

# ACADEMIC TECHNOLOGY SUPPORT



## **Desire2Learn:**

*Building HTML Content*

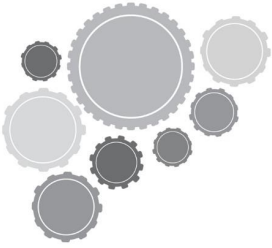


INFORMATION TECHNOLOGY  
SERVICES

EAST TENNESSEE STATE UNIVERSITY

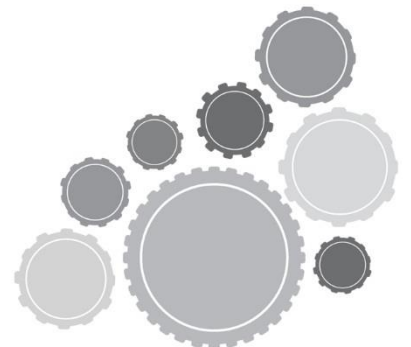
[ats@etsu.edu](mailto:ats@etsu.edu) | 439-8611

[www.etsu.edu/ats](http://www.etsu.edu/ats)



## Table of Contents:

Managing Expectations.....	1
Why Create HTML Content? .....	1
D2L Prep Work .....	2
What is HTML? .....	7
Exercise: Matching Content Types .....	9
Basic HTML Syntax .....	10
Quiz: HTML Basics .....	14
Conclusion.....	15
What about the WYSIWYG Editor? .....	16
Exercise & Quiz Answers .....	17
Additional Resources .....	17



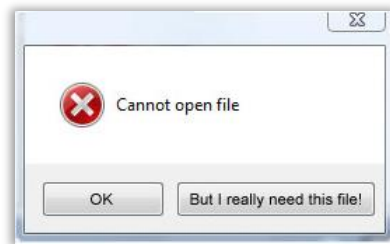
# Managing Expectations

I want to let you know what to expect in this workshop/guide before we get into deep water. The focus here will be on how to create HTML content in D2L. Since we will need to use a variety of tools to complete that task, this workshop/guide is technical in nature. We're going to spend a lot of hands-on time learning to understand and work with D2L, HTML, and file management. By the end, you should have the information and resources necessary to get started creating HTML content in your own D2L courses. This will be the beginning of a learning process, so do not expect to be an expert by the time you are finished here.

## Why Create HTML Content?

Before we get started with the “how” I do want to address the “why,” because if you aren't sure why you would want to do something in the first place then you won't be motivated to do it.

The primary reason for placing your course content in the HTML format is that it is *accessible*. It is easy for students—any student—to open. You see, with other popular formats like Microsoft Word or PowerPoint, even PDF, users must have an appropriate plug-in or software application in order to open and view the content. You may respond by saying, “Well most people have a PDF reader,” and based on my experience I would say that you are probably correct. However, compatibility across devices is becoming more and more complicated as new mobile devices are being introduced into the marketplace, and a large percentage of people are using these devices as their primary means of internet access. On many of these mobile devices, you can find ways to open a PDF file (with a little bit of effort), but all of them will open an HTML file with no questions asked. This is because all of these devices already have a web browsing application installed. You have to use a web browser to access D2L in the first place, so if you are on a D2L site then you already have the means to open an HTML file. In fact, D2L is, in itself, made up of HTML! This means that anyone that goes to your D2L course can open HTML content. This is a big advantage, and definitely a best practice. As an instructor, you ought to want to create an accessible, user-friendly experience for your course, because a friendly technical environment will foster a better learning environment.



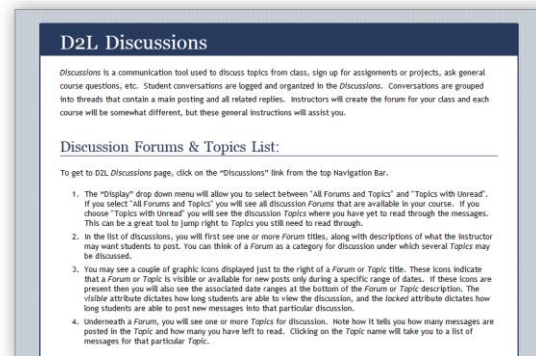
# D2L Prep Work

Though this guide does introduce you to the basics of HTML, it does not demonstrate or require you to go through the process of creating HTML files from beginning to end. Instead, we recommend that you utilize a pre-built HTML template (which will be supplied to you). However, to work with a template we need to do some prep work inside D2L before we start working with the HTML language.

Desire2Learn has developed their own set of HTML templates, which are free to download and use in your D2L courses. These templates consist of pre-built HTML pages which you can customize with your own content. It certainly is not required that you use one of these templates in your own course, but you will likely find that they make your content more readable than plain HTML. Here's a quick comparison of the same HTML content—one written in plain HTML and the other using one of the D2L templates:



Plain HTML



D2L Template ("Ordinary Blue")

These templates look better because they implement an additional language called *Cascading Style Sheets (CSS)* that modifies the plain HTML. CSS can be utilized to customize the look and formatting of HTML elements. This guide will not cover CSS (you don't need to know how to use it to implement one of these templates), but if you are interested in learning more about it then I encourage you to check out websites like [W3Schools](http://www.w3schools.com) to get started.

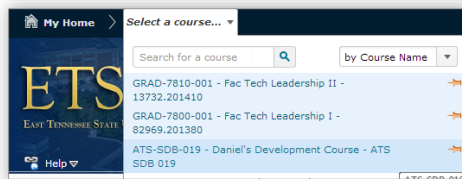
This guide is going to show you how to use these D2L templates, so go ahead and pick one that you like from the D2L HTML Templates page of the ATS site ([http://www.etsu.edu/academicaffairs/elearning/ats/training/exercisefiles/d2l\\_templates.aspx](http://www.etsu.edu/academicaffairs/elearning/ats/training/exercisefiles/d2l_templates.aspx)) and download it. Save the .zip file in a place that you will be able to find it (like your desktop), because you will need to use it to follow along with the guide.

Once you have a template package downloaded you need to put it in D2L. Uploading your template to your D2L site will require that you follow a specific set of steps, so get your mouse-clicking finger ready and follow along with the steps below:

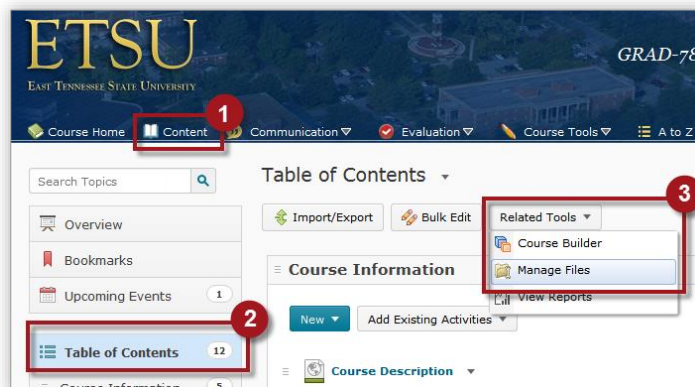
1. Go to ETSU's D2L login page (<http://elearn.etsu.edu/>) and sign in with your ETSU username and password. Once you have logged in, you will be taken to the D2L *My Home* page.



2. From the D2L *My Home* page, navigate to a course where you are the instructor. (You may have to click on the "Instructor" tab inside the "My ETSU Courses" box)

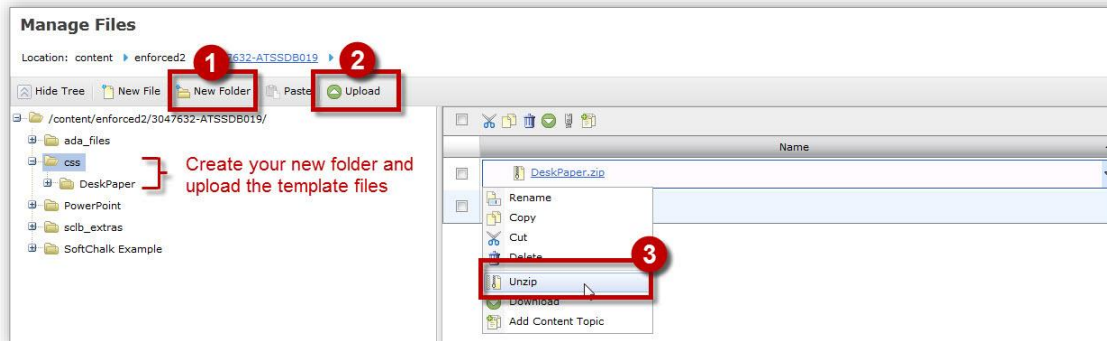


3. Navigate to the *Content* area of your course (1), then make sure that you are viewing the *Table of Content* (2), and then click on the *Related Tools* dropdown menu and select *Manage Files* (3)

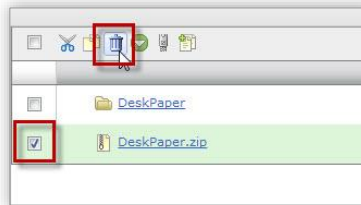


#### 4. Create a new folder, upload the template package, and then unzip it

Use the *Manage Files* tool to create a new folder called 'css' (1), then click on that 'css' folder which you just created and upload (2) and unzip (3) the template file. (See image below.)



Once you have unzipped the folder, you will have both a zipped and an unzipped version of the folder on your site. You can delete the .zip file. (Check the box to the left of the .zip file, and then click on the trashcan icon.)



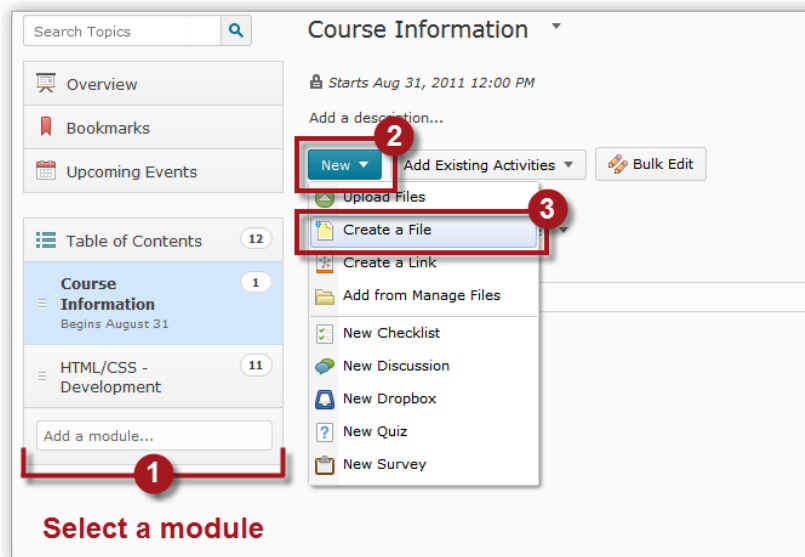
Now that all template files are in place, you can begin to build new topics that utilize template information.

#### 1. Go to the *Manage Content* area of your D2L course.



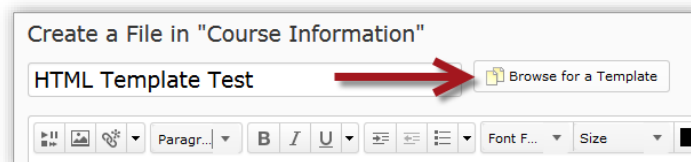
#### 2. Create a new topic.

Select a module from within your table of contents (1), then click on the blue *New* button (2), and select the *Create a file* option from the dropdown menu (3). This will create a new HTML file within your course.

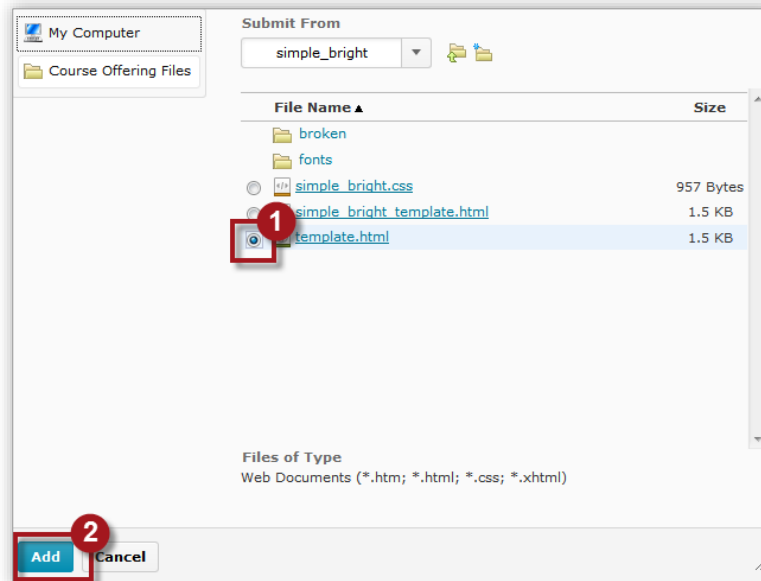


### 3. Load the template

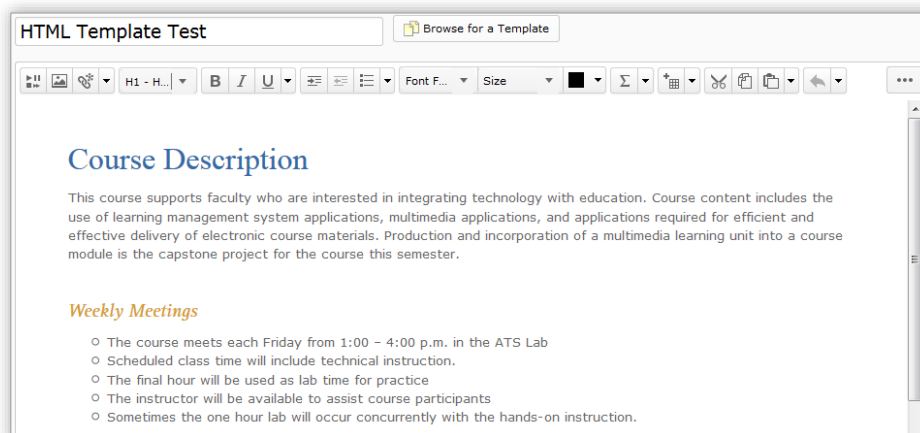
Give your new topic file a name, and then click on the *Browse for a Template* button just to the right of the name field. (See image below). This will open up a “popup” window.



Navigate to the ‘css’ folder you created earlier and then click on the template folder to see its contents. You should see either a ‘template.html’ or ‘blank\_template.html’ file. Click on the radio button just to the left of the file name to select the template file (1), and then click on the blue *Add* button (2). (See image below.)



You should now see a preview of your loaded template file in the HTML editor.



#### 4. Choose a destination for your new topic file.

This step is sometimes overlooked by instructors, but it can be a helpful part of keeping your course organized. By default, new files are placed in your course's root folder. After doing a lot of development in your course, you could end up with a mess of files of various sorts in your root folder, and these could become difficult to sort through. I recommend clicking on the *Change Path* button and placing your new HTML file within a folder that corresponds to the name of the parent module. You can create a new folder from this option if you desire.






## 5. Populate the template file with your content

# What is HTML?



HTML stands for Hypertext Markup Language. Those words alone may not provide much meaning, so it may be easier for you to think of HTML as a *means to describe web page content*. Conceptually, a markup language is very simple; it is a system for annotating documents in order to define the various components within those documents. A markup language is not the same as a programming language, which actually works *with* data to make decisions and perform tasks.

In this workshop, we are going to break down a fairly straightforward document into individual elements based on each element’s content type. This may sound more complicated than it really is. To demonstrate, let’s practice with a little exercise (on the next page).



### The Etymology of “Hypertext” (For word nerds)

The term *Hypertext* was coined by Theodor Holm Nelson, and was intended to connote “the extension or generality of text.” He coined the word to signify the non-linear nature and potential of a “web” of interconnected texts that could be contained in and retrieved from computational devices. It has become a bit of a misnomer as the medium has evolved, because *hypertext* now includes other media, like images and video, and not just text; therefore the word “hypermedia” (also coined by Nelson) would now be more appropriate. However, the original word seems to have found a permanent place in the HTML acronym, and it may never officially change.

**Do the exercise on the next page and then return to this text.**

If I were a gambling man then I’d bet \$1,000 that you aced that exercise. (Please observe that I am not actually a gambling man.) You may not be accustomed to using some of the web terms associated with these content types—for instance, you might say “bulleted list” instead of “unordered list”—but my guess is that you were able to figure it out with relative ease.

**Support Contact:** Daniel Gibson, *Technology Development Coordinator* ([gibsondv@etsu.edu](mailto:gibsondv@etsu.edu); 439-8244)

Working with HTML, conceptually, isn't much different than this exercise. However, instead of assigning number values to every content type, we are going to have to use HTML syntax—its lingo, or way of talking. Let's take a look at the HTML syntax now.

# Exercise:

## Matching Content Types

Match each content type (left) with an area of the document (right) by assigning each content type's corresponding number to one of the boxes on the right. Each option is used only once.

### Content Types:

1. Unordered List
2. Paragraph
3. Heading
4. Image
5. Horizontal Rule
6. Hyperlink

### Equipment Checkout

*Providing short-term equipment and device loans for ETSU faculty and staff*



Academic Technology offers a wide range of equipment ready for faculty and staff to check out. Need a video camera or audio recorder? Need a portable projector or just want to see how an iPod works? Check the list below for equipment we have available, and feel free to call ahead and reserve anything you know you might need for a certain event. Some items are limited in quantity, so be sure to call early.

- Cameras
- Laptops
- Microphones & Recorders
- PC Peripherals
- Projectors
- Response Systems
- Tablets, eReaders, & Mobile Devices

[Read more >>](#)

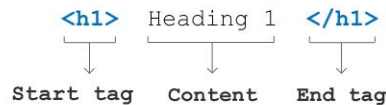
# Basic HTML Syntax

## Elements

HTML markup is made up entirely of individual, standardized *elements*, which describe both the document and the document's content. In order to define one of these elements you have to use a kind of coded marker, most commonly referred to as a "tag." (See the heading example below.)

## Elements that Contain Content

Here is a standard, top-level heading element:



This example demonstrates the basic syntax for any element which contains content. The content is surrounded by two tags, and each tag is composed of a standard element name (in this case "h1") which is enclosed in angle brackets. Both tags are necessary to define this content. The first tag indicates the beginning of the content, and the second is used to indicate the end of the content. The only required difference between an opening and closing tag is that the closing tag must contain a forward slash before the name of the element. Anything that is typed between these two tags will be considered a part of the heading.



### Element Names

Most elements are named with a kind of abbreviation or shorthand. You will have to memorize their names or consult a [reference](#) in order to identify and use them.

The heading above gives you a good idea of what most elements will look like in HTML. You could use this syntax to define several other forms of content by simply changing the element name. For example, here is the paragraph from our first exercise:

`<p>`Academic Technology offers a wide range of equipment ready for faculty and staff to check out. Need a video camera or audio recorder? Need a portable projector or just want to see how an iPod works? Check the list below for equipment we have available, and feel free to call ahead and reserve anything you know you might need for a certain event. Some items are limited in quantity, so be sure to call early.`</p>`

The only difference is that "p" has replaced "h1", and the paragraph content is now between the two tags.

Simple, no?

## Elements that Do Not Contain Content (*Empty elements*)

Not all elements contain content; some make declarations. These are called *empty elements*. One of the most common elements of this type is the *line break* element:

```
<br />
```

A line break causes the page content to jump down to the next line, but doesn't contain any content in itself. You will notice that there is a forward slash just before the closing angle bracket; this is used in place of a separate closing tag, because again, this element doesn't contain any content, and therefore doesn't need a separate closing tag to signal when it ends. An empty element is merely a one-off statement which declares that something should exist or happen at that particular point in the document.

Another example of an empty element is the *image* tag, shown below:

```
<img />
```

You may say, "An image can be a part of a page's content. How is it that an image tag is not considered to contain any content?" Good question. Images aren't truly defined from within an HTML document the way text is; they exist as a separate file which can be called into an HTML page. Since this is the case, you don't have to create the image content inside the HTML document, but merely state that it exists. However, you do have to tell the browser where to find the image you want to use, and for that we are going to need to use an element *attribute*.

## Element Attributes

Using plain elements, like in the examples above, will get you started, but you will quickly want to spice things up and modify some of the basic properties of your HTML content. The most basic way to modify an element is to use an element *attribute*. Element attributes are pre-defined properties, specific to each individual element, which can be given unique values in order to change the way an element behaves or appears. Here is the basic syntax for defining an attribute within a content containing element and an empty element, respectively.

```
<element attribute="value">Content</element>
```

```
<element attribute="value" />
```

Pay careful attention to the syntax of those two previous statements. First, notice that only the first tag contains the attribute information. Also, the spacing of is important and should be emulated. There needs to be a space between the name of the element and the name of the attribute, then an equal sign, then the value inside quotations. If you don't use that exact syntax then things are liable to break.

Let's look at a more practical example. Earlier I mentioned that we would need to use an attribute to define an image. Here is an image element with the image's location defined by the *source* attribute:

```

```

You can, and in many cases you will need to, use multiple attributes within a single element. Here is another image example, with all the attributes that you would typically see used for an image:

```

```

There are several attributes in place here, so let's break them down.

- src** "Src" is short for "source". This directs the browser to the image file that will be called with the HTML document is loaded. The image must be downloaded separately from the HTML file in order for the user to see it, so having many images, or large images, will slow down the process of loading the page.
- alt** "Alt" is short for "alternate text"—an accessibility measure which insures that those who can't see the image will be able to understand the content by reading a text description. You are legally required (encouraged?) to include accessibility information where necessary.
- height & width** These are pretty self-explanatory. The default unit of measure for these attributes is the pixel, so any number you put here will be interpreted as pixels.
- Other attributes** Each element has its own list of applicable attributes, and each attribute accepts a specific kind of value. The only way to know what your options are is to consult a reference (like [www.W3Schools.com](http://www.W3Schools.com)) and then memorize these as you go along.

## Nesting Elements

As you develop your content in HTML you will find instances where one type of content will need to exist inside of another. For instance, perhaps you want to create a link within a paragraph. This is called *nesting elements*, and it is common practice. Here's an example of what a link within a paragraph would look like:

```
<p>Academic Technology offers a wide range of equipment ready for faculty and staff to check out. <a href="equipment.html">Browse our available equipment.</a></p>
```

When multiple elements are nested, you must close the tags in order of last-created to first-created. Put another way, you have to work from the inside out. Here's a simplified representation of the principle (using fake element names):

```
<1>first <2>second <3>third</3></2></1>
```

If you do not close tags properly then you could end up with a final product that displays or behaves improperly, so be sure to look out for misplaced closing tags.

## Putting it all Together

I'm going to trust that you now have a basic understanding of how HTML markup works. There is a lot left to learn, but my hope is that you will now be able to look at HTML without being completely bewildered—maybe you can even understand some of it!

Remember that [matching content types exercise](#) (Pg. 8)? Here it is in plain HTML:

```
<!DOCTYPE html>
<html>
<head>
<title>ATS Solutions & Support - Equipment Checkout</title>
</head>
<body>
<h1>Equipment Checkout</h1>
<p class="subhead">Providing short-term equipment and device loans for ETSU faculty
and staff</p>
<hr />

<p>Academic Technology offers a wide range of equipment ready for faculty and staff to
check out. Need a video camera or audio recorder? Need a portable projector or just
want to see how an iPod works? Check the list below for equipment we have available,
and feel free to call ahead and reserve anything you know you might need for a certain
event. Some items are limited in quantity, so be sure to call early.</p>
<ul>
<li>Cameras</li>
<li>Laptops</li>
<li>Microphones & Recorders</li>
<li>PC Peripherals</li>
<li>Projectors</li>
<li>Response Systems</li>
<li>Tablets, eReaders, & Mobile Devices</li>
</ul>
<a href="equipmentDetails.html">Read More >></a>
</body>
</html>
```

## Missing Pieces

There's a lot of code in the example above that we didn't cover, but that is okay. This guide isn't going to provide you with a comprehensive understanding of HTML in one sitting. For now you just need to be able to examine an example and identify the building blocks.

Before we move on, let's reinforce some of this information with a quick quiz!

# Quiz:

## HTML Basics

- 1. What is the function of HTML?**
  - a. It provides a system for annotating documents
  - b. It describes content
  - c. It delivers instructions to a web browser regarding how to display web page content
  - d. All of the above
  - e. None of the above
- 2. Which of the following definitions is false?**
  - a. An *empty element* does not contain content
  - b. An *element attribute* provides additional information about an element
  - c. *Tags* are coded markers which define elements within an HTML document
  - d. *HTML* is an acronym for “Hippopotamus Took My Lunch”
- 3. Which of the following is an empty element?**
  - a. `<p>Paragraph goes here...</p>`
  - b. `<hr />`
  - c. `<h1 class="blue">Blue Heading</h1>`
  - d. ``
  - e. a and c
  - f. b and d
- 4. Which of the following statements uses proper HTML syntax?**
  - a. `<p>Paragraph goes here...`
  - b. `<a href="webpage.html">Link</a>`
  - c. `<br >`
  - d. `<h1 class=blue>Blue Heading</h1>`
  - e. b and d
- 5. Which of the following statements uses properly nested elements?**
  - a. `<ul><li>List Item #1</li></ul>`
  - b. `<p><a href="webpage.html">Link</p></a>`
  - c. `<p>Name<br />E-mail address<br />Phone Number</p>`
  - d. a and c
  - e. None of the above



# Conclusion

If you've followed this guide all the way through then you should now have a complete, working HTML document integrated into your course. The process of creating HTML content may be long and cumbersome initially, but as you gain experience it will require less time and effort. (If you find that your time is already limited, you could, perhaps, find an assistant who could help you move your content into HTML.)

Remember that the reason we make this content in HTML is because of its incredible accessibility. HTML content will be accessible to anyone on any device. While the work is harder up at first, it will hopefully prevent you from having to worry about technical problems in the future.

# What about the WYSIWYG Editor?

I intentionally avoided the WYSIWYG editor for most of this guide, because if you don't understand the underlying HTML then the WYSIWYG editor will likely just create problems that you can't fix. However, once you get comfortable looking at HTML and fixing basic syntax problems I would encourage you to try out the WYSIWYG editor. I've already mentioned that it is your best option for incorporating images, but it can do some other things as well.

## Basic Text Editing

If you just need to fix a typo or change a word in a sentence then the WYSIWYG editor is a fine option. You may run into trouble, however, if you try to select a large portion of text (an entire paragraph, list, or heading) and then cut or edit it. Sometimes the WYSIWYG will select more underlying HTML than you anticipate, and then alter it when you try to make edits.

## Import Content from Microsoft Word

There is a special command in the WYSIWYG editor that helps you to paste content from Microsoft Word (it looks like a clipboard with a MS Word icon on top of it). It won't copy all the formatting perfectly, but it will help to avoid potential disasters when pasting content from Word.

## Basic Text Formatting

If you want to make some text italicized, or convert text to a heading, you can do so quickly in the WYSIWYG editor. There are also some paragraph alignment and character formatting options. These can all be used safely when they are being applied to specific sets of text.

## Working with Tables

The HTML code necessary to create tables is a bit monstrous. It is very tedious to manipulate the many tags that make up the average table, so I would recommend that you try using the WYSIWYG editor to create this kind of content. If your HTML suffers, you can check out [W3Schools](#).

## Inserting Special Content (Symbols, Equations, Multimedia)

D2L has some special features built into the WYSIWYG editor that allow you to insert special forms of content. It is possible to build this content in the HTML editor, but you are better off using the built in features.

# Exercise & Quiz Answers

**Matching Content Types (from top to bottom):** 3. Heading, 5. Horizontal Rule, 4. Image, 2. Paragraph, 1. Unordered List, 6. Hyperlink

**HTML Basics:** 1. d 2. d 3. f 4. b 5. d

## Additional Resources

W3Schools – HTML Element Reference:

<http://www.w3schools.com/tags/default.asp/>

D2L HTML Templates (ETSU ATS Modified):

[http://www.etsu.edu/academicaffairs/elearning/ats/training/exercisefiles/d2l\\_templates.aspx](http://www.etsu.edu/academicaffairs/elearning/ats/training/exercisefiles/d2l_templates.aspx)

D2L HTML Templates (Originals):

<http://www.desire2learn.com/products/accessibility/instructor-resources/HTML-templates/>