HTML Practice 1B

Marking Up Text

Introduction: This practice will take a look at how to make use of HTML block-level tags to markup textual content. These block-level tags add structure to your pages. When you are given content to place on a page, you need to determine how you will organize it. Focus on the meaning of the content when choosing the tags to use. After completing the markup process, you should not have any text inside the <body> and </body> that is loose or not an appropriate HTML tag. Part of creating a page that is well-formed means marking up all the text into appropriate HTML.

While HTML tags have a visual appearance when they are displayed in one of the browsers, do **NOT** use the appearance as the determining factor for deciding what HTML tags you choose to use for the content. I have had students mistakenly use an <h3> for a main heading on a page just because they thought the <h1> was too big. We will later be able to style and size the content exactly as we want with CSS. Instead of worrying about how things look, for now just focus on finding the most appropriate tag to fit the intended **meaning** of the content.

Headings: Titles and subtitles on a page should use <h1> thru <h6> tags as these are the 6 levels of HTML headings. Every page should have a **main heading** that describes the purpose of the page. Always use an <h1> tag for the page main heading. All second-level titles/sub-headings on a page should be <h2> headings. All third-level headings should be <h3> tags, on down. Very rarely will you find the need for <h5> or <h6> tags as most web pages do not have this complicated of a structure. Remember, that a heading should be short and concise so it is easy to scan quickly. If you have an entire sentence you want to make a heading, think of summarizing the sentence and then adding the summarized heading tag before the sentence itself in a following paragraph.

Paragraphs: Sentences of text that logically belong together should be placed in a paragraph. Do not worry about indenting paragraphs at this point. If you later decide that you want to style your paragraphs with a beginning indent, we can do this with CSS. You should never use space characters ( ) to indent a paragraph. When a text block of sentences is more that about 4 sentences long, you should break up this lengthy paragraph into two shorter paragraphs so your content is more readable. Ideally a paragraph would be more than a single sentence, but you may find you have content that is just a single sentence or phrase that should be placed in a paragraph. Where to begin and end paragraphs is a judgment call on your part as the web developer. So, **choose wisely**.

Lists: Content that is in the form of a list of things should be placed in ordered lists for items where the order matters or <ul> unordered lists for items where you dont care about the order. You can manage where the numbering of list items begins or the type of numbering by using attributes of the <ol> tag. Rarely will you find that the kind of list you need is a definition list <dl>. These lists are usually used for lists where you have a term and an associated definition. Styling of lists can be managed using CSS, so don't worry about how lists are formatted right now. We could even add small images as the bullets in an <ul> list using CSS. Just add the basic structure for the most appropriate kind of list and worry about its styling later after Project 2.

Quotations: Sometimes you will have page content that is a quotation from a person or another document. The first step when asking what HTML tag is most correct for this quotation is looking at the length of the quote. This kind of problem is similar when you are writing a term paper and you are deciding how to format a quotation. Short quotes are handled differently than long quotes. For longer quotations (usually more than 2 sentences long) use the

<blockquote> tag which is a block-level tag. Like a long quote in a term paper, the browser will automatically double indent both sides of a <blockquote> and add the empty line before and after it. Shorter quotations should be done inline within a paragraph with an inline HTML tag, the <q> tag. One of the biggest mistakes web developers make is using a <blockquote> just to indent a paragraph that is not a long quotation. There are more correct ways to implement indenting than misusing a <blockquote> tag.

Validation Note: You can have a paragraph or multiple paragraphs within a <blockquote></blockquote, but you cannot have a <blockquote> inside a paragraph.

Breaks and Horizontal Rules: Both the
 and the <hr> tags are self-terminating of empty HTML tags and there is no separate closing tag. Best practice places marked up page content within block-level elements so that they wrap within the window, no matter the size of the window. This default wrapping behavior is known as the **normal flow** of an HTML page.

Use the
 tag very sparingly because they force hard line breaks in your content and if the window is resized may leave awkward holes in your text. You cannot guarantee a user will never resize your window. If you are just trying to use
 tags to control the amount of white space between block-level elements, this should be done with CSS, not with inserting
 tags.

Horizontal Rules <hr> tags allow you to add a line between block-level element on your page. This can help you to keep like content logically together and help make your pages more readable. Styling a horizontal rule can be complicated and the align attribute for the <hr> tag is being deprecated or phased out. So for our early projects, just use the <hr> tag when you want to add a line to make your content more readable, but do not style it yet.

Instructions:

You may use the browser and the text editor of your choice for this practice.

Step 1. Copy the HTML1B zipped file found in this practice assignment to your desktop. Extract the HTML1B folder.
Click on the index.html filename and rename the file: **FAQ.html**. First let's view this FAQ.html page in the browser.
Notice, the page content has no separation, and is very unreadable.

Open the FAQ.html page in your text editor and take a look at the source code for the page.

Step 2. Notice that this page was created with the start.html file. The document-level tags we need to use are all correct. You can modify the <title> or <meta> tags if you want.
All the page content needed for a possible IS&T Department FAQ page has been dumped inside the <body> and </body> tags.
Your mission, should you choose to accept it is to markup this page text.

We will not worry about how the text is presented by the browser, but just make good decisions as to finding the most appropriate block-level HTML tags.

When the page content is structurally correct, our mission is done.

Step 3. First let's examine the page content to determine what text should be a main heading and sub-headings. Sometimes you may need to add an appropriate heading to a page if the content contributor has not provided this.

Looks like *Information Systems & Technology FAQs* should be the main heading. Make this an <h1> heading. There is a short paragraph just after the main heading. So make this a <p> paragraph.

Step 4. Now there are some decisions that have to be made on how to structure the *Questions* and *Answers*.

If we make the entire *Question* a heading, this would be too wordy for a good heading. The same goes for making the entire *Answer* a heading.

We could number the questions and answers in a list, but again these would make for very lengthy list items and make the content less readable.

We could use <hr> horizontal rules between each pair of question and answer and not use headings, but the user would have a hard time picking out each question and answer.

For now, we will make each instance of the *Question:* text and each instance of the *Answer:* text second-level headings <h2>. The actual text that makes up the question will be a <p>. As will most of the text that makes up the answer. If the answer is more complicated, we can provide more structure.

So go through and add the <h2> headings to just the *Question:* and the *Answer:* text on the page.

Save your FAQ.html page and view it in the browser. Already you can see the page is much more readable with just the main heading and sub-headings.

Step 5. Next, we will examine each of the *Answers:* to see if lists, blockquotes or another level of sub-headings may be needed to add structure.

For simple groups of sentences, make these into paragraphs <p>. Remember, if a paragraph is longer than about 4 sentences, you should break it up into two shorter paragraphs.

Step 6. Take a closer look at the second *Answer:* on the page. This looks like a step by step process. Since these steps have a certain order, we should use an ordered list <ol> to organize these items. I would make the *Follow this process:* text a short paragraph <p>. Next add the opening <ol> and the closing at the end of the list content. For each item in this process include the step in <li> tags.

Step 7. In the third *Answer:* on the page related to IT jobs, looks like we could use a short paragraph <p>. Then there looks like there are three sub-headings. These should be <h3> tags because these are third-level headings. And then there is a list of jobs under each of these sub-headings. Because we don't really care about the order of these items in the list, these lists can be unordered lists <ul>. Make each job in these lists <li> list items.

Step 8. The last *Answer:* on the page looks like it could be a paragraph <p>. When we learn how to add links to the page, we could eventually make the URL included in this paragraph an anchor <a> tag.

Step 9. To group together each *Question:* and *Answer:* add a <hr> horizontal rule between each *Question* and *Answer* pair on the page.

View the completed marked up FAQ.html page in the browser. While it may not look pretty yet, the page should be very readable. The page is quickly scanned with short concise headings and sub-headings, short paragraphs, lists and sections divided with horizontal rules.

Step 10. To submit this practice exercise, **upload this FAQ.html** file in the HTML1B assignment in your Canvas section.

Conclusion: Part of being a web developer is playing the role of architect. We are asked to structure page content into the most appropriate HTML tags. This process of placing content text into block-level and inline HTML tags is known as marking up the page. There may be more than one way to structure a page, but if you focus on making your page easy to read and very scannable, you cannot go wrong. At the end of this process, you should not have any page content outside of block-level tags.

Submission: Upon completion of this practice, upload the **FAQ.html** file you have modified in the Canvas ASSIGNMENTS area for this HTML Practice 1B by the due date.