|  |
| --- |
| **Course Code and Title:** ITE100- INTRO TO COMPUTING |
| **Lesson Number: Lesson 17 (Week 17)** |
| **Topic**: **Cascading Style Sheets (CSS)**  **C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify making web pages presentable.  CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices, and screen sizes as well as a variety of other effects.  CSS is easy to learn and understand, but it provides powerful control over an HTML document presentation. Most commonly, CSS is combined with the markup languages HTML or XHTML. |
| **Learning Objectives**  At the end of this lesson, you should be able to:   * Use adjust margin and padding. * Identify the Layout pages using CSS properties. * Apply advanced HTML tags (span and div) |
| **Pre-Assessment:**  Multiple Choice: Choose the letter of the best answer. Write your answer on the sheet provided (1 point each)   1. Which of the following is not a part of the box model?   A. Radius B. Margin C. Content D. Padding   1. Assuming the yellow area is the content, what is the area between the content and the border?   A. Height B. Margin C. Width D. Padding   1. What is the outermost area of the CSS Box Model?   A**.** width/height B. Border C. Margin D. Padding   1. When the CSS declaration *clear: both*; is applied to an element, it will appear below any floating elements above. 2. True B. False 3. Which of the following will add 20px of padding ONLY to the bottom of an element?   A. bottom-padding: 20px; C. padding: 20px 0;  B. padding: 0 20px 0 0; D. padding-bottom: 20px; |
| **What is CSS?**  **Cascading Style Sheets** (CSS) is used to format the layout of a webpage.  With CSS, you can control the color, font, text size, the spacing between elements, how elements are positioned and laid out, what background images or background colors to be used, different displays for different devices and screen sizes, and much more!  Tip: The word cascading means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue," all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!  **Using CSS**  CSS can be added to HTML documents in 3 ways:  **Inline** - by using the style attribute inside HTML elements  **Internal** - by using a <style> element in the <head> section  **External** - by using a <link> element to link to an external CSS file  The most common way to add CSS is to keep the styles in external CSS files. However, in this tutorial, we will use inline and internal styles because this is easier to demonstrate and easier for you to try.  **Inline CSS**  An inline CSS is used to apply a unique style to a single HTML element.  An inline CSS uses the style attribute of an HTML element.  The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:  Example  <h1 style="color:blue;">A Blue Heading</h1>  <p style="color:red;">A red paragraph.</p>  **Internal CSS**  An internal CSS is used to define a style for a single HTML page.  An internal CSS is defined in the <head> section of an HTML page, within a <style> element.  The following example sets the text color of ALL the <h1> elements (on that page) to blue and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powder-blue" background-color:  **Selected Style Sheet Topics**  **Grouping Selectors**  You can group selectors if they share the same properties. Just remember to separate each other by a comma. For example  h1, h2, h3, h4 { font-family:Arial; font-style: italic }    **Classifying Selectors**  Classifying selectors can speed up things. For example, you may want some paragraphs of your HTML document to be aligned to the left while others aligned at the center if you format each paragraph manually. Even if you are using the inline style sheet, it will take you a lot of time. Here's what you can do first, you need to identify the classes inside the style sheet. The Format for identifying naming classes is *.name*. For example, you may decide to call the classes .*one* and .*two*  The first class will format a paragraph aligned to the left and the second aligned at the center of your document.  p.one {text-align:left;}  p.two {text-align:center;}    Now we can move on the HTML document using the class attribute inside all the <p> opening tags  <p class="one">This paragrap will be left-aligned</p>  <p class="two">This paragrap will be left-centered</p>  **Page layout: Layout Table vs. CSS Positioning**  Before, almost everything connected to the Web page layout was done using tables. But now, building pages is much simpler through the positioning method of CSS.  In the previous lesson, we dealt with data tables in which information is organized into rows and columns. We can use a table to organize content on the screen through borderless tables. In these layout tables, the information in a particular cell necessarily implies a relationship with information in other cells, unlike in data tables.  On the other hand, CSS has properties that always come in handy when a content presentation is concerned. We are talking about the position property, which has the absolute, relative, static, and fixed values; the float property has the values left and right, and the clear property has the values left, right, and both. Choose any element and specify which part of the Web pages you want that element to appear using a property.  **CSS Positioning Properties**   |  |  |  | | --- | --- | --- | | Description | Properties | Example | | Absolute - The element stays in one place. Even if you resize the browser window, it will remain in the part of the page you assigned it to. | Left, top, right, and bottom. | Abs {  position: absolute;  bottom: 4em;  right: 0; } | | Relative – Its position depends on other elements. Even if you resize the browser window, it will not overlap into other elements (unlike the absolute value). | Left, top, right and bottom | .rel {  position: relative;  top: 0; } | | Static – The default position. The element would be in this position if you did not specify any value. |  | hr {  position: static;  } | | Fixed – The element stays in place even if you scroll up and down, left or right. Its position depends on the browser windows and not on the page itself (unlike the absolute value) | Left, top, right, and bottom. | Pre {  Position: fixed;  Left: 0; | | Float – Its position is either to the left or right of another element, with all the other contents enclosing it. | Left and right | . left {  float: left;  } | | Clear – So that the element will not have any other elements around it (unlike the *float* value) | Left, top, right, and bottom | .right {  clear: right;  } |   **Span and Div Tags**  The <span> tag is an inline container used to mark a part of a text or a part of a document.  The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.  The <span> tag is much like the [<div>](https://www.w3schools.com/tags/tag_div.asp) element, but <div> is a block-level element and <span> is an inline element.  **CSS Box Model**  All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.  The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:    Explanation of the different parts:   * **Content** - The content of the box, where text and images appear * **Padding** - Clears an area around the content. The padding is transparent. * **Border** - A border that goes around the padding and content * **Margin** - Clears an area outside the border. The margin is transparent.   The box model allows us to add a border around elements and to define space between elements.  Demonstration of the box model:    Source code    Output  **Margin and Padding**  The CSS **margin** properties are used to create space around elements outside of any defined borders.  With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).  CSS has properties for specifying the margin for each side of an element:   * margin-top * margin-right * margin-bottom * margin-left   All the margin properties can have the following values:   * auto - the browser calculates the margin * *length* - specifies a margin in px, pt, cm, etc. * *%* - specifies a margin in % of the width of the containing element * inherit - specifies that the margin should be inherited from the parent element   **Tip:** Negative values are allowed.  Set different margins for all four sides of a <p> element:    The CSS ***padding*** properties are used to generate space around an element's content inside any defined borders.  With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).  CSS has properties for specifying the padding for each side of an element:   * padding-top * padding-right * padding-bottom * padding-left   All the padding properties can have the following values:   * *length* - specifies padding in px, pt, cm, etc. * *%* - specifies padding in % of the width of the containing element * inherit - specifies that the padding should be inherited from the parent element   Set different padding for all four sides of a <div> element:    Source Code  To shorten the code, it is possible to specify all the padding properties in one property.  The ***padding***property is a shorthand property for the following individual padding properties:   * padding-top * padding-right * padding-bottom * padding-left   So, here is how it works:  If the ***padding***property has four values:   * **padding: 25px 50px 75px 100px;**   + top padding is 25px   + right padding is 50px   + bottom padding is 75px   + left padding is 100px   Example: Use the padding shorthand property with four values: |
| **Synthesis-summary:**  Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify making web pages presentable.  Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices, and screen sizes as well as a variety of other effects.  CSS is easy to learn and understand, but it provides powerful control over an HTML document presentation.  Most commonly, CSS is combined with the markup languages HTML or XHTML. |
| **Evaluation:**  **Multiple Choice: Choose the letter of the best answer.** Write your answer on the sheet provided**. (1 point each)**  **Evaluation**  Multiple Choice: Write the correct answer. Submit to the Google classroom or LMS answer only.   1. Which of the following is not a part of the box model?   A. Radius B. Margin C. Content D. Padding   1. Assuming the yellow area is the content, what is the area between the content and the border?   A. Height B. Margin C. Width D. Padding   1. What is the outermost area of the CSS Box Model?   A**.** width/height B. Border C. Margin D. Padding   1. When the CSS declaration *clear: both*; is applied to an element, it will appear below any floating elements above. 2. True B. False 3. Which of the following will add 20px of padding ONLY to the bottom of an element?   A. bottom-padding: 20px; C. padding: 20px 0;  B. padding: 0 20px 0 0; D. padding-bottom: 20px;   1. Give a acronym of CSS 2. Computer System Service C. Cascading Style Sheets 3. Cascading Sheets Style D. Cascading Short Sheets 4. Which CSS code centers text? 5. align: center; B. align=center; C. text-align: center; D. center=align; 6. Assuming the yellow area is the content, what is the color of the margin area. 7. Pink B. Red C. Green D. Blue 8. Select the CSS **shorthand** declaration that accomplishes the result of 5px of padding on all sides of an element. 9. padding: 5px; 10. padding-all-sides: 5px; 11. padding-top: 5px; padding-bottom: 5px; padding-left: 5px; padding-right: 5px; 12. padding: 5px all; 13. If an element is set to *display: inline*;, it will not apply top or bottom margins or any *width* values given to it. 14. True B. False |
| **Reinforcement Activity**  Create this activity title Using Individual padding properties    Output    Source Code |
| **References:**  Florida Valencia Ortiz, Emmanuel Kazanidis and Jaime D.L. Caro, Ph D (Web Design)  Website:  w3schools.com;<https://www.w3schools.com/html/>  <https://www.tutorialspoint.com/html/html_fonts.htm#:~:text=Font%20face%20and%20color%20depends,size%2C%20face%2C%20and%20color.> |