**INT102 Lab 9: Formatting Text (CSS) (1/2021)**

1. Download menu9.html.
2. Open the menu9.html in the browser. It should look like the picture below:

Graphical user interface, text, application, email

Description automatically generated

1. Open VS Code editor > Open the menu9.html.

**Formatting a menu**

1. Set the main text to appear in Verdana or some other sans-serif font. Instead of writing a rule for every element in the document, write one rule for the body element that will inherited by all elements it contain. Add this rule to the embedded style sheet in <style> … </style> in the head of the document, like this:

<style>

body {

font-family: Verdana, sans-serif;

}

</style>

1. Set a fancy font for the “Black Goose Bistro, Summer Menu” heading. Choose a free display font called “Marko One” from Google Web fonts ([www.google.com/webfonts](http://www.google.com/webfonts)). Google provide the code for linking the font file on their server to HTML files. Copy the code and place in the head of document as follows:

<head>

<meta charset="utf-8">

<title>Black Goose Bistro Summer Menu</title>

<link href='http://fonts.googleapis.com/css?family=Marko+One' rel='stylesheet'>

<style>

1. Write a rule that applies it to the h1 element. Notice Georgia or another serif font as a list “back-up” fonts:

body {

font-family: Verdana, sans-serif;

}

h1 {

font-family: "Marko One", Georgia, serif;

}

1. Save the document and reload the page in the browser. It should look like as below:

Graphical user interface, text, application, email

Description automatically generated

**Setting font Size**

1. There are many approaches to sizing text on web pages. One way is to set the font-size of the body element to 100%.

body {

font-family: Verdana, sans-serif;

font-size: 100% ;

}

1. The browser default of 16 pixels is a fine size for the main page text. To set the appearance of the heading level to be 24 pixels or one and a half times larger than the body text (24/16=1.5). Thus, add a new rule that set the size of the h1 to 1.5em or 150% as follows:

h1 {

font-family: "Marko One", Georgia, serif;

font-size: 1.5em ;

}

1. Make the h2 the same size as the body

h2 {

font-size: 1em ;

}

**Making text bold and italic**

1. Set all of the menu item names to be bold text by applying a style to the dt (definition term) element to make them all bold. Add this rule to the end of the style sheet, save the file and see it in the browser:

dt {

font-weight: bold ;  
}

1. Now that all the menu item names are bold. Some of text marked as strong isn’t standing out very well. Thus, we should make them italic for further emphasis. To do this, simply apply the font-style property to the strong element:

strong {

font-style: italic ;

}

1. Save and reload. It should look like the picture shown below:

Text, letter

Description automatically generated

**Using the shorthand font property**

1. To save space, we can replace all the font properties we’ve specified for the h1 element with one declaration with the shorthand font property:

h1 {

font: bold 1.5em "Marko One", Georgia, serif;

}

**Using selectors**

This time, we’ll add a few more style rules using descendant, ID, and class selectors combined with the font and color properties.

1. Add some attention-getting color to the “new item!” elements next to certain menu item names. They are marked up as strong, so we can apply the color property to the strong element. Add this rule to the embedded style sheet, save the file and reload it in the browser:

strong {

font-style: italic ;

color: tomato ;

}

That worked, but now the strong element “Very spicy” in the description is “tomato” red too, and that’ s not what to be. The solution is to use contextual selector that targets only the strong elements that appear in dt elements. Remove the color declaration from the strong element rule, and create a new rule that targets only the strong elements within definition list terms:

strong {

font-style: italic ;

~~color: tomato~~ ;  ~~/\* remove this line \*/~~

}

dt strong {

color: tomato ;

}

1. Look at the document source, and you will see that the content has been divided into three unique divs: info, appetizers, and entrees. We can use these to our advantage when it comes to styling. For now, let’s do something simple and apply a teal color to the text in the div with ID “info”. Because color inherits, we need to apply the property only to the div and it will be pass down to the h1 and p:

#info {

color: teal ;

}

1. Now let’s get a little fancier and make the paragraph inside the “info” section italic in a way that doesn’t affect the other paragraphs on the page. Again, a contextual selector is the answer. This rule selects only paragraphs contained within the “info” section of the document:

#info p {

font-style: italic ;

}

1. To give special treatment to all of the prices on the menu. Fortunately, they have all been marked up with span elements:

<span class="price">$3.95</span>

So now all we have to do is write a rule using a class selector to change the font to Georgia or some serif font, make the prices italic, and gray them back:

.price {

font-family: Georgia, serif;

font-style: italic;

color: gray;

}

1. Similarly, in the “info” div, to change the appearance of the spans that have been marked up as belonging to the “label” class to make the labels stand out:

.label {

font-weight: bold;

font-variant: small-caps;

font-style: normal;

}

1. Finally, there is a warning at the bottom of the page that you can make it small and red. It has been given the class “warning”, so use that class as a selector to target just that paragraph for styling.

p.warning {

font-size: small;

color: red;

}

1. Save the file and try it out in the browser.

Graphical user interface, text, application, email

Description automatically generated

1. Add a global changes to the body element. Change the font-family to Georgia that would be more sophisticated and appropriate for a bistro menu and use the line-height property to open up the text lines and make them easier to read. Make these updates to the body style, as shown:

body {

font-family: Georgia, sans-serif ;

font-size: 100%;

line-height: 1.75em;

}

1. Redesign the header section of the document. Remove the teal color setting by deleting that whole rule. Then, make the h1 olive green and the paragraph in the header gray.

~~#info { /\* delete \*/~~

~~color: teal;~~

~~}~~

h1 {

font: bold 1.5em "Marko One", Georgia, serif;

color: olive ;

}

#info p {

font-style: italic;

color: gray;

}

1. Next, make a fancy print menu by making a few key elements to center on the page using the text-align property. Write a rule with a grouped selector to center the headings and the info section.

h1, h2, #info {

text-align: center;

}

1. Make the “Appetizer” and “Main Courses” h2 headings kind of special. Instead of large, bold type, use all uppercase letters, extra letter spacing, and color to call attention to the headings. Here’s the new rule for h2 elements that includes all these changes.

h1 {

font: bold 1.5em "Marko One", Georgia, serif;

color: olive;

}

h2 {

font-size: 1em;

text-transform: uppercase;

letter-spacing: .5em;

color: olive;

}

1. Make those paragraphs right after the h2 headings. Let’s center those too and make them italic.

h1, h2, #info {

text-align: center;

}

h2 + p {

text-align: center;

font-style: italic;

}

1. Next, add a softer color “sienna” to the menu item names (in dt elements). Note that the strong elements in those dt elements stay tomato red because the color applied to the strong elements overrides the color inherited by their parents.

h2 {

font-size: 1em;

text-transform: uppercase;

letter-spacing: .5em;

color: olive;

}

dt {

font-weight: bold;

color: sienna;

}

strong {

font-style: italic;

}

dt strong {

color: tomato;

}

1. Finally, Add a drop shadow under the h1 heading.

h1 {

font: bold 1.5em "Marko One", Georgia, serif;

color: olive;

text-shadow: .1em .1em .2em lightslategray;

}

1. Save the file and try it out in the browser.

Graphical user interface, text, application

Description automatically generated

1. When you finish, rename the “menu9.html” to xxxxxxxxxxx.html (xxxxxxxxxxx is your student id ) and upload to LEB2.