

LEARNINGTHEBASICS

PRACTICAL DESIGN CONCEPTS, XHTML & CSS

Copyright 2004 by Telian Adlam. All Rights Reserved.

www.buniek.com

I understand that not everyone can afford to have a webmaster on call 24/7 so I created this ebook to help the complete novice learn the basics to standards compliant web design. I receive a lot of questions about updating and making site changes, so this ebook should get you off on the right foot so you can easily update or make changes to your own HTML based website and of course links are provided for further learning.

Some ground rules:

This ebook may be freely redistributed with the following restrictions:

You may not sell or receive compensation for this ebook.

You may not change any of the content in this ebook.

You may not claim owner/authorship of any content within this book.

You may not reprint or publish this content elsewhere without prior permission.

You do not have any rights (implied or otherwise) to this ebook or its content.

You can also read this complete ebook online – as well as see how colors can affect the feel of a site by visiting <http://www.buniek.com/resources/css-xhtml/>

Did I leave something out? Is there something else you would like to see or learn? How can I make this ebook better? I welcome all questions, comments, concerns, or even if you just want to say “hey there and hello” – please feel free to e-mail me at teli@buniek.com

Section 1 – Basic (and Practical) Design Principals

1.1 [Selecting your colors](#) [3-4]

1.2 [Selecting your font\(s\)](#) [4-5]

1.3 [Using graphics](#) [6-7]

1.4 [Page flow and layout](#) [7-8]

Section 2 – Basic HTML (XHTML)

2.1 [What you need to get started](#) [9]

2.2 [What is HTML and XHTML](#) [9]

2.3 [Which one should you use](#) [9]

2.4 [Basic structure](#) [9-10]

2.5 [DOCTYPE Declaration](#) [11]

2.6 [The head](#) [11-12]

2.6.1 [Meta tags](#) [12-14]

2.6.2 [Linking external files](#) [14-15]

2.7 [The body](#) [15]

2.7.1 [Some basic tags](#) [15]

2.7.2 [Some basic tag attributes](#) [16-17]

2.7.3 [Putting it together](#) [17-19]

2.8 [Saving your XHTML document](#) [20]

Section 3 – Basic CSS

3.1 [Getting started](#) [21]

3.2 [Code layout – how to write correct CSS](#) [21-22]

3.2.1 [Some basic selectors and their properties](#) [22-23]

3.2.2 [Putting it together](#) [23-26]

3.3 [Saving your CSS document](#) [26-27]

Section 1

Basic (and Practical) Design Principals

1.1 Selecting your colors

Selecting colors for your website is an important task, one, that if done well, will produce a pleasant and inviting environment for your visitors. When selecting your colors, keep in mind that certain colors evoke certain emotions in people (the psychology of color).

To start, it's a good idea to use moderate or subdued colors, and then you can begin to branch out and explore bold unique colors once you have a handle on the subject. For starters, you can find a lot of great colors around you, even in nature.

One of my personal favorite seasons is fall, so I've selected the picture aptly titled autumn leaves as an example:



	#5B5915 - dark olive green Text and content
	#E06720 - orange red Links and other highlights
	#FEF3BD - creamy almond Background

Photo by Kathryn Gruber (<http://www.sshnug.com/>)

Using the eyedropper tool from photo editing software, such as Photoshop, choose the foundational or base colors that will be used throughout the page and make a note of their [hexadecimal](#) values (the 6 character combination of numbers and letters).

It's a good idea to make a small design chart and start with your color selections:

Color Description	Hex Code	Notes:
Dark Olive Green	#5B5915	Body Text and content
Orange Red	#E06720	Links and other highlights Used sparingly throughout page
Creamy Almond	#FEF3BD	Page background

Some things to keep in mind when selecting colors:

- Limit your color palate to three or fewer colors until you have a firm understanding of how colors work together and compliment each other. Too many colors or the wrong colors can take away from the overall theme and content of your page.
- Be careful and make sure your colors work well together – a rule of thumb is to step back and look at the color selection you've chosen and ask yourself whether the combination works well together and whether or not it will send the appropriate message for your page. If you feel you are completely devoid of any taste in color what so ever, don't be afraid to ask someone for help or to get someone else's opinion.
- Colors are usually placed into one of two groups: warm or cool. Warm colors are reds, oranges and yellow and they tend to have a cozier feel, but using too many or the wrong combination can make your page seem cramped and heavy.

Cool colors are blues, greens, and purples and they feel a lot more open and less restricting, but using too many or the wrong combination can make your page seem sparse and unwelcoming. Make sure to space your content accordingly.

=== **Sidebar** ===

Tip: Look at an image and think about how it makes you feel just by looking at it. If it evokes the type of emotion that you want for your page, ask yourself which of the colors are doing it, then start choosing your color palate.

If you don't have access to photo editing software or an eyedropper tool, you can visit <http://www.buniek.com/resources/html-colors.html> for an html color chart and the corresponding hexadecimal values.

Some basic hex values:

White: #FFFFFF

Black: #000000

Red: #FF0000

Green: #00FF00

Blue: #0000FF

Dark Gray: #333333

Gray: #888888

Light Gray: #CCCCCC

=== **End sidebar** ===

1.2 Selecting your font(s)

You would be amazed at how much emotion and style you can convey using only fonts. Choosing your fonts should not be taken lightly as there is also a psychology of typography that most people are not even aware of.

When you start your font selection, it's a good idea to divide up your site into sections – logo/name, headings, body, links, and "other". The other section can be comprised of specials, deals, advertisements, etc. Instead of using a different font for the "other" section, try using the same font as your body and simply change the color or weight.

Choose two to three fonts that compliment each other well throughout your page. When choosing a titling font for your logo or branding try to choose a font that speaks about your business, what you do, or what message you want your page to send.

Although there are many groups of general font styles, the two most common groups of fonts that you see are [serif](#) fonts and [sans-serif](#) fonts. An example of a common serif font would be "Times New Roman" and a sans-serif font would be "Verdana".

Some things to note about fonts:

- Fonts vary on different operating systems and browsers – instead of only specifying one font, have a list of closely related fonts and a generic font that would be acceptable for a browser to use when visiting your page. The only one to guarantee users will see the same font on their browser is to use images for the font, but use images for fonts sparingly. It's not a good idea to make your entire page one big image just so it can be displayed in 10 pt [Jokerman](#).
- Limiting the number of fonts you use throughout the site provides more uniformity and professionalism.
- There are wide assortments of fonts out there ranging from western themes to formal script to gothic to old English, and so much more – get creative with your logo or branding and let the font speak about who you are and what you are representing.

- Fonts offer many choices and options such as weight, size, and color – experiment and have fun.
- Serif fonts tend to be more classical and formal. Sans-serif fonts tend to be more modern and streamlined.

Example:

The only way to guarantee a font will show up even if it isn't available on your visitors' computer is to use an image – such as for titling or logo text:



60 pt [Futura](#) (sans-serif font)

Once you've chosen your fonts and their uses, it's best to create a chart for reference:

Section	Font	Size
Titling/Logo	Futura	60pt
Headlines	Arial, Verdana, Geneva, sans-serif	Main headings: 1.4em Secondary headings: 1.2em Third level headings: 1.0em
Body and Content	Verdana, Arial, Geneva, sans-serif	0.8em
Links and special text	Verdana, Arial, Geneva, sans-serif	0.8em

=== **Sidebar** ===

The most common **serif** font grouping for the web is "Times New Roman, Times, serif" and more recently "Georgia, Times New Roman, Times, serif"

The most common **sans-serif** font grouping for the web is "Arial, Helvetica, sans-serif" and more recently "Verdana, Arial, Geneva, sans-serif"

Most visitors will have at least one of the fonts in these groups on their computer.

To learn which fonts are usually installed on the different operating systems, visit <http://www.buniek.com/resources/fonts.html>.

To learn more about choosing typography and how the different fonts can affect your visitors, visit <http://www.buniek.com/links/typography.html>.

Visit <http://www.buniek.com/links/fonts.html> for a wide variety of available fonts for Windows and Macintosh computers.

=== **End Sidebar** ===

1.3 Using graphics

When talking about graphics I am referring to an image or picture file. The most common graphics used on the web are GIFs and JPGs (or JPEG. You can tell which is which by looking at the 3 letter extension after the file name.

GIF, or Graphic Interface Format, files are generally used for fonts as images (such as with the titling font above in Futura) or files with little detail and less than 256 colors. You can create animations and transparent files using GIF images.

JPG (JPEG), or Joint Photographic Experts Group, is considered a [lossy](#) file compression technique. It is one of the best file formats for the web as you can cut the file size by half or more without sacrificing too much quality. Generally, you would use JPG format for images with more detail and color. Keep in mind, jpg cannot save transparency (such as a clear background).

There are a number of great pages online that use a very limited amount of graphics and use color and fonts to achieve their desired effects. Before using graphics on your page, there are some things to consider. Graphics can slow your page load time significantly if used incorrectly. Although various sources report that the average surfer will wait about 8-10 seconds for a page to fully load, realistically the average surfer will only wait 3-4 seconds for a page to fully load unless they already trust your website or they require information that is only found on your page.

Putting yourself in your page visitor's shoes and asking yourself, how long would you wait for your page to finish loading before you leave, is a good idea.

Optimizing all of your graphics for the web can shave up to a couple seconds off your page loading time. Optimize simply means to get the file size as small as possible without sacrificing its quality. Make sure you optimize your images with photo editing software and not just change the width and height in the HTML code.

You do not want your page to be larger than 40k – 60k total with graphics included. Ideally, your page should be 10k or less and the more graphics you use, the bigger the page file size and the bigger the page file size, the longer it takes to load.

Some things to note about graphics:

- Use graphics sparingly – unless you are running a business that specializes in art or graphics, there is no need to have an unusually high number of images or pictures.
- You can decrease your file size by opening the file with image editing software and reducing the resolution to 72 DPI and saving it as a medium quality JPG. The file size will change, but the image size will stay the same (i.e. it will display as if it were the same size).
 - If you have Photoshop you can use the “save for web” feature which cuts most of the guess work for you. Simply select it from the File menu.
- Make sure the graphics you use are not only aesthetically pleasing, but also useful. Instead of having a graphic taking up space on your page and sucking up bandwidth, make it earn its keep. Ask yourself, does this image illustrate a product or service that is available on my page?
- Avoid graphics that blink, flash, or detract from the over all professionalism of your page. If you do decide to include banner advertising, try to keep it uniform and consistent with the content and flow of your page. Try to use as few flashy animated banners as possible.

=== **Sidebar** ===

Photoshop offers a free 30 day trial, but if Photoshop is a bit out of budget, you might want to try some other more cost effective software programs specifically designed to reduce graphic file sizes. <http://www.buniek.com/links/graphics.html>

You don't need to know how to create cool animations or other graphics; there are many sources online where you can find stock photos or clipart.

<http://www.buniek.com/links/stock.html>

<http://www.buniek.com/links/clipart.html>

Make sure you select stock art that is specific for your needs and royalty free.

=== **End sidebar** ===

1.4 Page flow and layout

When deciding on the page layout that your visitors will see, think of your page in three sections:

1) **The header** usually contains the company logo, slogan, and sometimes the very basic navigation such as "home", "contact us", and "site map" just in case a visitor enters on a page other than the homepage.

2) **The body** usually contains the navigation and the main content of the page.

3) **The footer** usually contains your copyright information and links to the more legal pages such as the site disclaimer, terms of service, or privacy policy if your site has one. Consider the footer as the fine print of your page.

To provide continuity, your header and footer should be the same, or at least very similar, throughout the other pages on your site.

Bear in mind that some basic design concepts such as the directional flow of your page can arouse different feelings in your visitors. A horizontal page flow suggests calmness, stability, and tranquility and a vertical flow suggests balance, formality, and alertness.

To achieve a horizontal flow, consider using horizontal lines and a more prominent horizontal navigation bar. Use longer horizontal lines and justify the text (stretches the text from left to right).

To achieve a vertical page flow, consider using columns to separate your content (like those used in a newspaper or magazine) or a more prominent vertical navigation bar.

A few things to keep in mind when deciding on the flow and layout:

- Keep it simple. Remember that less is more and most search engines reward pages that provide more useful content without a lot of needless frills.
- Continuity provides a more professional and organized touch to your page and that builds the trust of your visitors.

=== **Sidebar** ===

There are other design principles to be aware of when creating a web page, to learn about some of them, visit <http://buniek.com/articles/2004/06/user-friendly-website-hints-tips.html>.

If you've come this far and you're thinking to yourself that this requires too much thought and planning for you, there is always the use of a template.

<http://www.buniek.com/resources/templates.html>

Templates provide you with a basic design for your website and you simply customize it with your own content, so you will want to know at least basic HTML (XHTML) and CSS so you understand what you're doing.

=== **End sidebar** ===

Summary:

- Choose 2-3 colors to start and stick with it.
- Pick your fonts, but not too many. Try to stick with 2 or 3 nice fonts that compliment each other well. Get fancy with your titling/logo font and make sure it tells people what you or your company is about. Experiment with weight, size, and color.
 - Need more fonts? <http://www.buniek.com/links/fonts.html>
- Use your graphics sparingly. Don't forget to optimize your graphics for the web and make sure your graphics are doing double duty by being aesthetically pleasing as well as functional. Avoid graphics that serve no real purpose to your page.
 - Want to optimize your graphics? <http://www.buniek.com/links/graphics.html>
 - Want to find stock photos? <http://www.buniek.com/links/stock.html>
 - Want to find clipart? <http://www.buniek.com/links/clipart.html>
- Think of your page in three parts, header, body, and footer. The flow of your website should be simple and continuity helps create a professional image.
 - Want to use a template instead? <http://www.buniek.com/resources/templates.html>

Section 2

Basic HTML (XHTML)

2.1 What you need to get started

You really don't need too much to get started writing your own web page.

- A computer (Mac or Windows is fine)
- Basic computer knowledge
- A word processor such as Notepad or another plain text editor is recommended.

If you downloaded this ebook, figured out how to open it and have read this far, I will assume you have at least two out of three.

Open up a blank text document and let's get started. The lessons in this section build upon itself and by the end, you should have created a basic webpage – you can catch a sneak preview at <http://www.buniek.com/resources/css-xhtml/xhtml-example.html>.

=== Sidebar ===

To keep things more organized, you might want to create a folder for your images, your css files and your script files. If you need to find an image or graphic, you will already know that it is in the 'images' folder or if you need to update your CSS file (covered in section 3) you will know it is in your 'css' folder, etc.

=== End sidebar ===

2.2 What is HTML and XHTML

HTML is the acronym for **H**ypertext **M**arkup **L**anguage. The language or format used to create documents for the web (i.e. web pages).

XHTML is an acronym for **E**xtensible **H**ypertext **M**arkup **L**anguage. XHTML is a hybrid of HTML and XML and is a way of making your web pages/documents compatible with the new XML standard.

2.3 Which one should you use?

As the internet continues to evolve, it's best to do it better and do it well the first time around instead of having to play catch up later. With that said, it is highly recommended that you create a page that is Web Standards Compliant which will ultimately enable you to easily and efficiently separate your content from your design, leaving you with a faster loading, leaner, and more search engine friendly pages.

XHTML also has a number of economic and accessibility advantages over HTML and creating a Standards compliant page in XHTML is actually not that much more difficult than making a regular page, in some cases, it's actually easier. XHTML uses the same tags as HTML so that will not be your biggest learning curve. If you already have some experience or knowledge of HTML, a couple of your biggest learning curves in writing XHTML will be closing *all* your tags, nesting your tags properly, and using only lowercase for your tags and attributes.

2.4 Basic structure

When you create a webpage, there are three basic elements to take into consideration:

- 1) The document type (XHTML, HTML, XML, etc.)
- 2) The head
- 3) The body

A valid XHTML will page will basically look like this:

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

</html>

Below you will learn about each section, some of the rules they must follow as well as how to set up a basic XHTML page.

Some things to consider before you get started:

- All tags need to be lowercase. HTML is not case sensitive, XHTML is.
- All tags are enclosed by < > (less than, greater than signs),
- All attributes need to be quoted
- All tags need to be closed. In HTML, there are some tags that do not have an independent close – they are called empty tags, but they still need to be closed with " />" (space, forward slash, greater than sign) in XHTML.
- All tags must be properly nested.

Some of the things you could get away with in HTML, you can't get away with in XHTML.

HTML is not CaSe SeNsAtIvE, but XHTML is. It is a good idea to get in the habit of writing in lowercase.

 This is valid HTML, but this **is not** valid XHTML
 This **is** valid XHTML

All your tags must be enclosed by < > (less than, greater than signs) and you would use </ > (less than, forward slash, greater than) to end your tags

your "start head" tag would look like:

<head>

your "end head" tag would look like:

</head>

For **empty tags**, you would use " />" (space, forward slash, greater than sign):

Three tags that are empty tags are a line breaks, horizontal rules, and image tags:

 - This is valid HTML, this **is not** valid XHTML

 - this **is** valid XHTML

<hr> - this is valid HTML, but it **is not** valid XHTML

<hr /> - this **is** valid XHTML

 - this is valid HTML, but this **is not** valid XHTML

 - this **is** valid XHTML

Quote all attributes:

 while this is valid HTML, this **is not** valid XHTML, the width and height attributes need to be quoted and the tag needs to be closed

** this **is** valid XHTML, your width and height attributes are quoted and your tag is closed

All tags must be properly nested:

<i>This text will be bolded and italicized</i></i> - you can get away with this in HTML, but it **is not** valid XHTML

<i>This text will be bolded and italicized</i> - this **is** valid XHTML.

2.5 DOCTYPE Declaration

A DOCTYPE DECLARATION (DTD) is what tells your browser what language (XHTML, HTML, XML, etc.) you used to create the page and it also tells it how to interpret the elements in the page (i.e. how the page should display itself).

Examples

XHTML DTD:

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

This tells your browser that you are using XHTML 1.0 transitional for your webpage and to use the element rules associated with it.

HTML DTD:

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

This tells your browser that you are using HTML 4.01 transitional for your webpage and to use the element rules associated with it.

There are a number of other DTDs which you can use, for the complete list, visit <http://www.w3.org/QA/2002/04/valid-dtd-list.html>.

For the purpose of this ebook, we will be using the first DTD (XHTML 1.0 Transitional).

While you can still create an HTML document without using a DTD it is highly recommended for accessibility and functionality purposes. If you want to create valid XHTML documents, it must contain a DTD.

2.6 The Head

Now that you've set your DTD and have told the browser what rules to use in order to interpret your page, it's time to begin your coding.

To tell the browsers where your page starts, you want to write `<html>` just below your DTD – this tells the browser that you are beginning your actual page so it's okay to start interpreting the information.

Example:

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

=== Sidebar ===

If you are using software such as Dreamweaver or view the source code of some other websites, instead of `<html>` you may see:

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
```

`xmlns` is the namespace declaration and is beyond the scope of this ebook, but it's a good idea to include it and a great habit to get into when writing XHTML. If you want to learn more about namespaces, visit <http://www.w3.org/TR/REC-xml-names/>

`lang` and `xml:lang` is the language declaration for the page. It tells the browser that the content on the page is in English.

=== End Sidebar ===

Once you've told the browser that you are starting your actual document (the html start tag), you will want to move on to the head element.

Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TF/xhtml1/DTD/xhtml1-transitional.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en" >
<head>
```

In the head, you will put the information that you don't necessarily want the browsers to display to your visitors, but it does contain all the information you want the browsers and robots to use while it's at your site. Some examples of this include meta tags, script tags and style tags.

2.6.1 Meta Tags

You can use Meta tags to tell a robot how to index your page, how to describe your page and what keywords to use. You can also use Meta tags to tell a browser to refresh your website after a certain amount of time or not to save any files to the visitor's computer (cache). In this ebook however, I'll discuss the most frequently used Meta tags.

Character Encoding: `<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />`

This meta tag tells the browser what type of document your page is – such as html – and what character encoding rules to follow. The character encoding tells the browser how to convert bytes into characters (turning the coded information into readable content for your visitors). It is very important that the character encoding of the document be clearly defined. Ideally, this should be the first tag after you start your head.

For documents in English and most Western European languages, the widely supported iso-8859-1 is generally used.

For a list and descriptions of other character encoding, visit <http://www.buniek.com/links/charset.html>.

Title: `<title> </title>`

The title tag, although not a Meta tag, is inserted in the head of the document and tells the browser and robots what the title of the page is. It is important that if you are implementing search engine optimization that you optimize your title tag with the main keyword you use on the page. Many search engines put more weight on the keyword density of your title tag than your keyword and description meta tags.

Meta keyword: `<meta name="keyword" content="widgets" />`

This tells the search engines what words you want to be listed under when someone does a web search. Although many search engines do not put a lot of weight on the keyword tags anymore, it is still an important tag to have for search engine optimization. Only use a few highly relative keywords, some search engines penalize a page for using unrelated keywords – this is called "keyword spamming" or "keyword stuffing".

Keyword rule of thumb: if you don't use the keyword anywhere on your page, it really doesn't belong in your keyword meta tag.

Meta description: `<meta name="description" content="We sell the greatest selection of widgets on the internet today. Please visit the Widget store for all your widget needs." />`

This is the text a search engine will most likely use to describe your website in search engine results, so make sure it is eye catching, concise, and is optimized with your keywords. It is said that search engines put more weight on the keyword relevance in your description than in your keyword tag, so

making sure you have a good description of your website that is optimized with your keywords is important.

Meta robots: <meta name="robots" content="index, follow, all" />

The robots tag simply tells a robot what to do at your website.

Index: means spider the page and allow the website to be indexed by the search engine.

Noindex: The page will not be crawled and the page will not be indexed by the search engine, i.e. not to show up in search results

Follow: means follow all hyperlinks on the page

Nofollow: means not to follow any hyperlinks on the page

All: you can use this in conjunction with any of the above, separated by a comma, which tells the robot to do all of what you specify. For instance, "follow, all" tells the robot to follow all the links on the page.

The robots tag is best used if you do not have a robots.txt file or if you feel that some robots will not use the robots.txt file

=== **Sidebar** ===

A robots.txt file is a text file that tells a robot what to do and where to go on your website. To learn more about robots.txt files, you can visit <http://www.buniek.com/resources/robotstxt.html>.

=== **End sidebar** ===

Meta revisit: <meta name="revisit-after" content="15 days" />

If you update your page regularly with new content, it is a good idea to include a meta revisit tag. This tells the robot to come back and crawl your page again after a certain amount of days. Keep in mind that this does not guarantee the robot will come back, but chances are good if you continue to update your page with good content rich information, it will.

Although there are many other meta tags you can include, these are the basic meta tags you will need for a search engine friendly page.

Example of the head with title and meta tags:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TF/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>The Widget Homepage</title>
<meta name="keywords" content="widget" />
<meta name="description" content="We have the greatest selection of widgets on
the internet today. Please visit the Widget store for all your widget needs." />
<meta name="robots" content="index, follow, all" />
<meta name="revisit-after" content="15 days" />
```

Some of the other more common, but optional meta tags include:

Copyright: <meta name="copyright" content="2004 widgets plus" />

Gives the copyright information about your page

Language: <meta http-equiv="content-language" content="en" />

Defines the language used on your website. The above means the website is in English.

Rating: <meta name="rating" content="general" />

Tells the search engines whether your site is suited for a general, restricted, or adult audience.

Abstract: <meta name="abstract" content="the world of widgets online." />

An overview of your entire web page in one sentence.

Expires: <meta name="expires" content="Mon, Aug 23, 2004 24:00 GMT" />

Tells the search engine when the content of your site should expire

=== Sidebar ===

If you really want to see what other meta tags exist out there, many of which are for specific browsers or robots you can visit <http://www.buniek.com/resources/meta-list.html>.

=== End sidebar ===

2.6.2 Linking External Files

If you need your visitors' browser to use a different file such as javascript or CSS (Cascading Style Sheet) to display the page correctly, you can tell a browser to grab another file by placing a direct link or import link in the head of your document.

The format for linking javascript files and CSS files:

Script files: <script type="text/javascript" src="location/of/file.js"></script>

Style Sheet: <link rel="stylesheet" type="text/css" href="location/of/stylesheet.css" />

type = "this tells the browser what kind of file will be used"

href or src = "tells the browser where it can find the file to use"

There are other techniques for telling the browser what other information to use to display your page correctly, such as importing the file (rather than linking to it) or writing the code directly into the head of the document. Generally, linking to the external file will be better for search engine optimization and organization.

=== Sidebar ===

If you want to learn more about the other techniques for importing files or writing the code directly into the webpage, visit <http://www.buniek.com/resources/import-css.html>.

=== End sidebar ===

If you don't need any other files for your page to load properly, you don't have to worry about linking or importing and can omit this section and simply close the head after the meta tags have been completed. Since this ebook continues to discuss basic CSS, you will be linking to a style sheet (example.css) by the completion of your web page.

Once you've finished linking your external files, it's time to close your head and you should now have something that looks like the example below.

Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TF/xhtml1/DTD/xhtml1-transitional.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en" >
<head>
<title>The Widget Homepage</title>
<meta http-equiv="content-type" content="text/html; charset="iso-8859-1" />
<meta name="keywords" content="widget" />
<meta name="description" content="We have the greatest selection of widgets on the internet today.
Please visit the Widget store for all your widget needs." />
<meta name="robots" content="index, follow, all" />
```

```
<meta name="revisit-after" content="15 days" />
<link rel="stylesheet" type="text/css" href="css/example.css" />
</head>
```

2.7 The body

The body is what will show up on your visitors' computer screen. This is where you tell the browser what you want to show to your site visitors and how you want to show it. The body will always start after the head's closing tag. To start the body, you would use the start body tag <body>.

2.7.1 Some basic tags

There are a number of tags that can be used in the body section, but we will focus on the most common and important tags, headings, paragraphs, and line breaks.

Headings: <h#></h#>

Headings are generally used for section titles. Replace the # with a number 1-6. H1 defines the largest heading and h6 defines the smallest. HTML automatically adds a blank line before and after the heading.

Paragraph: <p></p>

The content between the start paragraph and end paragraph tag will be displayed as one chunk of text. HTML automatically adds a blank line before and after the paragraph.

Line Break:

The line break is used when you want to end a line without starting a new paragraph. The
 forces a line break wherever you place it.

Horizontal Rule: <hr />

This adds a horizontal line that can separate content on a page. You can use the horizontal rule to divide up sections on your website.

Unordered List:

This creates a bulleted list of items. You will need to define the list items using the list item tag which goes inside the unordered list tag. this is an unordered list item.

Ordered List:

This creates an ordered numbered list. You will need to define the list items using the list item tag which goes inside the ordered list tag. this is an ordered line item.

List Item:

Contained within the unordered or ordered list tags, this tells the browser to start a new bulleted or numbered line. You can have paragraphs, headings, links, line breaks and even other unordered or ordered lists within a list item.

Anchor: <a>

You can use an anchor to link to another document. You can also link to another section within the same document or specific parts of another document by applying the name attribute to section you want to link to.

Bold: or

This tells the browser to make the content inside the tags bolder or stronger.

Italicize: <i></i> or

This tells the browser to make the content inside the tag italicized or embellished.

Image:

This tells the browser to display an image file

Comment: <!-- comment goes here -->

Anything within the <!-- comment goes here --> will be ignored by the browser and will not be shown to your visitors. Comments are useful for making notes in your HTML for reference at a later date.

2.7.2 Some basic tag attributes

HTML (XHTML) element tags can have attributes. Attributes provide more information for the browser and can tell the browser to show the element a certain way or apply certain properties to it. An example of an attribute would be `<p class="center">` and attributes always come in name/value pairs such as `name="value"` and are always added to the start tag `< >`.

The core attributes in XHTML are `class`, `id`, `style`, and `title`. These attributes can be applied to any element with the exceptions of `base`, `param`, `head`, `meta`, `html`, `title`, `script`, and `style` elements.

Class: `class="value"`

If you define a special class with specific styling rules in a CSS file, you can tell the browser where to find the rules by applying a class to the element – discussed in section 3.

Id: `id="value"`

A unique id for the element which can also have specific styling rules in a CSS file.

Style: `style="value"`

An inline style definition – this means that you write the styling rules you want the element to use right in the value section.

Ex: `<p style="font-face: verdana, arial, geneva, sans-serif">`this paragraph will be displayed in verdana, arial, geneva, or the default sans-serif font`</p>`

Title: `title="value"`

This defines the text to use in the tool tips. Tool tips usually appear when you hover over a specific element such as a link or image.

There are, of course, other tag attributes which can be applied to specific tags or HTML elements.

Reference: `href="url"`

The href attribute tells the browser where to find the file you are linking to. Generally, you would use the href attribute in the link (`<link />`) and anchor (`<a>`) elements.

``links to another file``

`<link href="url" />` links an external file in the head of your document

You can also create a link to open your visitors' mail client with a new message addressed to you:

``you@youremail.com``

Source: `src="url"`

The src attribute tells the browser where to find the file to use in your page. Generally, you would use the src attribute in the script (`<script />`) and image (``) elements.

`<script src="url" />`

``

Target: `target="value"`

The values are `_blank`, `_parent`, `_top`, `_self` and it defines where the linked documents will be opened. The target attribute is usually applied to the anchor element (`<a>`).

Name: `name="value"`

The name attribute is used in the anchor element (`<a>`) when creating named anchors for linking to specific sections of a document. Generally used for creating a table of contents at the beginning of very long documents.

``this link will point to your named anchor``

`` this is where clicking on the above link will bring you``

Alternate Text: `alt="value"`

The alt attribute defines the alternate text to use for an image. The alt attribute is required for valid XHTML, it's a good idea to get into the habit of writing good alt text for your images and also optimizing it with your keywords.

```
<img alt="you define the text" />
```

```
=== Sidebar ===
```

There are more tags and attributes, but we've discussed the basic ones – if you want to learn more about the other possible tags and attributes, please visit <http://www.buniek.com/links/html.html>

```
=== End sidebar ===
```

2.7.3 Putting it together

In section 2.6 you put together the DTD and head of your document and started your body tag. Now that you have a basic idea of some of the tags you can use in the body and what they do, you can begin to create the page that your visitors will see.

If you remember from the First section regarding your page layout – you want to look at it in three sections, the header, content, and footer. If you've used an image file for your header or logo/branding, then your very first paragraph will contain an image.

```
...
</head>
<body>
<p>

</p>
```

This paragraph will only contain the image file you want to use for your logo/branding. If, however, you want to use titling text and not an image, you can use the heading tag.

```
...
</head>
<body>
<h1>The Ultimate Widget Resource</h1>
```

Once you have your header situated, you can move on to the body copy. The body copy usually has the navigation and the main content of your website.

If you want a horizontal navigation menu, you can line up the text in one paragraph.

```
...
</head>
<body>
<p>

</p>
<p>
<a href="url">Link #1</a> | <a href="url">Link #2</a> | <a
href="mailto:you@youreemail.com">Contact you@youreemail.com</a>
</p>
```

Now you have a heading and you have some navigation, next you may want to create a sub heading for your page content using the second level headers.

```
...
<h2>Everything you ever wanted to know about widgets</h2>
```

If you have a lot of content on the page, you may want to consider setting up a table of contents near the top for the convenience of your site visitors.

```
...
<h2>Everything you ever wanted to know about widgets</h2>
<ul>
<li><a href="#widget1">This is for widget section 1</a></li>
<li><a href="#widget2">This is for widget section 2</a></li>
<li><a href="#widget3">This is for widget section 3</a></li>
</ul>
```

Now that you've linked your widget sections, it's time for you to create the sections for your table of content links to go.

```
...
<ul>
<li><a href="#widget1">This links to widget section 1</a></li>
<li><a href="#widget2">This links to widget section 2</a></li>
<li><a href="#widget3">This links to widget section 3</a></li>
</ul>

<h3><a name="widget1"></a>This is widget section 1</h3>
<h3><a name="widget2"></a>This is widget section 2</h3>
<h3><a name="widget3"></a>This is widget section 3</h3>
```

Now that you've created your table of contents and corresponding named anchors (where the visitor will be taken once they click on the table of content link), it is time for you to go back and add your content to each section.

```
...
<ul>
<li><a href="#widget1">This links to widget section 1</a></li>
<li><a href="#widget2">This links to widget section 2</a></li>
<li><a href="#widget3">This links to widget section 3</a></li>
</ul>

<h3><a name="widget1"></a>This is widget section 1</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis tempus nunc vel odio. Nullam sodales, dui non vehicula volutpat, leo nibh condimentum ipsum, at aliquet lectus ipsum fermentum magna. Aenean tincidunt odio eu tortor. Integer molestie, ligula non aliquam condimentum, elit nunc aliquam sapien, ac hendrerit elit lacus ut sem. Ut cursus. Vivamus congue congue diam. Maecenas commodo sagittis lorem. Aliquam erat volutpat. Fusce vitae mi in ipsum tempus mollis. Praesent mi turpis, rutrum quis, accumsan scelerisque, hendrerit at, ante. Phasellus sollicitudin lacus quis est. Aliquam erat volutpat. Quisque vulputate sem vel augue. Nunc arcu. Maecenas adipiscing.</p>

<h3><a name="widget2"></a>This is widget section 2</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis tempus nunc vel odio. Nullam sodales, dui non vehicula volutpat, leo nibh condimentum ipsum, at aliquet lectus ipsum fermentum magna. Aenean tincidunt odio eu tortor. Integer molestie, ligula non aliquam condimentum, elit nunc aliquam sapien, ac hendrerit elit lacus ut sem. Ut cursus. Vivamus congue congue diam. Maecenas commodo sagittis lorem. Aliquam erat volutpat. Fusce vitae mi in ipsum tempus mollis. Praesent mi turpis, rutrum quis, accumsan scelerisque, hendrerit at, ante. Phasellus sollicitudin lacus quis est.</p>

<h3><a name="widget3"></a>This is widget section 3</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis tempus nunc vel odio. Nullam sodales, dui non vehicula volutpat, leo nibh condimentum ipsum, at aliquet lectus ipsum fermentum magna. Turpis, rutrum quis, accumsan scelerisque, hendrerit at, ante. Phasellus sollicitudin lacus quis est. Aliquam erat volutpat. Quisque vulputate sem vel augue. Nunc arcu. Maecenas adipiscing.</p>
```

Once you feel comfortable with your body content, then you can finish up by adding your footer information. If you want some way to separate the footer from the actual body content, you may want to use a horizontal rule.

```
...
<h3><a name="widget3"></a>This is widget section 3</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis tempus nunc vel odio. Nullam
sodales, dui non vehicula volutpat, leo nibh condimentum ipsum, at aliquet lectus ipsum fermentum
magna. Turpis, rutrum quis, accumsan scelerisque, hendrerit at, ante. Phasellus sollicitudin lacus quis
est. Aliquam erat volutpat. Quisque vulputate sem vel augue. Nunc arcu. Maecenas adipiscing.</p>

<hr />
<p>This is the Widget page footer and it contains the copyright information and the other fine
print</p>
```

Once you've completed your footer, it means you're almost done with your page, but not quite. Remember in XHTML you need to close all tags and you still have two tags open – your html and your body tag. To finish up your document, you want to close them.

```
...
<hr />
<p>This is the Widget page footer and it contains the copyright information and the other fine
print</p>

</body>
</html>
```

Now it's time to save your document as a web page file so that it can be viewed in a browser such as Internet Explorer or Netscape Navigator.

=== Sidebar ===

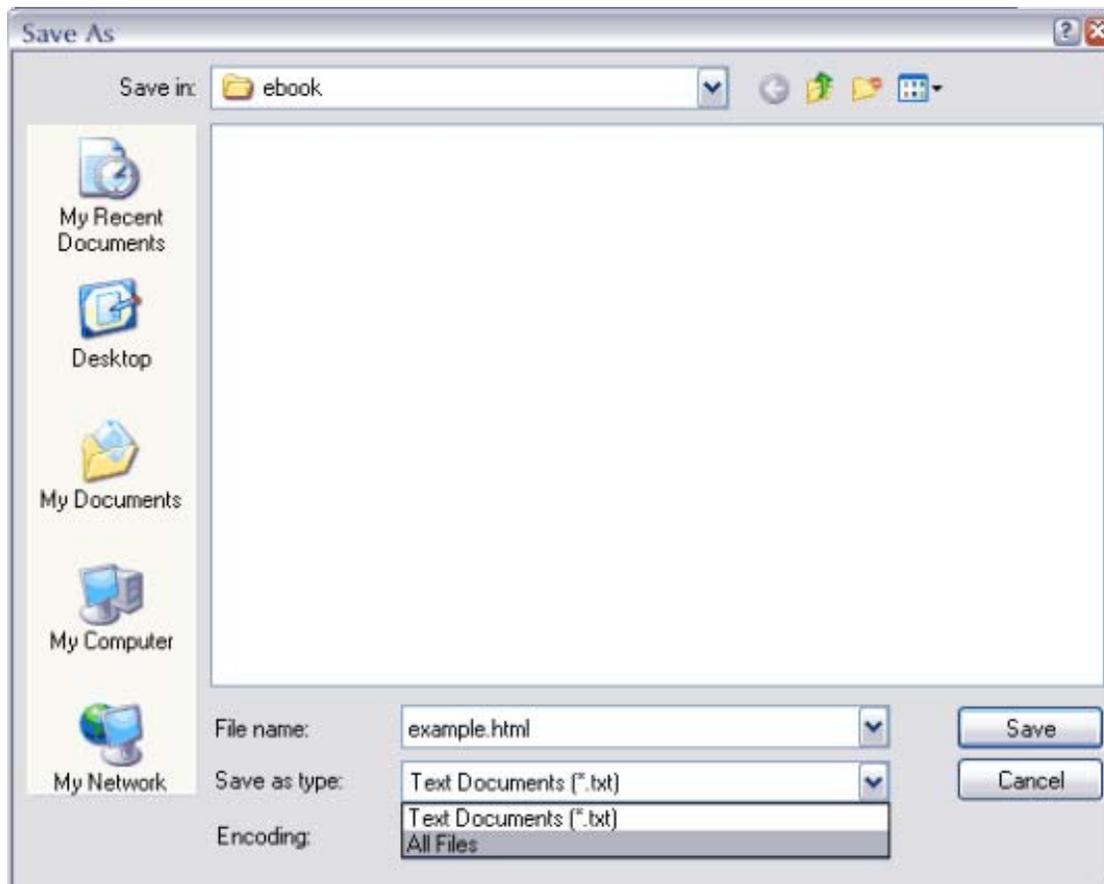
If writing HTML by hand seems too tedious for you, there are many cost effective HTML/XHTML editors. TopStyle which has a free lite version but it does not have as many features as Macromedia's Dreamweaver.

To try TopStyle visit <http://www.buniek.com/links/topstyle.html>

=== End Sidebar ===

2.8 Saving your XHTML document

If you are using a text editor such as notepad, select 'File' -> 'Save As' and a window will pop asking you to specify a name and location to save your file. Select the drop down menu next to 'Save as type' and make sure you select 'All Files'. In file name, type in the name you want to use for your document plus the four letter .html (or three letter .htm), then click 'Save'.



You have just finished up an XHTML compliant web page, but you may want to validate it just to make sure you didn't miss anything – people make mistakes, that's why pencils have erasers. Validating your page will allow you to catch small errors you may have overlooked in the code. Validating your page is also free and easy at <http://validator.w3.org/>.

If your code does not validate the first time, don't panic – the validator will tell you exactly where the error occurred and what the error is so you can easily track it down and correct it.

Section 3

Basic CSS

3.1 Getting Started

I bet you're wondering "what can CSS do anyway?" and to answer – it can actually do quite a bit. CSS stands for Cascading Style Sheet and it does just that – it styles your document for you. CSS is very good for separating the design from content. It can reduce the amount of time spent updating your page as well as reduce the overall page size.

It can tell a browser what font to use for each paragraph or the whole page, what you want your background to look like on just one section or the whole page, and it can even tell the browser what information you want your visitors to see and what you don't want them to see on your page – such as creating alternate hidden text for people browsing with text readers or who have their images disabled. The possibilities are almost endless with CSS and you have a lot more flexibility to style your pages.

Yes, you can do some of these things with plain old fashioned HTML, but you do not get the flexibility that CSS affords you. For instance, if you create a 5 paragraph HTML document using font tags, you will have to tell the browser what font to use for each paragraph – i.e. you have to write your font tag 5 times. If you create a CSS file that tells the browser what font you want to use for all your paragraphs, you only need to write this information once. This may not seem like a big deal now, but when it comes time for updates or changes to the design of your page or site – such as changing the color of your text – you will need to change it in all 5 font tags in the HTML document, when you only need to change one value in the CSS file. 5 paragraphs may not seem like much, but when you apply this concept to a complete website or a page with 50 paragraphs, you can now see how CSS can save your time for more important things.

3.2 Code Layout – how to write correct CSS

Just like writing HTML, writing CSS has the same requirements to get started, a computer, basic knowledge of the computer, and a word processor such as notepad or another plain text editor.

Open up a new document – do not use the same notepad document you used for your XHTML.

CSS is CaSe SeNsAtIvE – there are only a few attributes that are not case sensitive, but the majority is. The syntax used for valid CSS is

```
selector.class {property: value; }
```

The selector is the HTML element/tag you wish to style or change such as the body or the paragraph.

Think of the class as being an optional item. If you use an element multiple times in your page, but you want each one to be different in some way, then class will definitely come in handy. Class names are also defined by you, but be wise when choosing your class name so you can easily tell what it does are simply by looking at the name.

The property is the attribute you wish to change such as the background or the font.

The value is what you want the attribute to do such as appear black on the visitors' screen. Each property can have a value and the value varies depending on what the property is. If the value is more than one word that needs to be read or interpreted as one item, you would put quotes around it.

If you want to make a comment in the code that you don't want the browser to use – maybe something to help you remember why you used a special color for your links – you would write `/* your comment here */` - the browser ignores everything within the `/* */` .

Example:

The syntax:

element/tag you used in the XHTML { attribute of the element: what you want the attribute of the element to do; }

For instance, I want all my paragraph tags (<p>paragraph</p>) to have the sans-serif font, then you would write:

```
p {font-face: sans-serif; }
```

If you want an element to do different things, use the class selector. If you decide you want your paragraphs to be justified (stretched from left to right), but you want some to be centered, you would write:

```
P { text-align: justify; } - your paragraph text will automatically be justified
```

```
p.center { text-align: center; } – any paragraph with a class called center will automatically be centered.
```

Once you've done this, you would then go back to your XHTML document and add a class attribute for any paragraph you want to be centered

Example (in XHTML document):

```
<p>this paragraph is justified</p> - the text in this paragraph will be justified
```

```
<p class="center">this paragraph is centered</p> - the text in this paragraph will be centered
```

If you wanted to use the Times New Roman font, but it is more than one word, you would need to quote it to tell the browsers to interpret it as one item.

```
p {font-face: "times new roman", times, serif; }
```

Commenting

```
/* this is a comment and will not be ignored by the browser */
```

Pretty simple, right? Once you're comfortable with the syntax, let's move on to some of the elements and attributes we'll be setting up in a basic CSS file.

3.2.1 Some basic selectors and their properties

Some selectors include:

Body: the body selector applies to all the content inside the body tag in your XHTML document (<body></body>). To start the body selector, write "body" without the quotes.

Paragraph: the paragraph selector applies to all the content inside the paragraph tag in your XHTML document (<p></p>). To start the paragraph selector, write "p" without the quotes.

Hyperlinks: the hyperlink selector applies to the hyperlink tags in your XHTML document (<a>). To start the hyperlink selector write "a" without the quotes.

Some properties (attributes of the elements that you can change) include:

Background: the background property is used to define how an element will appear. You can set a color, an image, make the image stationary, or tell the image where to begin on the page. Primarily, you will be focusing on changing the background's color. To do this, you can write "background" or "background-color" followed by the value.

Value for the background color is set with a # (pound) sign followed by the hexadecimal value.

Margin: tells the browser how close to the edge to display the content. You can use negative values to overlap information or run content right off the page.

The values for margins are usually set using px (pixels) or % (percentage).

Font-family: the font face property is used to define what typeface or font is used in a particular selector group.

The value can be defined by using a list of fonts separated with commas in order of preference. For instance: Verdana, Arial, Geneva, Sans-serif – the browser will try to load Verdana first, then Arial next if Verdana isn't available, then Geneva and then the default sans-serif font.

Font-size: the font size property determines how large or small the font in a particular selector group is displayed on the screen.

The value is calculated in px (pixels), [em](#), pt (points), or by some common English signifiers such as x-small, small, medium, large, and x-large.

Typically, [em](#) is the preferred measure to use as it allows the font to be easily resized from the visitors' browser menu options. It's best to make your page as accessible as possible in the beginning instead of trying to fix it later.

Although this value varies, typically 1em is equal to 12pt font.

Text-align: This tells the text where to line up on the page such as the left side, the center, the right side or stretched from left to right.

The values for this property are left, center, right, and justify. If you do not use the text-align property, the browser will use the default alignment which is usually left.

Color: the color property determines what color to display the text in a particular selector group on the screen.

The value for color is defined with a # (pound sign) followed by a hexadecimal value or you can use some common English signifiers such as white, black, red, blue, yellow, green, teal, purple, and orange. However, hexadecimal is highly recommended because some browsers don't fully support English signifiers.

=== Sidebar ===

Of course there are many other selectors and properties, but these are the ones focused on throughout this ebook. If you want to learn more about CSS and the many ways you can enhance your web pages using it – visit <http://www.buniek.com/links/css.html>

=== End sidebar ===

3.2.2 Putting it all together

In this section, we will be utilizing the color and font information we collected in the first section. You will need the notes you took on the color scheme and the font selection to create your style sheet.

Color Scheme:

Color Description	Hex Code	Notes:
Dark Olive Green	#5B5915	Body Text and content
Orange Red	#E06720	Links and other highlights Used sparingly throughout page
Creamy Almond	#FEF3BD	Page background

Fonts:

Section	Font	Size
Titling/Logo	Futura	60pt
Headlines	Arial, Verdana, Geneva, sans-serif	Main headings: 1.4em Secondary headings: 1.2em Third level headings: 1.0em
Body and Content	Verdana, Arial, Geneva, sans-serif	0.8em
Links and special text	Verdana, Arial, Geneva, sans-serif	0.8em

==== Sidebar ====

Although you have four font choices listed in your body and links section, consider it one – your main choice is verdana, the remaining three are fall back options is verdana is not available.

==== End sidebar ====

It's generally a good idea to start with the body as it encompasses the whole webpage. Decide what attributes you want to set up for your entire page, such as the background and the margins and what you want each one to do. Remember, anything you set in the body tag will be used throughout the entire page that your site visitors see.

```
body {
background: #FEF3BD; /* this is the background color of the whole page */
margin-top: 10px; /* how far away from the top of the page you want the to content to be */
margin-right: 150px; /* how far away from the right of the entire page you want the content to be */
margin-bottom: 10px; /* how far away from the bottom of the entire page you want the content */
margin-left: 150px; /* how far away from the left of the entire page you want the content */
}
```

Using the notes taken in section 1, you can set your paragraphs to have a specific font and color. If two elements are going to have exactly the same properties and values, you can group them together by separating them by a comma.

```
p, ul {
font-family: verdana, arial, geneva, sans-serif; /* the font for <p></p> and <ul></ul> */
font-size: 0.8em; /* size of the text in the <p></p> and <ul></ul> */
color: #5B5915 ; /* color of the text in the <p></p> and <ul></ul> */
text-align: justify; /* where the text lines up in the <p></p> and <ul></ul> */
}
```

If you decide that you want your footer paragraph to be centered – create a class.

```
p.center {
text-align: center; /* where the text lines up (centered) in the <p class="center"></p> */
}
```

So far so good, now you can move on to creating the special colored font (orange-red) for your links

```
a {
color: #E06720 ; /* this is the text color for all the anchors or links <a></a> */
}
```

If your links are included in a paragraph, you will not need to include the font-face or font-size attributes because they will be carried over from the paragraph selector.

Now you can define how you want your page headings to look

```
h1 {
font-family: arial, verdana, geneva, sans-serif; /* font used for main headings <h1></h1> */
font-size: 1.4em; /* text size for main headings <h1></h1> */
color: #5B5915; /* this sets the text color for the main headings <h1></h1> */
}

h2 {
font-family: arial, verdana, geneva, sans-serif; /* font used for secondary headings <h2></h2> */
font-size: 1.2em; /* text size for main headings <h2></h2> */
color: #5B5915; /* this sets the text color for secondary headings <h2></h2> */
}

h3 {
font-family: arial, verdana, geneva, sans-serif; /* font used for third headings <h3></h3> */
font-size: 1.0em; /* text size for third headings <h3></h3> */
color: #5B5915; /* this sets the text color for third heading <h3></h3> */
}
```

=== Sidebar ===

Everything within the /* */ will be ignored by the browser.

=== End sidebar ===

When you're finished, you should have a document that looks similar to this:

```
body {background-color: #FEF3BD;
margin-top: 10px;
margin-right: 150px;
margin-bottom: 10px;
margin-left: 150px;}

p, ul {font-family: verdana, arial, geneva, sans-serif;
font-size: 0.8em;
color: #5B5915;
text-align: justify;}

/* This is for centering certain paragraphs */
p.center {text-align: center;}

a {color: #E06720 ;}

h1 {font-family: arial, verdana, geneva, sans-serif;
font-size: 1.4em;
color: #5B5915;}

h2 {font-family: arial, verdana, geneva, sans-serif;
font-size: 1.2em;
color: #5B5915;}

h3 {font-family: arial, verdana, geneva, sans-serif;
font-size: 1.0em;
color: #5B5915;}
```

Special note: each property/value does not need a new line, but it helps with organization.

=== Sidebar ===

If writing CSS by hand seems too tedious for you, there are many cost effective CSS editors. TopStyle which has a free lite version allows you to edit your CSS.
<http://www.buniek.com/links/topstyle.html>

Western Civilization has what I call the CSS cheat sheet. Their Style Master software points out any mistakes you make in red and offers helpful hints while you code – they offer a free 30 trial version.
<http://www.buniek.com/links/stylemaster.html>

=== End Sidebar ===

When you're done with your code, don't forget to go back into your XHTML document and add the class="center" attribute to the paragraphs you want to be centered (on the example page, the navigation paragraph and footer paragraphs are centered).

Now, you're ready to save your CSS file. Don't be afraid to play around with the property values (colors, sizes, alignments, etc).

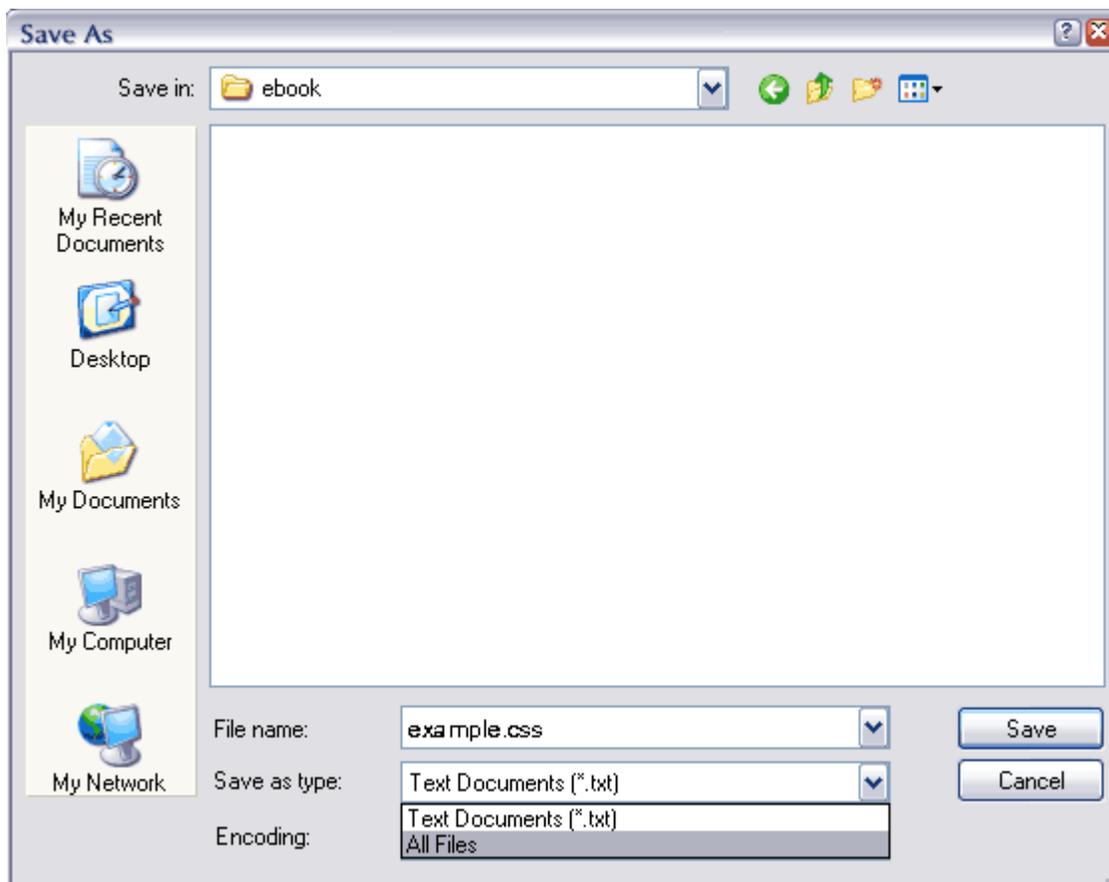
3.3 Saving your CSS document

If you are using a text editor such as notepad, select 'File' -> 'Save As' and a window will pop asking you to specify a name and location to save your file. Select the drop down menu next to 'Save as type' and make sure you select 'All Files'. In file name, type in the name you want to use for your file plus the three letter .css file extension, then click 'Save'.

That's it, you've now finished and saved your style sheet, but don't forget to validate. When you validate your code, you may catch some small errors you may have missed in the code. Validation is free, quick and easy at <http://jigsaw.w3.org/css-validator/>.

If your code does not validate the first time, don't panic – the validator will tell you exactly where the error occurred and what the error is so you can easily track it down and correct it.

example.css – it is very important to include the 3 letter file extension otherwise your computer (and your browser) will not recognize what type of file it is.



Since you've already linked to this file in your XHTML document, just make sure you save this file in the location and using the same name you specified in the head of your XHTML document.