HTML Practice 4A

Introduction: Many of the HTML formatting tags have been deprecated. They are being phased out in favor of using Cascading Style Sheets, CSS. This practice will help you to use inline CSS and embedded CSS for text formatting. Inline CSS is often used to temporarily test out a style. Embedded CSS is used to apply a set of style rules to a specific page. Often styles will be used on an entire Web site, so external CSS is favored over embedded CSS. We will cover external CSS in Chapters 13-16.

See the CSS Orientation and Text Formatting Slideshows for further information about Inline CSS, Embedded CSS and text-formatting with CSS.

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

Step 1.   Save the HTML4AFiles zipped file to your Desktop.
    Extract the HTML4A folder from the HTML4AFiles zipped file.

Step 2.   View the index.html file in a web browser.
    Notice that block-level elements have been used to organize this text from the IS&T Department FAQ information.
    While this content is structurally correct, the page is bland and carries no textual styling or formatting.

Step 3.   We will add some inline CSS formatting to the index.html file.
    In the <h1>Information Systems & Technology heading, use the style attribute to add the inline styles for
        style="text-align: center; font-family: Century, serif; font-variant: small-caps;"
    In the <h2>Frequently Asked Questions heading, use the style attribute to add the inline styles for
        style="text-align: left; font-family: Century, serif; font-size: 18px;"
    Save your file and then view it in a web browser to make sure these headings look the way you would like.

Step 4.   Use the span tag to surround the first instance of the text "Question:" Add the style attribute to add inline styles for style="font-weight: bolder; font-size: 15px; text-decoration: underline;"
    Use the span tag to surround the first instance of the text "Answer:" Add the style attribute to add inline styles for style="font-weight: bolder; font-size: 15px; text-decoration: underline;"
    Save your file and then view it in a web browser to make sure these headings look the way you would like.
    Note: Using underlined text often confuses a user because they think the text is a hyperlink.

Step 5.   While inline CSS can be used to tryout a set of styles, it should very rarely remain in your code.
    We will create an embedded style sheet on this page and then move the inline styles into the embedded style sheet.

In the <head> tag add the follow style tag to hold the embedded style sheet:

```
<style type= " text/css">

</style>
```

Step 6.   You should always create a body selector in an embedded style sheet. This body selector will set the default styles for the entire page. In Lesson 7 we will learn how to add a background-color and text color to the body selector, but for now we will just set text formatting styles for the page.

```
body {  font-family: Arial, sans-serif;
        font-size: 14px;
    }
```

Step 7.   Other selectors needed for the embedded style sheet are h1, h2, and h3.
Cut the style properties from the <h1> tag and move them to the h1 selector inside the <style> tag

```
h1 {      text-align: center;
          font-family: Century, serif;
          font-variant: small-caps;
    }
```

Cut the style properties from the <h2> tag and move them to the h2 selector inside the <style> tag

```
h2 {      text-align: left;
          font-family: Century, serif;
          font-size: 18px;
    }
```

Create an h3 selector inside the <style> tag

```
h3 {      text-align: left;
          font-style: italic;
    }
```

Create an ul selector inside the <style> tag

```
ul {      font-variant: small-caps;
    }
```

Save your file. View the index.html file in the browser to make sure you are happy with these style rules.

Step 8.   Next, we will create a user-defined class in the embedded style sheet for the "Question" and "Answer" text. The class name must begin with the . character. Because we want to give the class a name reflecting the use for both Question and Answer text, the class name will be .QA.   Add this class to
the <style> tag embedded style sheet.

```
.QA {    font-weight: bolder;
          font-size: 15px;
          text-decoration: underline;
    }
```

Applying a class to selected text needs to be done inside a <span> tag and using the class attribute. For each instance of the text Question or the text Answer in the code, add a <span> tag around the text.
Add the class attribute and class value. Notice when applying a class, do not use the . character in the name.

```
<span class= "QA">Question</span>
<span class= "QA">Answer</span>
```

Step 9. Modifying the appearance of links can be important, especially when the background color might make the link text hard to see. We will learn more about adding color in the lesson on images. For now, we will just modify the color on the link states using the pseudo-classes for the anchor selector. Add the following pseudo-class styles to the <style> embedded style sheet.

```
a:link  {        color: Red;
}

a:visited  {     color: Green;
}

a:hover  {       color: DarkGreen;
}
```

Save the index.html file and try out the link states. The browser may cache your page and so the default link state may only appear once. You can clear your browser cache to get back to this unvisited state.

Step 10. Validate your file. Upload your HTMLPractice4 folder to your Practices folder on the Yoda web server.

Conclusion: It is a delicate balance to choose the appropriate formatting styles for text you use on your pages. Most of the HTML formatting tags and attributes have been deprecated in favor of CSS. Using an embedded style sheet instead of inline CSS is more efficient and allows the separation of the presentation from the content of the page. We will learn how to use external CSS in a future lesson.

Submission: Validate your index.html page. Upload your entire HTML4A folder to the Yoda server under the Practices directory.