Lesson Proper for Week 8

What is CSS?

Cascading Style Sheets (CSS).

§ CSS is a W3C (World Wide Web Consortium) the standard for describing the appearance of HTML elements.

§ Another common way to describe CSS's function is to say that CSS is used to define the presentation of HTML documents. With CSS, we can assign font properties, colors, sizes, borders, background images, and even position elements on the page.

Defining Style Sheets

A style sheet is usually a text file that is separate from your HTML document. Style sheets can also be internal, residing within your HTML code. A style sheet holds formatting codes that control your Web page's appearance. You can use style sheets to change the look of any Web page element, such as paragraphs, lists, backgrounds, and more.
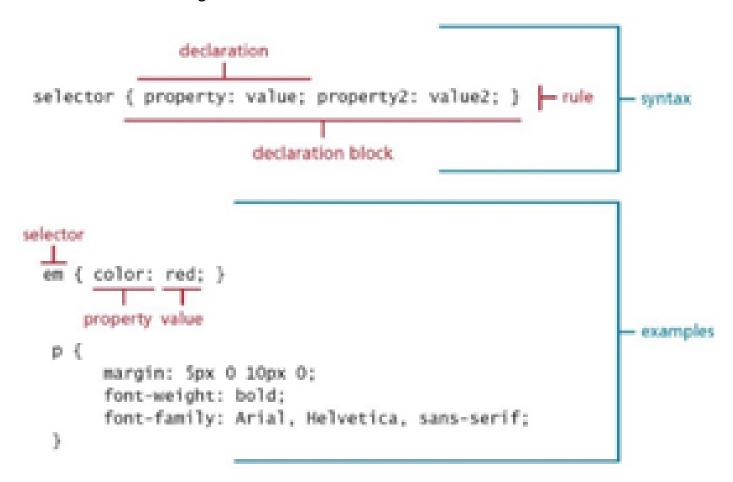
Style Sheet Syntax

Style sheets are made up of rules, and each rule has two distinct parts: a selector and a declaration. The selector specifies the element to which you want to apply a style rule, and the declaration specifies the formatting for the selector. For example, in the style rule H2 {color: silver}, the selector is H2 and {color: silver} is the declaration. When applied to a page, this rule will make all level 2 headings appear in silver.

A CSS document consists of one or more **style rules**. A rule consists of a **selector** that identifies the HTML element or elements that will be affected, followed by a series of ***property:value*** pairs (each pair is also called a **declaration**).

The series of declarations is also called the **declaration block**

A ***declaration*** consists of one or more property and value pairs such as font-size: 12px or position: absolute. The **property** and **value** are separated by a **colon " ; "** multiple property-value pairs in a declaration are separated by **semicolons**. It is good form to put each ***property-value*** pair on a separate line when writing your rules. Similar to HTML, you can add extra spaces and line breaks to your style sheet code to make it more readable.

Different method of declaring CSS

CSS Selectors

Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern that is used by the browser to select the HTML elements that will receive the style.

CSS Element Selector	
| ```
h1 {
    color: #36CFFF;
}
``` | The element selector selects the HTML element by name. |
| CSS Id Selector | |
| ```
#black {
    color: #000000;
}
``` | The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element. |

CSS Class Selector

CSS Class Selector
 	
``css
.black {
    color: #000000;
}
``
	The class selector selects HTML elements with a specific class attribute. It is used with a period character (.) full stop symbol followed by the class name.

CSS Universal Selector
 	
``css
* {
    color: #000000;
}
``
	The universal selector is used as a wildcard character. It selects all the elements on the pages.

CSS Group Selector
 	
``css
h1, h2, h3 {
    color: #36C;
    font-weight: normal;
    letter-spacing: .4em;
    margin-bottom: 1em;
    text-transform: lowercase;
}
``
	The grouping selector selects all the HTML elements with the same style definitions.

Where You Can Add CSS Rules

CSS rules can also appear in two places inside the HTML document:

*   Inside a <style> element, which sits inside the <head> element of a document.

*   As a value of a style attribute on any element that can carry the style attribute.

When the style sheet rules are held inside a <style> element in the head of the document, they are referred to as an internal style sheet.

```html
<head>
    <title>Internal Style sheet</title>
    <style type="text/css">
    body {
        color:#000000;
        background-color:#ffffff;
        font-family:arial, verdana, sans-serif; }
    h1 {font-size:18pt;}
    p {font-size:12pt;}
    </style>
</head>
```

When style attributes are used on XHTML elements, they are known as inline style rules.

The <link> Element

§ Is used in web pages to describe the relationship between two documents; for example, it can be used in an HTML page to specify a style sheet that should be used to style a page. You may also see the <link> element used in HTML pages for other purposes, for example to specify an RSS feed that corresponds with a page.

It is a very different kind of link than the <a> element because the two documents are automatically associated — the user does not have to click anything to activate the link.

The <link> element is always an empty element, and when used with style sheets it must carry three attributes: type, rel, and href. Here is an example of the <link> element used in an HTML page indicating that it should be styled by a CSS file called interface.css, which lives in a subdirectory called CSS:

The <rel> Element

§
The rel attribute is required and specifies the relationship between the document containing the link and the document being linked to. The key value for working with style sheets is stylesheet.

The <type> Element

The type attribute specifies the MIME type of the document being linked to; in this case, we are dealing with a CSS style sheet, so the MIME type is text/css:.

The <href> Element

The href attribute specifies the URL for the document being linked to.

The value of this attribute can be an absolute or relative URL, but it is usually a relative URL because the style sheet is part of the site.

CSS Properties

Each individual CSS declaration must contain a property. These property names are predefined by the CSS standard.

Most of the common CSS properties:

Property Type	Property

Spacing	padding
	padding-bottom, padding-left, padding-right,
	   padding-top
	margin
	margin-bottom, margin-left, margin-right,
	   margin-top

Sizing	height max-height max-width min-height min-width width
Layout	bottom, left, right, top clear display float overflow position visibility z-index
Lists	list-style list-style-image list-style-type

Property Values

Each CSS declaration also contains a value for a property. The unit of any given value is dependent upon the property. Some property values are from a predefined list of keywords. Others are values such as length measurements, percentages, numbers without units, color values, and URLs.

For each selector there are "properties" inside curly brackets, which simply take the form of words such as color, font-weight or background-color.

A value is given to the property following a colon (NOT an "equals" sign). Semi-colons are used to separate the properties.

Example:

body {

  font-size: 14px;

  color: navy;

}

This will apply the given values to the font-size and color properties to the body selector.

Lengths and Percentages

Here are many property-specific units for values used in CSS, but there are some general units that are used by a number of properties and it is worth familiarizing yourself with these before continuing.

* px (such as font-size: 12px) is the unit for pixels.

* em (such as font-size: 2em) is the unit for the calculated size of a font. So "2em", for example, is two times the current font size.

* pt (such as font-size: 12pt) is the unit for points, for measurements typically in printed media.

* % (such as width: 80%) is the unit for... wait for it... percentages.

Other units include pc (picas), cm (centimeters), mm (millimeters) and in (inches).
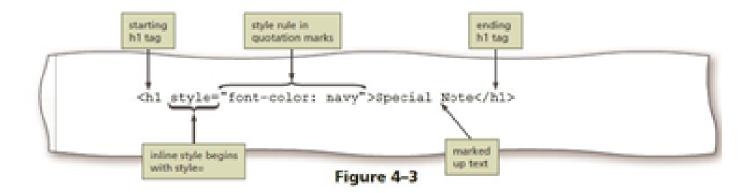
When a value is zero, you do not need to state a unit. For example, if you wanted to specify no border, it would be border: 0.

Internal Style Sheet

You can create an internal style sheet that resides within the <head> tag of your HTML document. Internal style sheets are handy if your Web site consists of a single page because you can change both style rules and HTML in the same file.

Inline Styles

With an inline style, you add a style to the start tag for an element, such as a heading or paragraph, using the style attribute. The style changes the content marked up by a specific pair of tags, but does not affect other content in the document. Because inline styles take precedence over other types of styles and affect the style for individual pieces of content, they are helpful when you need to format only one section of a webpage in a unique way. However, inline styles defeat the purpose and advantages of separating style from content, so they should be used sparingly. An example of an inline style is shown in Figure 4–3.
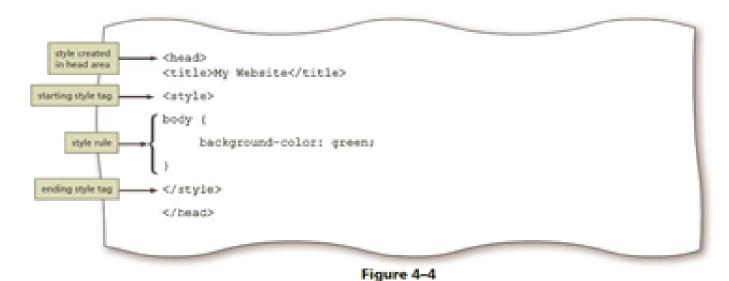
Figure 4–3

The style rule applies only to the content in this h1 element, the "Special Note" text. In the starting <h1> tag, you begin writing an inline style using the style attribute. The inline style shown in Figure 4–3 defines the font color of the "Special Note" text as navy blue.

You can link to a style sheet to assign a set of formatting rules to your HTML document. You can link multiple documents to the same style sheet to give all the pages in your site a consistent look and feel.

Embedded Style Sheets

An embedded style sheet, also called an internal style sheet, includes the style sheet within the opening <head> and closing </head> tags of the HTML document. Use an embedded style sheet when you want to create styles for a single webpage that are different from the rest of the website. An embedded style sheet takes precedence over an external style sheet. An example of an embedded style sheet is shown in Figure 4–4.

Figure 4–4

In embedded style sheets, you place the style rules between the opening <style> and closing </style> tags. The style rule shown in Figure 4–4 sets the background color for the body section to green for the current webpage only.

External Style Sheets

An external style sheet, also called a linked style sheet, is a CSS file that contains all of the styles you want to apply to more than one page in the website.

You can use an external style sheet to define formatting and layout instructions and then apply those instructions to your HTML documents. You can save the style sheet as a text file and assign the .css file extension to identify the file as a Cascading Style Sheet.

Figure 4–5

An example of a style rule for an external style sheet is shown in Figure 4–5.

Style Sheet Precedence

Style sheets are said to "cascade" because each type of style has a specified level of precedence (or priority) in relationship to the others. For example, suppose you create an inline style to change the text color of an h1 heading to red. In the head section of a webpage, you also create a style in an embedded style sheet to change h1 headings to blue, and in an external style sheet, you create a style to change h1 headings to green. What color would h1 headings be in the webpage with the inline style? They would be red because the style closest to the content takes precedence. In other words, inline styles beat embedded styles, and embedded styles beatexternal styles.

CSS properties can be inherited from a parent element. This principle is called inheritance. For example, paragraphs and headings inherit the font and color rules for the body selector. If a selector has more than one CSS rule, specificity determines which CSS rule to apply. The more specific selector is applied.

For example, if the value of the background-color property for the body selector is green but the one for p is blue, the p elements will have a blue background because the p selector is more specific than the body selector.

The best practice is to apply inline styles when you want to control the style of content within one pair of HTML tags, an embedded style sheet when you want to change the style of one page, and external or linked style sheets for the styles that apply to many or all pages in the website.

CSS Text Properties

You can use CSS styles to format webpage text in a variety of ways. For example, use the font-family property to define a specific font. Use the font-size property to define a specific font size.

Table 4-4 Common Text Properties and Values

Table 4–1 Common Text Properties and Values

Property	Description	Common Values
font-family	Specific and general font names	font-family: Cambria, "Times New Roman", serif; font-family: Verdana, Arial, sans-serif; font-family: Georgia, "Times New Roman", serif;
font-size	Absolute or relative size of a font	font-size: 1.5em; font-size: 50%; font-size: x-large; font-size: 14pt;
font-weight	Weight of a font	font-weight: bold; font-weight: bolder; font-weight: lighter;
font-style	Style of a font	font-style: normal; font-style: italic; font-style: oblique;
text-align	Alignment of text	text-align: center; text-align: right; text-align: justify;
color	Color of text	color: red; color: blue; color: green;

Notice that the font-family property may include multiple values. In fact, you should provide more than one value for this property in case the browser does not support the primary font. The additional values are called fallback values. If the browser does not support the primary font, it displays the second font family indicated and if the browser does not support the second font family value, the browser uses the next font family.

Commas separate each value. The desired value is listed first and the value of serif or sans-serif is listed last. For example, the declaration font-family: Cambria, "Times New Roman", serif; means that the browser should use the Cambria font; if the browser cannot use Cambria, it should use Times New Roman, which is listed in quotation marks because the font family name contains more than one word. Finally, if the browser does not support Cambria or Times New Roman, it should use its default serif font.

Why would a browser not support certain fonts?
Fonts are installed on a computer, so a computer must have the font installed before a browser can display it. For a list of common web fonts, visit www.w3schools.com/cssref/css_websafe_fonts.asp.

CONSIDER THIS

CSS Colors

One way to help capture a webpage visitor's attention is to use color as a webpage background or for text, borders, or links. HTML uses color names or codes to designate color values. When using a color name, you specify a word such as aqua or black as a value. Following are 16 basic color names.

Table 4– 3 shows a common list of colors, with the corresponding hexadecimal and RGB color codes. You can use

two types of color codes with CSS: hexadecimal and RGB.

Table 4–3 Color Values		
Color	**Hexadecimal**	**RGB**
Black	#000000	rgb(0,0,0)
White	#FFFFFF	rgb(255,255,255)
Red	#FF0000	rgb(255,0,0)
Green	#008000	rgb(0,128,0)
Blue	#0000FF	rgb(0,0,255)
Yellow	#FFFF00	rgb(255,255,0)
Orange	#FFA500	rgb(255,165,0)
Gray	#808080	rgb(128,128,128)

Hexadecimal values consist of a six-digit number code that corresponds to RGB(Red, Green, Blue) color values. When noting color values in CSS, include a number sign (#) before the code. Hexadecimal is a combination of the base-16 numbering system, which includes letters A through F.

RGB notation is used to display colors on a screen, though not in print. RGB blends red, green, and blue color channels to create a color. (A channel contains the number of red, green, or blue pixels necessary to create a specified color.) Each color channel is expressed as a number, 0 through 255.

To use a color in a style rule declaration, use the color value as the property value. For example, to style a background color as gray, you use the background-color property with a value of #808080, as shown in the following example:
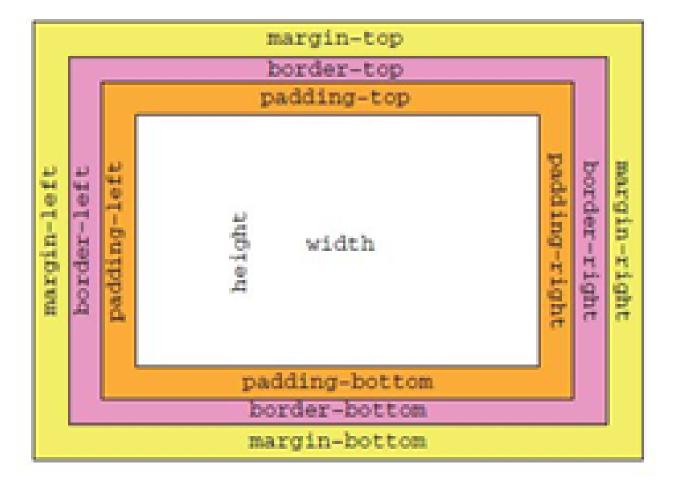
```
background-color: #808080;
```

Think Inside the Box

In CSS, all HTML elements exist within an element box. In order to become proficient with CSS, you must become familiar with the element box.

According to the box model, every element in a document generates a box to which properties such as width, height, padding, borders, and margins can be applied. You probably already have a feel for how element boxes work, from adding backgrounds to elements.
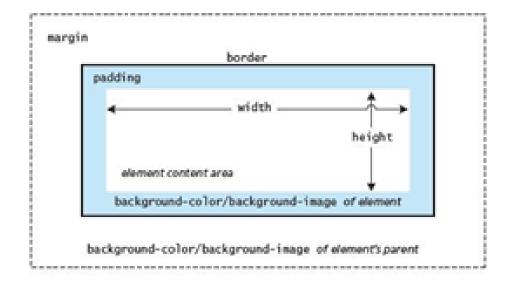
We'll begin with an overview of the components of an element box, then take on the box properties from the inside out: content dimensions, padding, borders, and margins.
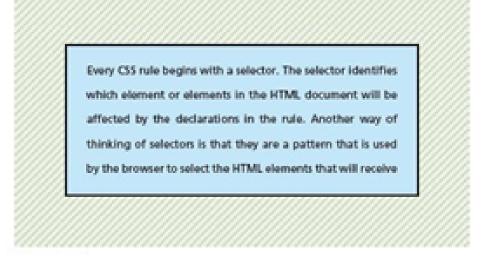
As you can see in the table that follows, every box has three properties you must be aware of

Property	Description
border	Even if you cannot see it, every box has a border. This separates the edge of one box from other surrounding boxes.
margin	The margin is the distance between the border of a box and the box next to it.
padding	This padding is the space between the content of the box and its border.

The Element Box

> Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule. Another way of thinking of selectors is that they are a pattern that is used by the browser to select the HTML elements that will receive
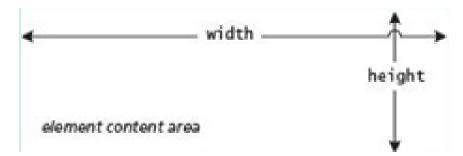
The parts of an element box according to the CSS box model.

§ **Content Area**

At the core of the element box is the content itself. The content area is indicated by text in a white box.

§ **Inner edges**

The edges of the content area are referred to as the inner edges of the element box. This is the box that gets sized when you apply *width* and *height* properties. Although the inner edges are made distinct by a color change, the edge of the content area would be invisible.

§ **Width**

The width property is a pretty simple property; it sets the width of an element.

PROPERTY	VALUE
width	\<length\> | \<percentage\> | auto initial value: auto

The *width* property accepts a length unit, which is indicated in the preceding table with the \<length\> notation. You've already seen the width property in use in your examples so far.

§ **Height**

Like the width property, the height property sets the amount of space between the top-inside padding edge and the

Like the width property, the height property sets the amount of space between the top-inside padding edge and the bottom-inside padding edge.

The **_height_** property works in the same way as the width property, but is less commonly used as most web pages are designed to scroll vertically, and setting a fi xed height causes some strange effects.

§ **Padding**

The padding is the area held between the content area and an optional border. In the diagram, the padding area is indicated by a blue color. Padding is optional.
The following table outlines the possible values for these properties.

PROPERTY	VALUE
padding	[<length> | <percentage>] {1,4}
padding-top padding-right padding-bottom padding-left	<length> | <percentage>

Like margin, **_border-width, border-style, and border-color_**, the **_padding_** property is a shorthand property, meaning that it is a simplified representation of the other **_padding_** properties, **_padding-top, padding-right, padding-bottom, and padding-left_**. In the preceding table, the square brackets are used to group the values. In this context, the **_padding_**property can accept either a length or a percentage value, and can have one to four space-separated values.

§ **Border**

The border is a line (or stylized line) that surrounds the element and its padding. Borders are also optional. The following table outlines the possible values for these properties.

PROPERTY	VALUE
border-top border-right border-bottom border-left	<border-width> || <border-style> || <color>border-right
border	<border-width> || <border-style> || <color>

The notation for the **_border-top, border-right, border-bottom, border-left_**, and **_border_** properties indicates that one to three values are possible; each value refers to a **_border-width_** value, a border-style value, and a **_border-color_** value.

§ **Margin**

The margin is an optional amount of space added on the outside of the border. Margins are always transparent, but

The margin is an optional amount of space added on the outside of the border. Margins are always transparent, but you can modify the background color of it. The following table outlines the possible values for these properties.

PROPERTY	VALUE
margin	[<length> | <percentage> | auto] {1,4}
margin-top margin-right margin-bottom margin-left	<length> | <percentage> | auto

The **margin** property is a shorthand property for the four individual **margin properties, margin-top, margin-right, margin-bottom**, and **margin-left**.

§ **Outer Edge**

The outside edges of the margin area make up the outer edges of the element box. This is the total area the element takes up on the page, and it includes the width of the content area plus the total amount of padding, border, and margins applied to the element. The outer edge in the diagram is indicated with a dotted line, but in real web pages, the edge of the margin is invisible.

§ **Minimum and Maximum Dimensions**

The **min-width, max-width, min-height**, and **max-height** properties define minimum and maximum boundaries when it is necessary to constrain a width or height from expanding or contracting past a certain point. In a variable width design, where you design content to adapt to multiple screen resolutions, it is sometimes helpful to define where you want the document to stop stretching or stop contracting.

§ min-width and min-height

The **min-width** property defines a lower-size constraint on an element. The available values for the **min-width** property are outlined in the following table.
The **min-width** and **min-height** properties define when an element should stop shrinking to fit the user's window or its content.

PROPERTY	VALUE
min-width min-height	<length> | <percentage> Initial value: 0

§ max-width and max-height

In contrast to the **min-width** property, the **max-width** and **max-height** properties are used to set an upper

constraint for width and height for elements.

PROPERTY	VALUE
max-width max-height	<length> | <percentage> | none initial value: none

The **max-width** and **max-height** properties allow you to defi ne a maximum length if the area available to the element becomes larger.

§ **Overflowing Content**

The CSS **overflow** property exists to manage content that is susceptible to dimensional constraints, where the content could possibly overflow the boundaries of those dimensional constraints. The following table outlines the **overflow** property and its possible values.

PROPERTY	VALUE
overflow	visible | hidden | scroll | auto initial value: visible

The two most common uses of the **overflow** property are to hide content when more content than space is available, or to apply scroll bars so that the extra content can be accessed. By default, the value of the overflow property is the **visible**.
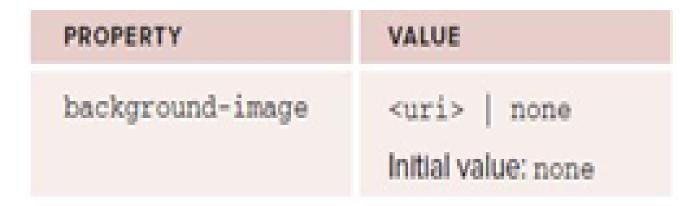
§ **Background Colors**

The background-color property is used to specify a solid background color. The following table shows the possible values for the background-color property.

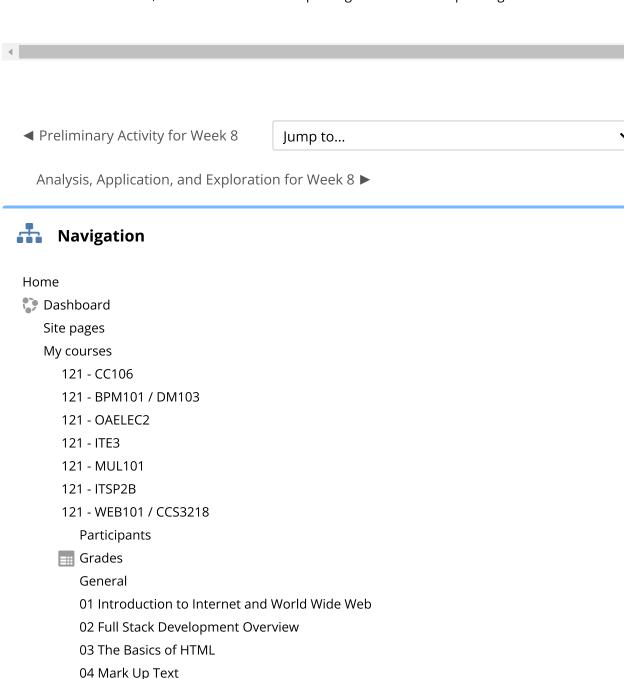| PROPERTY	VALUE
background-color	<color> | transparent initial value: transparent

The **background-color** property allows any of the color values supported by CSS, such as a color keyword, an RGB value, or a hexadecimal, or short hexadecimal value. It may also be given the **transparent** keyword, which indicates that no color should be used.

§ **Background Images**

As you probably guessed, the **background-image** property enables you to provide an image for the background. The following table outlines the values available for the **background-image** property.

PROPERTY	VALUE
background-image	<uri> | none Initial value: none

The ***background-image*** property allows you to reference a URL, which is indicated by the ***<uri>*** notation in the preceding table, or a keyword of none. When you specify a background image, by default the image tiles across the entire area available to it, which is the area encompassing the content and padding of the element being styled.

Jump to...

🔗 Navigation

Home
🔵 Dashboard
   Site pages
   My courses
      121 - CC106
      121 - BPM101 / DM103
      121 - OAELEC2
      121 - ITE3
      121 - MUL101
      121 - ITSP2B
      121 - WEB101 / CCS3218
      Participants
      📊 Grades
      General
      01 Introduction to Internet and World Wide Web
      02 Full Stack Development Overview
      03 The Basics of HTML
      04 Mark Up Text
      05 HTML Link, Image and Table
      06 - Preliminary Examination

ℹ️ Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for *free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission*.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

ℹ️ 2nd Semester Enrollment

Activities

📄 Assignments
💬 Forums
✅ Quizzes
📄 Resources

Bestlink College of the Philippines
College Department

Powered by eLearning Commons