hold all that information, but it's also a double edged sword. If you create too many webpages then users can get bogged down and lost, eventually just giving up and never visiting your site again.

A good practice for designing a web page is to keep information displayed on a page to 2.5 "Screen Lengths." What this means is that if a user has to go through more than 2.5 screens of information while scrolling you should probably think about how to break it up. This is not to say that you can't have more information than that on one page, it's just that you have to think about what you are presenting and if someone will lose interest quickly.

Navigation Element Example:

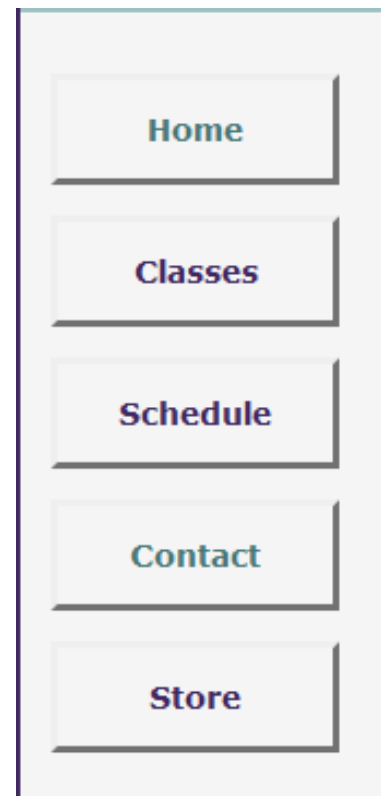
```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="classes.html">Classes</a></li>
    <li><a href="schedule.html">Schedule</a></li>
    <li><a href="contact.html">Contact</a></li>
    <li><a href="store.html">Store</a></li>
  </ul>
</nav>
```

As you can see, within the `<nav>` tags we have a nested unordered list. This is where the links to other pages would be seen. The reason they are stored in an unordered list instead of an ordered list is simply because the order of these links does not really matter. If we used an ordered list it would imply a hierarchy of more or less important links. Also when linking from the main page to a secondary page you do not need to include the whole URL. For example, say I ran www.danielchong.com and I had a secondary page called www.danielchong.com/classes.html, to link to that page I would only need to have `Classes`. This assumes you have all your files stored in the same place.

The links would look something similar to this, when we finish the class we will have them styled to look more like this example, but the basic idea is the same, we would create a list of links that guide us through the internal web pages of our website.

Another good web practice is to NEVER link to the page you are currently on. For example you are sitting on the home page, if a user clicks on the home page button they will expect to be taken to the home page, but instead the page will just reload. This can lead to confusion or just a general dislike of a web site design and has been proven in many, many focus groups. Instead we just leave the link out of that page, thus ensuring they can never reload the page that they are currently on unless they specifically mean to, using the refresh button.

In the same vein, it is also good practice to leave users what is called bread crumbs, or a path back to where they originally landed on your website. This can be done by changing the colors of navigation links visited or changing the color of the link on the current page, this way a user has an easier time navigating through your website.



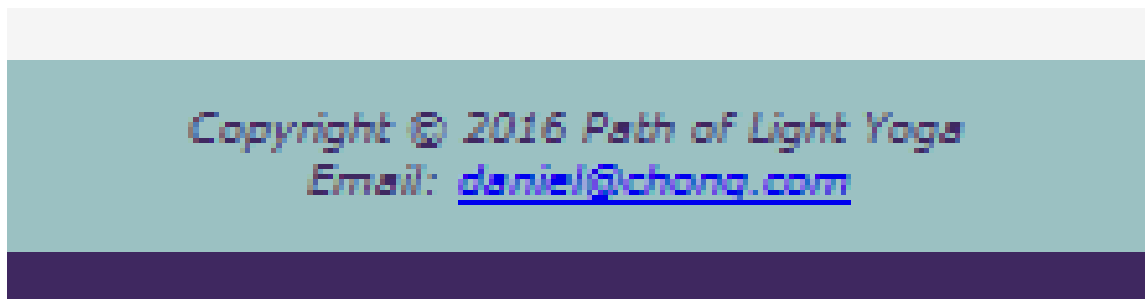
Linking to this is one of the most powerful tools you can utilize within the web. Not only can you navigate to internal pages of your website, but let's say that you want to put some contact information within your page. It would be a good idea to have someone be able to access your business email, and you can link directly to that email address using the **mailto:** attribute. This allows a user to click that link and immediately begin writing a letter to the designated address. The one fall back of this is that the user needs to have a default mail client set up, for example Outlook. If they do not have a default mail client, then the link will not work, although people will still be able to see the email address like normal text that can be copied and pasted into a web based email service.

To create an email link you would use the following syntax:

```
<a href=mailto:dchong@mybpl.org> Daniel Chong – Computer Services </a>
```

Usually these things are kept in the footer element, you would see source code similar to the following syntax:

```
<footer>
    Copyright &#169; 2016 <span class="studio"> Path of Light Yoga </span>
        <br />
    Email: <a href="mailto:daniel@chong.com"> daniel@chong.com </a>
</footer>
```



You can also link to audio and video, this can be done using the `<audio></audio>` tags respectively.

Audio Syntax:

```
<audio controls>
    <source src="horse.ogg" type="audio/ogg">
    <source src="horse.mp3" type="audio/mpeg">
    Your browser does not support the audio element.
</audio>
```

Video Syntax:

```
<video height=312 width= 540 controls>
    <source src="movie.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
    Your browser does not support the video element.
</video>
```

When we are creating the content that goes on our webpage we have to be conscious of formatting, the following is a list of formatting elements used in HTML5.

Formatting Elements HTML5

Tag	Description
<u></u>	Defines bold text
<u></u>	Defines emphasized text
<u><i></u>	Defines italic text
<u><small></u>	Defines smaller text
<u></u>	Defines important text
<u><sub></u>	Defines subscripted text
<u><sup></u>	Defines superscripted text
<u><ins></u>	Defines inserted text
<u></u>	Defines deleted text
<u><mark></u>	Defines marked/highlighted text
<u><abbr></u>	Defines an abbreviation or acronym
<u><address></u>	Defines contact information for the author/owner of a document
<u><bdo></u>	Defines the text direction
<u><blockquote></u>	Defines a section that is quoted from another source
<u><cite></u>	Defines the title of a work
<u><q></u>	Defines a short inline quotation

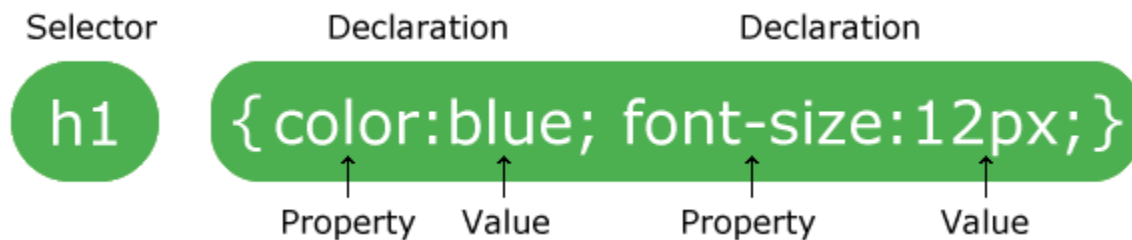
New Structure Elements for HTML5



- `<header>` - Defines a header for a document or a section
- `<nav>` - Defines a container for navigation links
- `<section>` - Defines a section in a document
- `<article>` - Defines an independent self-contained article
- `<aside>` - Defines content aside from the content (like a sidebar)
- `<footer>` - Defines a footer for a document or a section
- `<details>` - Defines additional details
- `<summary>` - Defines a heading for the `<details>` element

Styling Pages with CSS

CSS stands for Cascading Style Sheets, it is used to write rules on how we want our webpage to appear visually. Each rule consists of a Selector and a Declaration Block. The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, these are separated by a colon. Declaration blocks are contained by curly braces {}.



This example styles the header(1) element to have blue font color, and a font-size of 12px. If we were to write this out we would want to make it a little more readable so it would look something like this:

```
h1{
    color: blue;
    font-size:12px;
}
```

There are a few ways that we could go about implementing CSS into our webpages. In the first class I talked about inline styling, where we actually wrote the rule for our html element within the line it was on. The next way we can go about getting CSS into our page is called Internal styling.

This is done by calling the Style Element, with the <style></style> tags. We always do this within the Head element, so at the top of our page. We do this because when a page loads, it goes sequentially through to the bottom, so we want to make sure that the browser knows that these are the rules we want to style with and they must appear before any our main elements.

The syntax for this would look like this:

```
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

The final way to add CSS into our page is by linking to an external CSS file. We would write our rules like we normally would, but instead of them being on the page or inline with an element we would say, “Hey – look over here, we have a stylesheet for you to reference.” You can do this with the following syntax:

```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

If you’re looking for a quick and easy way to add CSS into your webpage you can access the W3.CSS templates for free. These can be found at <https://www.w3schools.com/w3css/default.asp> and can be called using the following external CSS syntax.

```
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
```

CSS can be one of the most powerful tools you can use to build a website. For a full list of properties and values you can go to <https://www.w3schools.com/css/default.asp>, which will also provide you with comprehensive tutorials to show you how to do advanced styling rules, such as drop boxes and slideshows.

Getting a Domain & Web Hosting Services

The first step in publishing our website to the wider web is getting a domain name for our website. A domain name is the name of your website. For example Google’s domain name is google.com (just the underlined part, a .com is a domain name extension.) This is used for finding and identifying computers on the Internet. Computers normally use IP address, in fact they still do, but the domain name converts that string of numbers into a Fully Qualified Domain Name, this way we do not have to remember the numbers, but rather the domain.

To have access to a domain name you will have to get one registered. There are many websites that offer this service such as GoDaddy or Google Domains. When you register a domain name, you are paying a yearly fee to keep that domain in your name. If you let the payment lapse you will lose the domain, so it is important to keep your website in good standing. If you can’t find the .com domain you would like consider using a different domain extension such as .net or .io.

Now that you have your domain name registered you will want to look into a Web Hosting Service. Websites are hosted, or stored, on special computers called servers. If someone wants to visit your website all they have to do is type in your domain name and their computer will connect to the server that your website “lives” on. When you pay for a Web Hosting Service you are paying rent for room on those servers. Some of the most popular Web Hosting Services include: GoDaddy, HostGator, DreamHost and InMotion. There are free hosting websites such as Weebly or Wix that allow you to create a domain tied to their website. For example if you were to use Weebly’s free hosting service your domain name would look like: **exampledomian.weebly.com.**

It is possible to self-host your website but it would require a super powerful computer running 24/7 and a business class Internet service. While this is not the route I would recommend, I just wanted to let you know that it was possible if that was something you would like to look into.

File Transfer

Now that we have our domain and web hosting service chosen we are ready to upload our files to the servers our website resides on. Depending on the Hosting Service that you have chosen, you may have the option to do a browser upload. Specifically if you are using GoDaddy, this is called the File Manager. It would be located under “Manage Website Builder.” This will open up the file manager for any domain that you own. From here you can look at source code of files already uploaded, or upload files to be linked to your domain. If your Web Host offers this it can be a convenient way to upload files, but sometimes you can’t do a bulk upload. This is where the File Transfer Protocol becomes a handy tool to use.

File Transfer Protocol or FTP for short. One of the oldest protocols in use today, and is a convenient way to move files around. An FTP server offers access to a directory, with sub-directories. Users connect to these servers with an FTP client, a piece of software that lets you download files from the server, as well as upload files to it. One of the most popular FTP clients is called Filezilla. This is a free tool that will allow you to connect to the server where your domain lives and upload files directly to it. It is essentially a file directory that allows you to interact with the server. It is a hierarchical folder structure makes it very similar to navigating. Windows Explorer or Finder.

To connect to the website server to upload files you will need to know the following:

Field	What to enter...
Host	Your domain name or IP address (cPanel / Plesk / Web & Classic)
Username	Your FTP username (cPanel / Plesk / Web & Classic)
Password	Your FTP password (cPanel / Plesk / Web & Classic)
Port	21

On the following page is a screen shot of the Filezilla Client for our website here at the library

mybpl@192.254.163.135 - FileZilla

File Edit View Transfer Server Bookmarks Help

Host: Username: Password: Port: Quickconnect

Status: Initializing TLS...
 Status: Verifying certificate...
 Status: TLS connection established.
 Status: Logged in
 Status: Retrieving directory listing of "/public_html/images/slides"...
 Status: Directory listing of "/public_html/images/slides" successful

Local site: C:\Users\dchong\
 Remote site: /public_html/images/slides

Filename	Filesize	Filetype	Last modified
..			
.aws		File folder	6/5/2018 7:04:19 PM
.idlerc		File folder	6/30/2018 12:48:59...
.thumbnails		File folder	8/11/2018 9:16:44 ...
.VirtualBox		File folder	7/14/2018 4:56:19 ...
AppData		File folder	9/13/2018 3:30:14 ...
Application Data		File folder	
Contacts		File folder	9/19/2018 1:16:03 ...
Cookies		File folder	
Desktop		File folder	11/18/2018 3:41:40...
Documents		File folder	11/18/2018 11:26:3...
Downloads		File folder	10/17/2018 1:08:58...
Favorites		File folder	9/19/2018 1:16:03 ...
IntelGraphicsPro...		File folder	11/28/2018 12:52:1...
Links		File folder	9/19/2018 1:16:03 ...
Local Settings		File folder	
Music		File folder	9/19/2018 1:16:03 ...
My Documents		File folder	
NetHood		File folder	
Pictures		File folder	9/19/2018 1:16:03 ...

9 files and 31 directories. Total size: 5,089,375 bytes

Filename	Filesize	Filetype	Last modified
..			
1kbooks.png	152,036	PNG image	8/9/2018 12:09:...
BackonTrack.png	217,421	PNG image	9/10/2018 10:4:...
BB-webslide.png	110,514	PNG image	10/25/2018 4:2:...
ExamCram.png	153,087	PNG image	11/19/2018 2:1:...
GetReady.png	189,720	PNG image	9/30/2018 4:40:...
HOH WEB.png	131,795	PNG image	10/25/2018 4:2:...
HomeForHolidays.png	435,962	PNG image	11/12/2018 10:...
HowDoesItGlow.png	894,981	PNG image	9/30/2018 4:40:...
ILMusic.png	862,554	PNG image	11/12/2018 10:...
JingleBellExpress.png	131,332	PNG image	11/12/2018 3:3:...
MainStreetLL.png	295,939	PNG image	11/12/2018 5:2:...
MainStreetLL1.png	295,939	PNG image	11/12/2018 5:2:...
Strangell.png	678,047	PNG image	11/12/2018 10:...
TurkeyWeek.png	72,063	PNG image	11/12/2018 3:3:...

14 files. Total size: 4,621,390 bytes

Server/Local file	Direction	Remote file	Size	Priority	Status

Queued files Failed transfers Successful transfers

Queue: empty