Rev 1.0
April 2019

Lab 2: CSS: Configuring colour and text
WEB1201: Web Fundamentals

1 Introduction

This lab is intended to get you accustomed to CSS to modify the colour and text of your website from Lab 1. It is expected that you have completed Lab 1 and is well versed with HTML and the different terms associated with HTML.

2 Objectives

At the end of this lab, you should be able to:

1. Modify the colour and text of a web page using CSS.
2. Describe what CSS is and how it can be used to modify the colour and text of a web page.

3 Deliverables, timing and additional notes

The estimated time of completion for this lab is approximately one and a half (1.5) hours (assuming you have gone through the theory). This does **not** include the time you will need to answer the questions. You are to compile your work after every lab - this includes all codes (and comments where necessary), documentation, completed tasks and answered questions. **You have to submit your work at the end of this lab.**

4 Materials and Equipment

1. A computer with a web browser.
2. A text editor (Notepad, Notepad++, Sublime, Vi, Vim, etc.). It is highly recommended that you use a text editor that support syntax highlighting as it will make it easier to work.
3. Files from “Lab 1 - HTML”.

Note

You should have a logbook as a place to write down notes for labs. This will help you revise and reflect on the work you have done in the lab.

5 An introduction to Cascading Style Sheets (CSS)

In our earlier lab, we were able to layout (to some extent) the information we want to present using HTML elements but was not able to style them. In this lab we will be looking at Cascading Style Sheets (CSS) that will help us style different parts of our web page.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

5.1 What does “cascading” mean?

“Cascading” means that styles can fall (or cascade) from one style sheet to another, enabling multiple style sheets to be used on one HTML document.

A HTML document may have three or more style sheets associated with it including (in increasing priority):

1. The user agent[1] (typically a browser) style sheet
2. The user style sheet
3. The author style sheet

5.1.1 User agent style sheets

A user agent is any software that retrieves and presents Web content for end users or is implemented using Web technologies that help in retrieving, rendering and interacting with Web content. In most cases, your user agent will typically be the web browser. The browser will apply style sheets to all web documents. Although these style sheets vary from browser to browser, they all have common characteristics such as black text, blue links, purple visited links etc. These are referred to as a “default” browser stylesheet.

As soon as you, the author, apply a style sheet to a document, it will override the browser’s style sheet. This is because author style sheets will always override browser style sheets.

[1]We will use the term user agent and browser interchangeably since that is most likely the user agent you will be using.

5.1.2 User style sheets

A user is anyone who looks at your website. Most modern browsers allow users to set their own style sheets within their browser. These style sheets will override the browsers default style sheets - for that user only.

It is important to know that users can set color and background color for HTML documents. If you, as an author, do not specify a color or background color, a user's style sheet could be used. More importantly, you should either specify both color and background color or neither. If you only specify a color and the user has specified the same color for their background, your entire document may be inaccessible to this user.

5.1.3 Author style sheets

The author is the person who develops the website and in this subject, that will be you!

As soon as you apply a basic style sheet or an inline style to a page, you have added what is referred to as an "author style sheet". Everything you do, from choosing fonts, colours and laying out pages in CSS is done using author style sheets.

Author style sheets can be applied inside an HTML document or by linking to an external file. You can also use multiple style sheets on an particular document to take advantage of the cascade.

5.2 So which style sheets applies?

With the different style sheets, CSS declarations are applied in this order (from lowest to highest priority):

1. user agent (the browser)
2. user normal
3. author normal
4. author important[2]
5. user important

Without the use of the `!important` declaration author$\rightarrow$user$\rightarrow$ browser styles.

[2]You can set "`!important`" on any declaration and this will override other declarations for the same element. "`!important`" declarations override normal declarations.

6 Ways of inserting style sheets

There are three ways of inserting style sheets:

1. Embedded:
   (a) found in: the **head section** of the HTML document
   (b) applies to: entire web page
   (c) HTML style **element**
2. Inline:
   (a) found in: the **body section** of the HTML document
   (b) applies to: specific element
   (c) HTML style **attribute**
3. External:
   (a) found in: separate file with .css file extension
   associated with the HTML using a HTML `link` element in the head section of the file

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.
   css">
</head>
```

   (b) applies to: any web page linked to the ,css file
   This allows you to have a consistent look and feel across multiple web pages.
4. Imported:
   (a) found in: another web site, your site in another location, etc. (anywhere a URL can locate)
   (b) uses @import to import into another CSS file
   More instructions found here @import page at mozilla.org

7 CSS Selectors and Declarations

Style sheets are composed of style rules that describe the styling to be applied. Each rule has two parts: a selector and a declaration:

1. **CSS Style Rule Selector** The selector can be an HTML element name, a class name or an id name.
2. **CSS Style Rule Declaration** The declaration indicates the CSS property you are setting

(such as color) and the value you are assigning to the property.

For example, the CSS rule shown in Figure 1 below would set the color of the text used on a web page to blue. The selector is the body tag, and the declaration sets the color property to the value of blue.

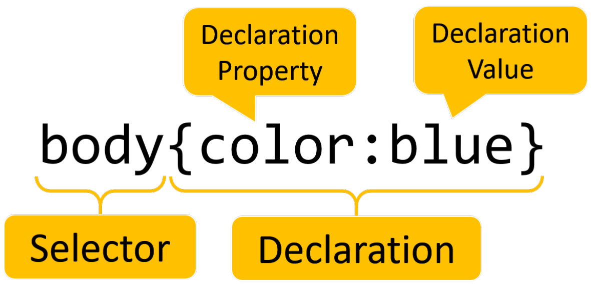


Figure 1: Examples of a CSS selector and declaration.

8 How to specify colour in CSS

There are several ways to specify colours in CSS.

1. Hexadecimal colors
2. RGB colours
3. RGBA colours
4. HSL colours
5. HSLA colours
6. Predefined/Cross-browser colour names

Here are some examples:

CSS Syntax	Colour Type
`p{color: red;}`	Using the colour name.
`p{color: #FF0000;}`	Hexadecimal colour value. Two hex values for each colour component
`p{color: #F00;}`	Shortened or shorthand hexadecimal. One hex value for each hex pair for use only with web-safe colours that have repeated hex pair values.
`p{color: rgb(255,0,0);}`	Decimal colour values (using RGB triplet values)
`p{color: hsl(0,100%,50%);}`	Hue Saturation Lightness value

You can find more information in this link.

8.1 Hexadecimal color values

1. Hexadecimal is the name for the base-16 numbering system, which uses the characters 0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F to specify numeric values.
2. The number represents a 24-bit value for colour (8-bits for each red, green and blue primary additive colours), broken down as #RRGGBB.
3. Begins with a # followed by 6 hex digits ranging from 00 to FF (0 to 255 in base 10)
4. The # symbol signifies that the value is hexadecimal. You can use either uppercase or lowercase letters in hexadecimal color values; `#FF0000` and `#ff0000` both configure the color red.

8.2 RGB(A) Colours

1. Specifies the intensity of each of the red, green and blue colour components.
2. It is specified in decimal values from 0 to 255, representing an 8-bit value with 256 different intensity values. 0$\rightarrow$no intensity, 255$\rightarrow$maximum intensity.
3. The RGBA version has an additional Alpha (A) channel with values that range from 0.0 to 1.0 that allows you to control the opacity of the colour. 0$\rightarrow$fully transparent, 1$\rightarrow$fully opaque. This is probably most noticeable when the Alpha channel is applied to a foreground object that will allow the background object to be partially seen.

8.3 HSL(A) Colours

1. Specifies the colour using the hue, saturation and lightness.
   (a) Hue:
   Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, 240 is blue. See Figure 2 for the HSL colour wheel to refer to the Hue colours.
   (b) Saturation:
   Saturation is a percentage value; 0% means a shade of gray and 100% is the full color.
   (c) Lightness:
   Lightness is also a percentage; 0% is black, 100% is white.
2. The HSLA version has an additional Alpha (A) channel with values that range from 0.0 to 1.0 that allows you to control the opacity of the colour. 0$\rightarrow$fully transparent, 1$\rightarrow$fully opaque.

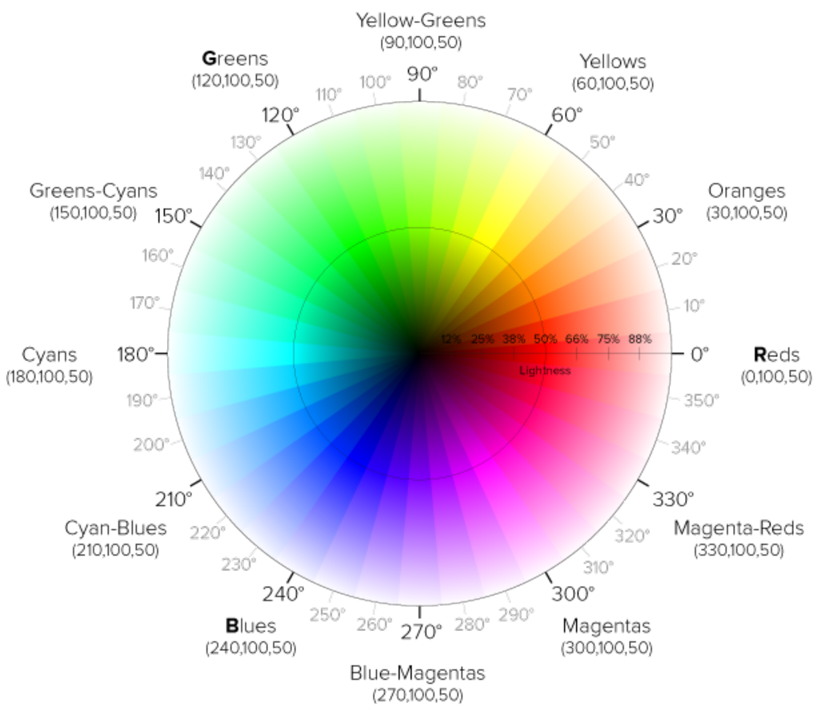


Figure 2: HSL colour wheel (from `https://tympanus.net/codrops/css_reference/hsl/`

8.4 Colour Names

1. You use one of the 140 predefined colour names.
2. Names are not case sensitive.

Key Word	Hexadecimal	RGB
Red	#FF0000	(255,0,0)
Yellow	#FFFF00	(255,255,0)
Lime	#00FF00	(0,255,0)
Green	#008000	(0,128,0)
Olive	#808000	(128,128,0)
Teal	#008080	(0,128,128)
Aqua	#00FFFF	(0,255,255)
Blue	#0000FF	(0,0,255)

Key Word	Hexadecimal	RGB
Navy	#000080	(0,0,128)
Purple	#800080	(128,0,128)
Fuchsia	#FF00FF	(255,0,255)
Maroon	#800000	(128,0,0)
Black	#000000	(0,0,0)
Gray	#808080	(128,128,128)
Silver	#C0C0C0	(192,192,192)
White	#FFFFFF	(255,255,255)

Task 1: Inline styling

In this task, we are going to specify a global body tag style: off-white background with teal text. These styles (the `color` and `background-color`) will be inherited by other elements by default. For example (shown without the closing tag):

```
<body style="background-color:#F5F5F5;
               color:#008080;">
```

For this lab, we will define two colours:

- off-white: #008080
- teal: #F5F5F5

You will modify `template.html` to include the style attributes added to the `<body>` and `<h1>` tags. Use the colours specified.

Steps:

1. Modify the title element to:
   Inline CSS example
2. In the body element, set body to have a teal background and off-white words.
3. Create a level 1 heading with the words:
   Inline CSS
4. Set h1 to have a off-white background and teal words.
5. Create a paragraph and Write the following in that paragraph:
   This paragraph inherits the styles applied to the body tag
6. Test your web page. It should look like Figure 3

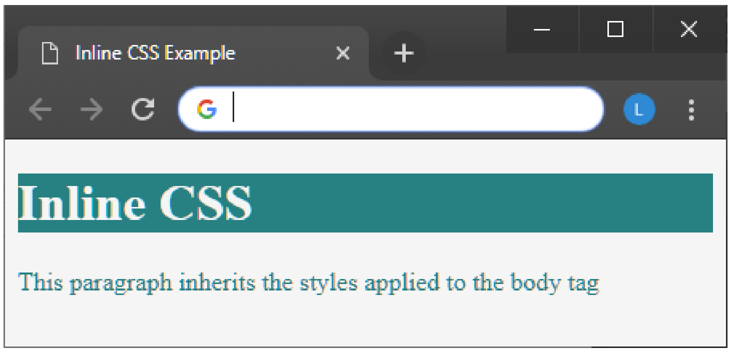
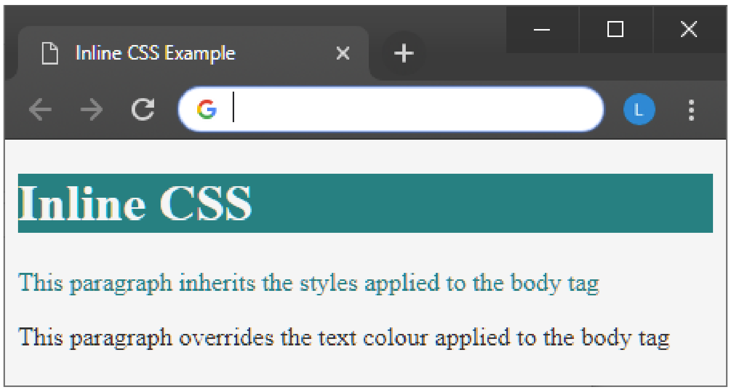


Figure 3: Example of inline CSS for styling the body and h1

7. Now add another paragraph and write the following in that paragraph:
   This paragraph overrides the text colour applied to the body tag.
8. Style just this paragraph (i.e. the 2nd paragraph) to have text with the colour #333333.
9. Test your web page. It should look like Figure 4
10. Save your final file as "styled_inline_CSS.html". You have to submit this file.

Your final code should look like the following:



Inline CSS

This paragraph inherits the styles applied to the body tag

This paragraph overrides the text colour applied to the body tag

Figure 4: Second example of inline CSS for styling the body and h1

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Inline CSS Example</title>
<meta charset="utf-8">
</head>
<body style="background-color:#F5F5F5;
                  color:#008080;">
<h1 style="background-color:#008080;
                  color:#F5F5F5;">
                  Inline CSS
            </h1>
<p>This paragraph inherits the styles applied to the body tag<
   /p>
<p style="color:#333333">This paragraph overrides the text
   colour applied to the body tag</p>
</body>
</html>
```

Task 2: Embedded CSS with Style Element

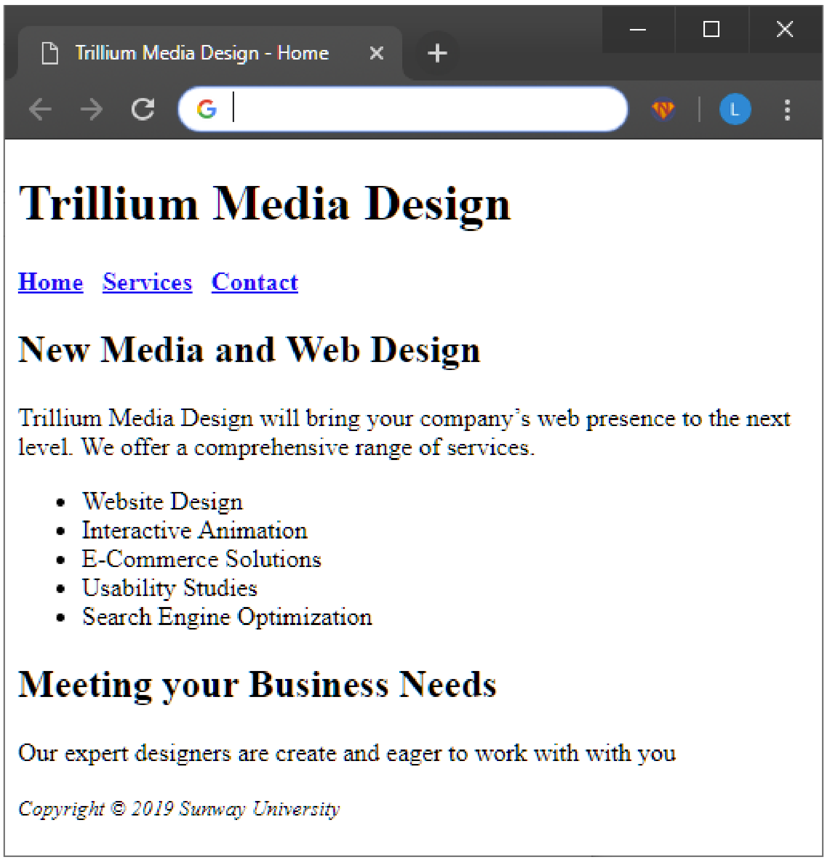
In the previous task, you added inline styles for one of the paragraphs. To do so, you coded a style attribute on the paragraph element. But what if you needed to configure the styles for 10 or 20 paragraphs instead of just one? Using inline styles, you might be doing a lot of repetitive coding! While inline styles apply to one HTML element, embedded styles apply to an entire web page.

- Embedded styles are placed within a `<style>` element located in the head section of a web page.

- The opening `<style>` tag and the closing `</style>` tag contain the list of embedded-style rules.

Use these colours as a reference.

- Dark purple: #191970
- Purple: #AEAED4
- Light purple: #E6E6FA



Trillium Media Design

Home Services Contact

New Media and Web Design

Trillium Media Design will bring your company's web presence to the next level. We offer a comprehensive range of services.

- Website Design
- Interactive Animation
- E-Commerce Solutions
- Usability Studies
- Search Engine Optimization

Meeting your Business Needs

Our expert designers are create and eager to work with with you

Copyright © 2019 Sunway University

Figure 5: Unstyled web page from the previous lab.

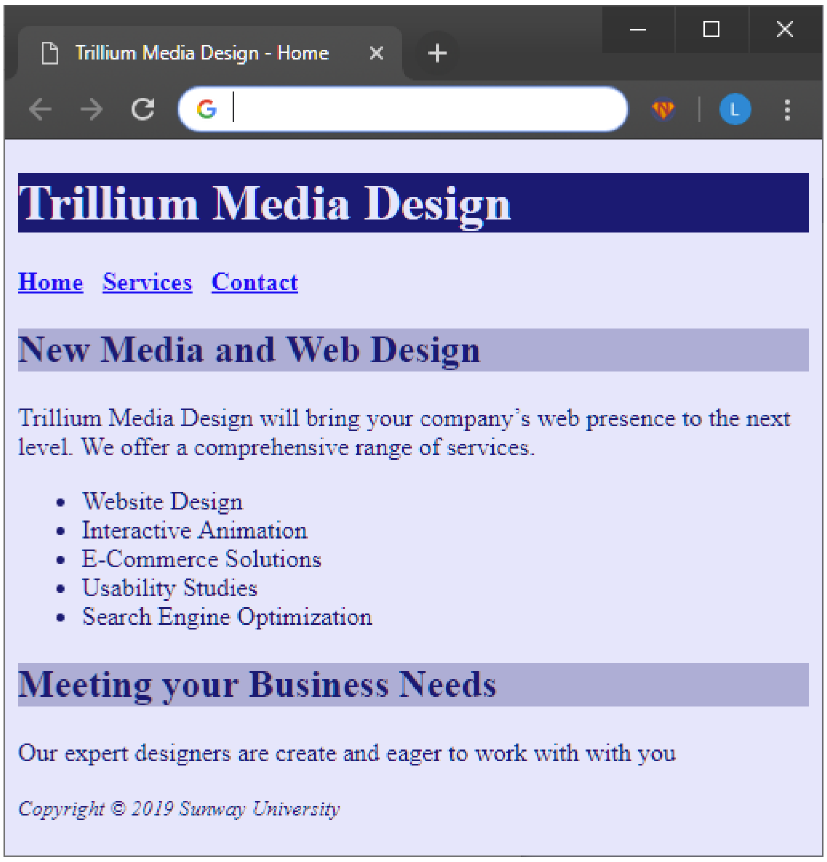
Steps:

1. Using your files from Lab1, modify the home page (index.html) to look like Figure 5. You will need to add additional HTML elements.
2. You are going to style the body, h1 and h2 using the `<style>` tags to the following:
   - body:
     background colour: light purple, colour: dark purple
   - h1:
     background colour: dark purple, colour: light purple
   - h2:
     background colour: purple, colour: dark purple

This is a monochromatic colour scheme. To help you with the coding, here is a snippet of only part of the codes shown here for body. You will have to do the rest for h1 and h2. Note: You need to put the `<style>` tags inside the head section of the HTML file.

```
<style>
  body{background-color: #e6e6fa; color: #191970}
</style>
```

3. After you have added the rest of the styling lines, your web page should look like Figure 6.
4. Save your file as "styled_embedded_CSS.html". Include this in your submission.



Trillium Media Design

Home Services Contact

New Media and Web Design

Trillium Media Design will bring your company's web presence to the next level. We offer a comprehensive range of services.

- Website Design
- Interactive Animation
- E-Commerce Solutions
- Usability Studies
- Search Engine Optimization

Meeting your Business Needs

Our expert designers are create and eager to work with with you

Copyright © 2019 Sunway University

Figure 6: Styled web page after applying the CSS.

What have you learnt from this exercise?

1. All the styles were in a single place on the web page. Since embedded styles are coded in a specific location, they are easier to maintain over time than inline styles.
2. Notice that you coded the styles for the h2 element selector only once (in the head section) and both <h2> elements applied the h2 style. This approach is more efficient than coding the same inline style on each <h2> element.

Further tasks:

These are additional tasks for you to explore.

1. Try using HSL to create roughly the same look.
   Hint: It is best to first determine the three (3) HSL values that you will be using before starting just as we did with the hex colour values earlier.
2. Try making the background of h1 and h2 semi-transparent by setting the opacity value.
3. Save your file as “styled_embedded_CSS_hsl.html”. Include this in your submission.

9 Configuring Text with CSS

In this section, you will learn to use CSS to configure font typeface. Using CSS to configure text is more flexible (especially when using an external style sheet, as you will discover later in the chapter) than using HTML elements and is the method preferred by modern web developers

9.1 `font-family` Property

The font-family property configures font typeface. A web browser displays text using the fonts that have been installed on the user's computer. When a font is specified that is not installed on your web visitor's computer, the default font is substituted. Times New Roman is the default font displayed by most web browsers. Table 1 shows font family categories.

Font family category	Font family description
serif	Serif fonts have small embellishments (or serifs) on the end of letter strokes; often used for headings.
sans-serif	Sans comes from the latin word “sine” to mean “without”. Sans-serif fonts are fonts *without* the serifs; often used for web page text.
monospace	Fixed-width font; often used for code samples.
cursive	Hand-written style; use with caution as it may be difficult to read on a web page.
fantasy	Exaggerated style; sometimes used for heading but use with caution as it may be difficult to read on a web page.

Table 1: Common fonts

9.2 `font-size` property

The font-size property sets the size of the font. Table 7 lists a wide variety of text and numeric values—there are almost too many choices available. See the notes in Figure 7 for recommended use

Notes about the em and percentage values

The em unit is a relative font unit that has its roots in the print industry, dating back to the day when printers set type manually with blocks of characters. An em unit is the width of a square

Value category	Values	Notes
Text Value	xx-small, x-small, small, medium (default), large, x-large, xx-large	Scales well when text is resized in browser; limited options for text size.
Pixel Unit (px)	Numeric value with unit, e.g. 10px	Pixel-perfect display depends on screen resolution; may not scale in every browser when text is resized.
Point Unit (pt)	Numeric value with unit, e.g. 10pt	Use to configure print version of web page; may not scale in every browser when text is resized.
Em Unit (em)*	Numeric value with unit; e.g. 0.75em	Recommended by W3C; scales will when text is resized in browser; many options for text size.
Percentage value*	Numeric value with percentage; e.g. 75%	Recommended by W3C; scales will when text is resized in browser; many options for text size.

*Recommended by W3C.

Table 2: Different options for configuring font size using CSS

block of type (typically the uppercase M) for a font and type size. On web pages, an em unit corresponds to the width of the font and size used in the parent element (typically the body element). So, the size of an em unit is relative to the font typeface and default size. Percentage values work in a similar manner to em units. For example, font size: 100% and font-size: 1em should render the same in a browser.

9.3 `text-shadow` property

The CSS3 text-shadow property adds depth and dimension to text displayed on web pages. Current versions of modern browsers, including Internet Explorer (version 10 and later) support the text-shadow property. Configure a text shadow by coding values for the shadow's horizontal offset, vertical offset, blur radius (optional), and colour:

1. Horizontal offset.
   Use a numeric pixel value. Positive value configures a shadow on the right. Negative value configures a shadow on the left.
2. Vertical offset.
   Use a numeric pixel value. Positive value configures a shadow below. Negative value configures a shadow above.
3. Blur radius (optional).
   Configure a numeric pixel value. If omitted, defaults to the value 0 which configures a

sharp shadow. Higher values configure more blur.

4. Color value. Configure a valid color value for the shadow.

Task 3: Changing the font styles

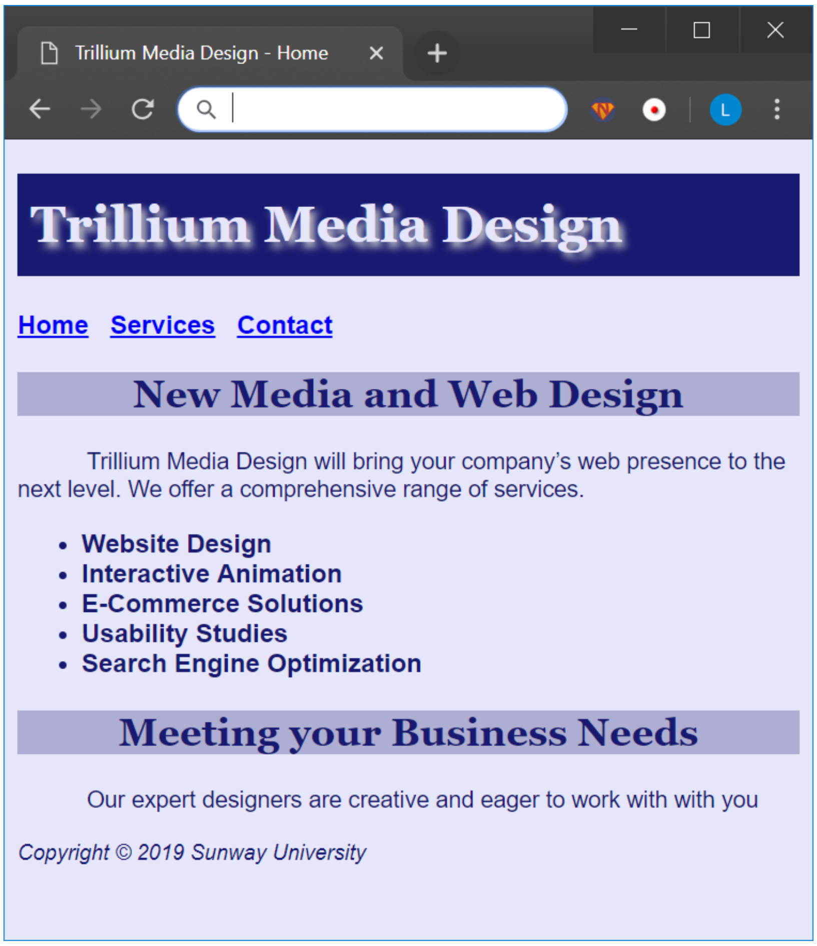
Steps:

1. Using your file "styled_embedded_CSS.html", modify to add in the last line of the style.

```
<style>
   body{ background-color: #e6e6fa;
         color: #191970;
         font-family: Arial, Verdana, sans-serif;}
</style>
```

   The new font typeface style rule shown in the following code will apply to the entire web page unless more specific style rules are applied to a selector (such as h1 or p), a class, or an id (more on classes and ids later).

2. Configure the h1 selector: (You add this as part of the `<style>`)
   - Set the `line-height` property: 200%
   - Set h1 to use a serif font. (Hint: use Georgia, Times New Roman and end with the general font family category)
   - Add a shadow with a gray (`#CCCCCC`) text shadow with a 3 pixel vertical and horizontal offset and a blur radius of 5 pixels.
   - Add a nonbreaking space special character in the body of the web page after the opening <h1> tag.
3. Configure the h2 selector: (You add this as part of the `<style>`)
   - Set the typeface to the same as h1.
   - Center the text. (Hint: `text-align`)
4. Configure the Paragraphs: (You add this as part of the `<style>`)
   - Text slightly smaller than the default text size: 0.90em
   - Configure the first line of each paragraph to be indented: 3em (Hint: `text-indent`)
5. Configure the unordered list by making the text bold. (Hint: Use an inline style and `font-weight`)
6. After you have added the rest of the styling lines, your web page should look like Figure 7.
7. Save your file as "styled_embedded_CSS_fonts.html". Include this in your submission.


Trillium Media Design - Home

Trillium Media Design

Home Services Contact

New Media and Web Design

Trillium Media Design will bring your company's web presence to the next level. We offer a comprehensive range of services.

- Website Design
- Interactive Animation
- E-Commerce Solutions
- Usability Studies
- Search Engine Optimization

Meeting your Business Needs

Our expert designers are creative and eager to work with with you

Copyright © 2019 Sunway University

Figure 7: Styled web page after applying the CSS font properties.

10 CSS Class, `id` and Descendant selectors

10.1 Class selector

Use a CSS class selector when you need to apply a CSS declaration to certain elements on a web page and not necessarily tie the style to a particular HTML element. When setting a style for a class, configure the class name as the selector. Place a dot, or period (.), in front of the class name in the stylesheet. The following code configures a class called "feature" (you can use any name) in a style sheet with a foreground (text) color set to a medium red and is added to the `<style>` section.

```
.feature{color:#C70000;}
```

The styles set in the new class can be applied to any element you wish. You do this by using the class attribute, such as class="feature". Do not type the dot in front of the class value in the opening tag where the class is being applied. The following code will apply the feature class styles to a `<li>` element:

```
<li class="feature">Usability Studies</li>
```

```
<li class="feature">Search Engine Optimization</li>
```

10.2 `id` Selector

Use a CSS `id` selector to identify and apply a CSS rule uniquely to a single area on a web page. Unlike a class selector which can be applied multiple times on a web page, an `id` **may only be applied once per web page**. When setting a style for an id, place a hash mark (#) in front of the `id` name in the style sheet. An `id` name may contain letters, numbers, hyphens, and underscores. `id` names may not contain spaces. The following code will configure an `id` called "main" in a style sheet:

```
#main{color:#333333;}
```

The styles set in the main id can be applied to any element you wish by using the `id` attribute, `id="main"`. Do not type the # in front of the id value in the opening tag. The following code will apply the `main id` styles to a `div` tag:

```
<div id="main">This sentence will be displayed using styles
   configured in the main id.
</div>
```

10.3 Descendant Selector

Use a CSS descendant selector when you want to specify an element within the context of its container (parent) element. Using descendant selectors can help you to reduce the number of different classes and ids but still allows you to configure CSS for specific areas on the web page.

For example, if you want to change the colour of all `<p>` elements, that are descendants of `<div>` elements, to "red".

```
div p {color:red;}
```

The descendant selector is one type of CSS combinator. A combinator is something that explains the relationship between the selectors. A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator. There are four different combinators in CSS:

1. descendant selector (space)
2. child selector (>)
3. adjacent sibling selector (+)
4. general sibling selector ( )

You are encouraged to read about the other combinators and how you can use them.

Task 4: Changing styles using classes

You will modify the CSS and the HTML in the Trillium Technologies page to configure the navigation and page footer areas

Steps:

1. For this task you will be using your file “styled_embedded_CSS_font.html”.
   In your `style` section, you are going to create a few classes to define styles.
2. Configure Nav Area:
   - Create a class called “nav” inside the `style` section.
   - To make the navigation links more prominent, we are going to display it in a larger (1.25em) and bold font.
   - Apply this style to the `nav` element.
3. Configure the Content Area:
   - Create a class named “feature” to display text colour as a medium dark red `(#C70000).`
   - Add this style to the last two items of the unordered list
4. Configure the Footer Area:
   - Create a class “footer” and apply it to the footer area.
   - Set the following properties: font colour: #333333, font size: 0.75em, font-style: italic
   - Apply this style to your footer.
5. Your web page should look like Figure 8.
6. Save your file as “embedded_CSS_class.html”.
7. Try applying two classes to one element. Record your observation in a text file called “<studentID>_observations.txt” where <studentID> is your Student ID. This is one of the files you will be submitting later on.

Note

Each task of your observation file should contain the following:

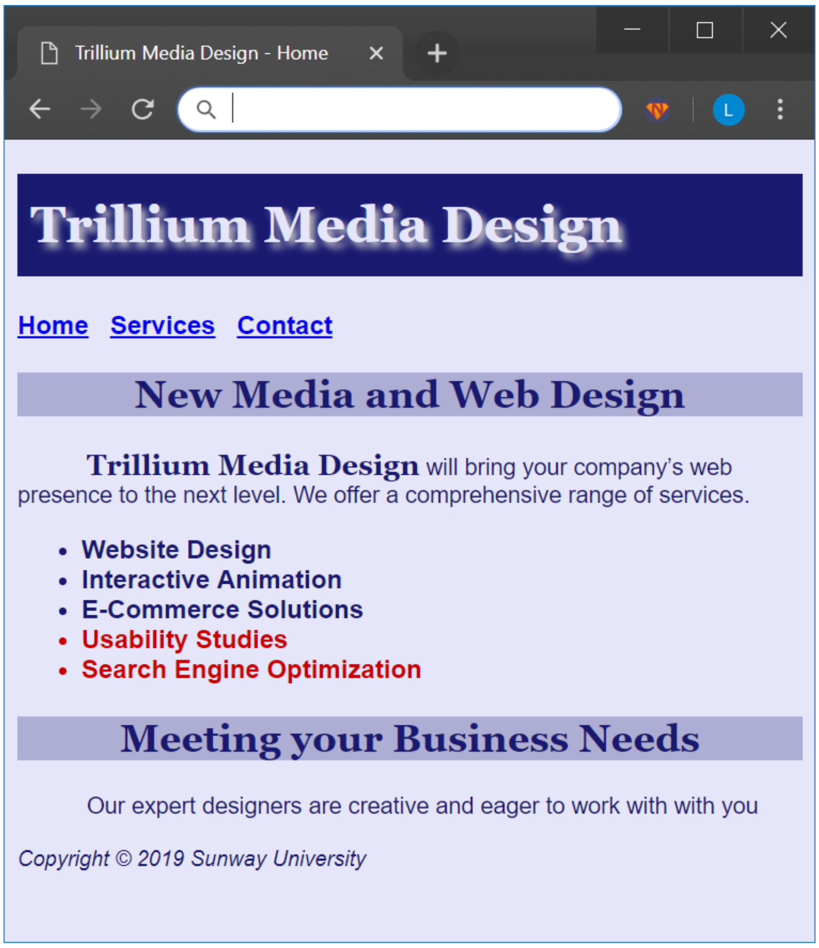
1. title of the task (just give it a short identifiable name)
2. what you did
3. what you observed
4. explain your observation
5. conclusion

What happens when you want to change just part of a text rather than the whole HTML element? We can use the `<span>` tags and enclose the part of the text in this element when we want to style it. We are going to try to do this by styling our company name.

1. Create a new CSS class called "company" with the following properties:
   (a) font weight: bold
   (b) font family: Georgia, Times New Roman, serif
   (c) font size: 1.25em
2. Apply this to just the company name. You can do this by doing:

```
<p><span class="company">Trillium Media Design</span> will
    bring....
```

3. Check if your company name appears different from the rest of the text.
4. Now change the company name to always appear as red.
5. Save your file as "embedded_CSS_span.html". Include this in your submission.



Trillium Media Design

Home Services Contact

New Media and Web Design

Trillium Media Design will bring your company's web presence to the next level. We offer a comprehensive range of services.

- Website Design
- Interactive Animation
- E-Commerce Solutions
- Usability Studies
- Search Engine Optimization

Meeting your Business Needs

Our expert designers are creative and eager to work with with you

Copyright © 2019 Sunway University

Figure 8: Styled web page using classes.

Task 5: Changing styles using id

In this task, you will use the id to apply styles to specific elements.

Steps:

1. Using id, change each colour of the first three items of the unordered list
2. Try using the same id in two different elements. Record your observation in your earlier observation text file.
3. Save your file as “embedded_CSS_id.html”. Include this in your submission.

Additional tasks

In this task, you will explore the following:

1. Create two classes of the same name. Apply the class name to an element. Record your observation in the text file.

11 Using external style sheets

The flexibility and power of CSS are best utilized when the CSS is external to the web page document. An external style sheet is a text file with a .css file extension that contains CSS style rules. The external style sheet file is associated with a web page by using the link element. This approach provides a way for multiple web pages to be associated with the same external style sheet file. The external style sheet file does not contain any HTML tags; it contains only CSS style rules.

11.1 The `link` element

The link element associates an external style sheet with a web page. It is placed in the head section of the page. The link element is a stand-alone, void tag. Three attributes are used with the link element: rel, href, and type.

- The value of the rel attribute is “stylesheet”.
- The value of the href attribute is the name of the style sheet file.
- The value of the type attribute is “text/css”, which is the MIME type for CSS.

The type attribute is optional in HTML5 and required in XHTML. Code the following in the head section of a web page to associate the document with the external style sheet named “colour.css” (which should be part of the <head> section of your HTML file):

```
<link rel="stylesheet" href="color.css">
```

Task 6: Using external style sheets

You are going to create a simple style sheet and associate it with a simple HTML file (i.e. “template.html” from Lab 1).

Steps:

1. Create a .css file with only the following:

```
body{background-color: #0000FF;
     color: #FFFFFF;}
```

2. Associate this CSS file to a copy of “template.html” (from Lab 1). Modify the title and text in the body as needed.
3. Save your file as “external_CSS.html”.

In this task, you have created a simple CSS and have associated it with a simple HTML file. This is just for you to test out how to associate a CSS to a HTML file. In the next tasks, you will be creating something more complex.

Task 7: Applying external style sheets to a website.

In this task you will now use external style sheets to apply the styles you have earlier created (the lavender coloured ones) to all three (3) of the pages of the web site you created in Lab 1.

Steps:

1. Put all the style elements you have created in this lab into a single CSS file. Put this file into a folder (called “css”) in the root folder of your website.
2. Link all the files of the website you created in Lab 1 to this style sheet.
   (Note: You will need to understand relative and absolute hyperlinks)
3. Save all your HTML files with the original file name. If you do not, you will need to modify the links in the file as needed.
   (Note: Move your entire root folder to another location and ensure that your web site still looks and behaves the same way.)
4. Center all the text on the web site using the `id` selector in your CSS. Use these properties:
   - margin-left: auto
   - margin-right: auto

    - width: 80%

    You will apply this `id` by wrapping the contents of the body in a `<div>` environment. To do this, all you need to do is add `<div>` just after the `body` opening tag in the HTML file and add `</div>` before the `body` closing tag.

5. Put all your website files under a single folder. You will later have to put this folder into your current working folder for submission.

Your web site folder structure should look like:

```
website_css ..................... This folder/directory holds all
 |                                your other web site files).
 |__css ......................... This folder/directory contains
 |   |                            all CSS files, which is
 |   |                            currently only one).
 |   |__external_css.css ........ This is your common CSS file
 |                                for the whole website.  All
 |                                other file should point to this
 |                                CSS file.
 |__index.html .................. This is your home page.
 |__contact.html ................ This is your contact page.
 |__services.html ............... This is your services page.
```

12 Summary

In this lab, you will have done the following:

1. Apply colour and text styles using inline CSS, embedded CSS and external CSS.
2. Create your own CSS file and link it to a HTML file.
3. Apply an external CSS file to several HTML files.

Instructions for submission can be found on the next page.

Submission

In your submission archive, you should have the following files:

1. files you have been asked to specifically name
2. your observation text file
3. an archive of your website (correctly named)

The submission link will be found on eLearn. Adhere to the guidelines and instructions on eLearn. If you are unfamiliar on how to create a 7zip archive, please ensure you leave sufficient time for submission. It is recommended that you submit your work 10 minutes before the end of time.

Note

As this is an official submission, you must not submit work that is NOT yours. Prevailing academic malpractice rules apply. Please refer to your student handbook what they are.

Example of the directory tree using 7zip as the archive for the student ID “00000000”. Do not use any other format than 7z.

```
00000000_Lab2.7z
├─ website_css(folder)
│  ├─ css(folder)
│  │  └─ external_css.css
│  ├─ index.html
│  ├─ contact.html
│  └─ services.html
├─ styled_inline_CSS.html
├─ styled_embedded_CSS.html
├─ styled_embedded_CSS_hsl.html
├─ styled_embedded_CSS_fonts.html
├─ embedded_CSS_span.html
├─ embedded_CSS_id.html
└─ 00000000_observations.txt
```

References

1. CSS from Maxdesign