

14.Cascading Style Sheets (CSS):

Cascading Style Sheets, often referred to as CSS, are files containing rules and definitions that you can create separately and then associate with your website's HTML documents. These rules serve as instructions for the visual presentation of your web content. Once you've established a style sheet, you link it to your HTML files, enabling you to centrally manage the appearance of multiple web pages by modifying a single source. This centralized approach not only fosters a consistent design across your site but also grants you greater control over the individual elements within it. CSS can be implemented in three ways: as an external text file separate from your HTML documents, as internal style code embedded within an HTML page, or as inline style code within specific HTML elements.

It's crucial to keep in mind that when employing multiple style directives, they follow a specific hierarchy, each building upon the previous ones while preserving their unique formatting. The order of precedence is as follows:

- 1. Inline Style**
- 2. Internal Style**
- 3. External Style Sheet**

14.2 - CSS Syntax Explained:

CSS syntax differs significantly from HTML. A CSS rule comprises two essential components: selectors and declarations. The selector specifies the HTML element to which the rule applies and is separated from the declarations by a space. Declarations, enclosed within curly brackets, dictate the formatting instructions and consist of a property and its corresponding value. Each declaration concludes with a semicolon. You can assign multiple declarations to a single selector by separating them with semicolons. For instance, the rule "h1 {font-weight: bold; color: maroon;}" sets all level 1 Headings to display in bold maroon text.

It's worth noting that declarations affect not only the designated element but also any nested elements within it. For instance, if you establish a rule that pertains to the <body> element, any elements contained within the <body> will inherit the same formatting properties.

14.3 - Crafting an External CSS Document:

To generate an external Cascading Style Sheet, commence by creating a new file in your text editor. Within this document, input your CSS rules, maintaining a single rule per line for ease of future modifications. After listing your rules, save the document as a text file, ensuring it bears the .css extension. We will delve into the specifics of crafting rules in the forthcoming chapter. Following the creation of your CSS file, the next step is to establish a connection between it and your HTML pages.

Step	Action	Example
Start	Open your text editor.	
Type	Enter your CSS rules.	h1 {color: maroon; font-style: italic;} h2 {color: yellow;}
Save	Save the document as a text file with a .css extension.	File name: teachstyles.css
Result	Generates a CSS file containing a rule that defines level 1 headings as maroon and italicized, and level 2 headings as yellow.	

14.4 - Establishing a Link to a Cascading Style Sheet:

Once you've crafted your CSS, the next step is to link it to each HTML page where you want to apply the formatting. Within the <head>...</head> tags of your HTML document, insert the following code snippet: <link rel="stylesheet" type="text/css" href="?">, replacing the "?" with the name of your style sheet.

Tag: <link>

Example: <head>
<link rel="stylesheet" type="text/css" href="teachstyles.css">
</head>

Result: Establishes a link between the HTML page and the style sheet.

14.5 - Incorporating Comments and Annotations into CSS:

Much like you can in your HTML documents, you have the ability to insert comments into your Cascading Style Sheets as a means of documentation or for providing notes to yourself or fellow designers who may review your code. It's important to note that these comments are solely visible within the CSS code and are not presented in the web browser.

To include comments in your CSS, commence your notes with a forward slash followed by an asterisk (`/*`), and conclude the comments with the reverse order—an asterisk followed by a forward slash (`*/`).

Start	<code>/*</code>	
End	<code>*/</code>	
Example	<code>h1 {color: #660033; font-style: italic;} /* Describes the logo color */</code>	
Result	The comments specifying the color of the logo will remain concealed.	

14.6 - Crafting an Internal Style Sheet:

Style sheets can also be internal, indicating that the rules are embedded within the HTML page itself, rather than residing in a separate text file. This approach can be advantageous if your website consists of a single, extensive page with numerous elements you want to control from a central location.

Internal style sheets are placed within the `<head>...</head>` tags. As the code is integrated into your HTML page, you must employ the `<style>...</style>` tags to encapsulate the CSS rules. No additional attributes are necessary. Simply format the rules in the same manner as you would for external style sheets.

Start tag	<code><style></code>	
End tag	<code></style></code>	
Attributes	<code>type="text/css"</code>	
Example	<code><head>
<style>
h1 {color: maroon; font-style: italic;}
h2 {color: yellow;}
</style>
</head></code>	
Result	Produces an internal CSS with a rule dictating that all level 1 headings should appear in maroon and italicized text, while level 2 headings should display in yellow.	

14.7 - IDs and Classes:

There are two distinct approaches for modifying the formatting of various elements in your HTML page, namely, IDs and Classes. The ID selector is employed when you intend to alter the style of a single, unique element, while the Class selector is used when you aim to modify a group of elements within your HTML document.

The ID selector utilizes the ID attribute of the HTML element and is represented by the hashtag symbol (#). For instance, if you have a heading with the ID attribute set to "heading1" in your HTML code, the associated style code would be designated as "#heading1". This directive exclusively impacts the element labeled as "heading1" while leaving all other headings unaffected. This approach proves valuable when you wish to emphasize a specific heading on your webpage.

You can establish classes within both internal and external style sheets when you need to apply distinct style rules to specific sets of tags. For example, you might want certain paragraphs to exhibit a particular font and color, while other paragraphs have a different appearance. To create a class in your CSS, begin by defining the class by specifying the tag you want to control, followed by a period and a name of your choice for the class. The name should be succeeded by declarations enclosed within curly brackets. After creating the class, assign the CLASS attribute to the relevant start tags in your HTML document, with the values corresponding to the class name.

It's imperative that the labels for both IDs and Classes match precisely. For instance, if your HTML document designates an ID as "id=heading1," your CSS document must reference it as "heading1." Any variation in capitalization, such as "Heading1," will result in code validation issues. In such cases, capitalization plays a pivotal role.

ID:

- CSS Definition:

```
#heading1 {font-family: "Times New Roman"; color: blue;}
```

- HTML Attribute: `id=`

Example:

```
<h3 id="heading1">This is my first Heading.</h3>
```

```
<h6>This is my second Heading.</h6>
```

Result:

[This is my First Heading.](#)

This is my second Heading.

Class:**- CSS Definition:**

.Introductions {font-family: georgia; color: red;}

.Speech {font-family: verdana; color: blue;}

or

p.Introductions {font-family: georgia; color: red;}

p.Speech {font-family: verdana; color: blue;}

- HTML Attribute: `class=`**Example:**

```
<p class="Introductions">Introduction text here.</p>
```

```
<p class="Speech">Speech text here.</p>
```

Result:

Introduction text here.

Speech text here.

14.8 - Applying Inline Styles:

Inline styling entails taking your CSS code and embedding it directly within the HTML tag of your code. To achieve this, you make use of the STYLE attribute within the specific tag you wish to modify. However, it's advisable to exercise restraint when employing inline styling, as it diminishes many of the benefits offered by style sheets.

Start Tag	The HTML tag you choose to style.
End Tag	The corresponding closing tag of the chosen HTML element.
Attribute	style= Enclosed in quotation marks along with the CSS code.
Example	<code><p style="font-weight: bold; color: green;"></p></code>
Result	The paragraph styled in this manner will exhibit bold text and appear in green.

Chapter 15:

Working with Text in CSS

15.1 - Enhancing Text (Making it Bold and Italic):

To apply bold formatting to text within a CSS style rule, utilize the FONT-WEIGHT property. You can employ the default setting of "bold," or you can specify the degree of boldness using a numerical value in increments of 100, with 100 representing the lightest and 900 representing the boldest.

When it comes to italicizing text in a CSS style rule, you can utilize the FONT-STYLE property. This property offers three possible values: "italic," "oblique," and "normal." "Italic" denotes the italicized variant of the assigned font. If the selected font lacks an italic version, you can opt for "oblique" to instruct the browser to attempt a slanted effect to mimic italics. In cases where an element has inherited italics from a previous element and you wish to eliminate the italics, you can use the "normal" value.

- Property for Bold Text: font-weight (Used for making text bold)
 - Value: "bold" or numerical values (e.g., 100, 200...900)
 - Example: `p {font-weight: 300;}`
 - Result: Establishes the CSS rule to apply a boldness level of 300 to paragraph text.
- Property for Italic Text: font-style (Used for italicizing text)
 - Value: "italic," or "normal"
 - Example: `p {font-style: italic;}`
 - Result: *Configures the CSS rule to italicize the text within paragraphs.*

15.2 - Text Decoration:

The TEXT-DECORATION property serves to add various text decorations, such as strike-through lines, underlines, and overstrike lines, to your text. Typically, this property is employed to remove underlines from hyperlinks. For instance, if a link appears within a sentence that is already underlined, it can stand out more by not having an additional underline. As a general practice, it's advisable not to underline text that isn't a hyperlink, as this can be confusing for some users.

- Property for Text Decoration: text-decoration

- Value: "underline," "line-through," "underline," or "none"

- Example: p {text-decoration: underline;}

- a {text-decoration: none;}

- Result: Establishes the CSS rule to underline all text within paragraphs and specifies that no underlines should be applied to any links.

15.3 - Adjusting Paragraph Indentation:

To create an indentation for the first line of a paragraph, you can utilize the TEXT-INDENT property. The value you specify can be expressed in various units, such as pixels (px), inches (in), millimeters (mm), centimeters (cm), points (pt), picas (pc), or x-height (ex). Among these options, pixels (px) are the most commonly used unit of measurement.

- Property for Indentation: text-indent

- Value: px, in, mm, cm, pt, pc, or ex

- Example: p {text-indent: 35px;}

- Result: Sets the CSS rule to create an indentation of 35 pixels for the first line of paragraphs.

15.4 - Text Transformation:

The TEXT-TRANSFORM property is employed to specify the letter case in which your text should appear. There are three available values: "uppercase," "lowercase," and "capitalize." "Uppercase" converts all letters to uppercase (similar to using the caps lock key), "lowercase" transforms all letters to lowercase, and "capitalize" instructs the rule to capitalize the first letter of each word.

- Property for Text Transformation: text-transform

- Value: "uppercase," "lowercase," or "capitalize"

- Example: p {text-transform: uppercase;}

- Result: CONFIGURES THE CSS RULE TO CONVERT ALL LETTERS WITHIN THE PARAGRAPHS TO UPPERCASE.

15.5 - Text Alignment:

To control the horizontal alignment of block-level text elements such as paragraphs, tables, and other elements with a blank line before and after them, you can utilize the TEXT-ALIGN property. The available alignment values include "left," "right," "center," and "justify."

- Property for Alignment: text-align
- Value: "left," "right," "center," or "justify"
- Example: h1 {text-align: center;}
- Result:

Configures the CSS rule to center-align all level 1 headings.

15.6 - Managing Fonts:

To manage fonts within a CSS style rule, apply the FONT-FAMILY property. When employing this property, it's essential to list multiple font options, as not all fonts are universally supported by all web browsers. The user's browser will attempt each font in the order listed. Multiple fonts should be separated by commas. For instance, listing "Arial, Helvetica, 'Times New Roman'" means the browser will first attempt to display Arial, then Helvetica, and finally 'Times New Roman' if the previous fonts are unavailable. When specifying font values for the FONT-FAMILY property, you only need to enclose multiple-word fonts in quotes, such as "Times New Roman."

- Property for Font Management: font-family
- Value: Font names
- Example: p {font-family: verdana, georgia;}
- Result: Sets the CSS rule to apply the verdana font first and then georgia to the text of paragraphs.

15.7 - Font Sizes:

The FONT-SIZE property allows you to create a CSS rule to control the font size. The value you assign can be expressed numerically in units such as pixels (px), inches (in), millimeters (mm), centimeters (cm), points (pt), picas (pc), x-height (ex), em (equal to 16px, the default browser font size), or as a percentage. The default font size in browsers is typically 16px. Using the "em" value is advisable, as it ensures that text is displayed correctly when users resize the webpage using the zoom tool.

- Property for Font Size: font-size

- Value: px, in, mm, cm, pt, pc, ex, em, or percentage
- Example: p {font-size: 1.5em;}
- Result: Sets the CSS rule to display all paragraph text with a font size of 1.5 em or 24px.

15.8 - Letter Spacing (Kerning):

Kerning refers to controlling the horizontal spacing between characters in text. To adjust the kerning of text, use the LETTER-SPACING property. The value can be expressed in points (pt), pixels (px), inches (in), millimeters (mm), centimeters (cm), picas (pc), x-height (ex), or em.

- Property for Kerning: letter-spacing
- Value: pt, px, cm, mm, in, pc, ex, or em
- Example: h1 {letter-spacing: 5pt;}
- Result: Sets the CSS rule to adjust the kerning of level 1 headings by 5 points.

15.9 - Line Spacing (Leading):

Leading pertains to controlling the vertical spacing between lines of text. You can manage leading using the LINE-HEIGHT property. The value is typically expressed as a multiple of the font's height. It can also be a percentage of the font or an absolute value in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em.

- Property for Leading: line-height
- Value: x.x (as a multiple of the font), x% (as a percentage of the font), pt, px, cm, mm, in, pc, ex, or em
- Example: p {line-height: 2.0;}
- Result: Sets the CSS rule to apply a leading of two times the font's height to paragraphs.

15.10 - Text Color:

To control the color of text, you can use the COLOR property. You can specify the color using a color name or a hexadecimal color value. Additionally, the COLOR property can be applied to other elements like tables, borders, and lines.

- Property for Text Color: color

- Value: Color name or hexadecimal color
- Example: `h1 {color: maroon;}`
- Result: Configures the CSS rule to apply a maroon font color to all level 1 headings.

15.11 - Margin Control:

For managing margins of elements within your HTML pages, the MARGIN property in your style sheets can be used. You can adjust all four margins of an element using the MARGIN property by specifying the margin you want to change (left, right, top, or bottom) preceded by a dash (-). The value can be in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em.

- Property for Margin Control: margin-x (Where "x" is the margin to change: left, right, top, or bottom)
- Value: pt, px, cm, mm, in, pc, ex, or em
- Example: `h2 {margin-left: 30px;}`
- Result

: Sets the CSS rule to apply a left margin of 30 pixels to all level 2 headings.

15.12 - Padding:

To add padding (blank spaces) around the content of an element, the PADDING property can be used. You can specify the padding value in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em.

- Property for Padding: padding
- Value: pt, px, cm, mm, in, pc, ex, or em
- Example: `h1 {padding: 30px;}`
- Result: Sets the CSS rule to apply padding of 30 pixels around all level 1 headings.

15.13 - Borders:

To emphasize elements on a webpage, you can use the BORDER property in your CSS rule. You can set the thickness of the border by using values like "thin," "medium," or "thick." Additionally, you must specify a border style such as "solid," "double," "groove," "ridge," "inset," "outset," "dotted," or "dashed" for web browsers to display the border. You can also add padding to elements surrounded by a border for better readability. Remember to separate the BORDER property and its values from the PADDING property with a semicolon. If you want to give your borders rounded corners, you can use the BORDER-RADIUS property, specifying values in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em.

- Property for Borders: border

- Value: Thickness ("thin," "medium," or "thick"), Border style ("solid," "double," "groove," "ridge," "inset," "outset," "dotted," or "dashed"), and optional color value

- Example: `h1 {border: solid navy; padding: 10px;}`

- Result: Sets the CSS rule to apply a solid navy border around all level 1 headings with a 10px padding.

- Property for Rounded Corners: border-radius

- Value: pt, px, cm, mm, in, pc, ex, or em

- Example: `h1 {border: solid navy; border-radius: 25px;}`

- Result: Sets the CSS rule to apply a solid navy border with 25px rounded corners to all level 1 headings.

15.14 - Styling Links:

To assign different colors and decorations to links in different states on your HTML pages, you can use CSS. By default, link colors change after a visitor clicks on them, helping users keep track of their visited links. To modify link properties, use selectors like ":link" (for unvisited links), ":visited" (for visited links), ":hover" (when the mouse pointer is over the link), and ":active" (when a link is clicked). You can also remove the default underline beneath links by assigning "text-decoration: none;".

- Property for Link Styling: `a:? {color:;}`

- Value: Link state (link, visited, hover, or active) and color value (to set the color)

- Example: `a:link {color: navy; text-decoration: none;}`

- Result: Sets the CSS rule to make all unvisited links navy with no underline.

15.15 - Number and Bullet Styles:

To change the appearance of ordered or unordered lists, you can use the LIST-STYLE-TYPE property. For ordered lists, available value choices include "decimal" (default), "lower-alpha," "upper-alpha," "lower-roman," and "upper-roman." For unordered lists, options are "disc" (default), "circle," or "square." You can also use an image as a list marker by employing the LIST-STYLE-IMAGE property and specifying the image file name.

- Property for List Styling (Ordered Lists): list-style-type

- Value: "decimal," "lower-alpha," "upper-alpha," "lower-roman," or "upper-roman"

- Example: `ul {list-style-type: circle;}`

- Result: Sets the CSS rule to use circles as bullets for all unordered lists.

- Property for List Marker Image: list-style-image

- Value: `url(sample.jpg)`

- Example: `ul {list-style-image: url(sample.jpg);}`

- Result: Sets all unordered lists to use the specified image as the list marker.

15.16 - Sizing Elements:

For precise control over the sizing of specific elements on your pages, you can employ the WIDTH and HEIGHT properties in your style sheets. You can specify the value in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), em, or as a percentage of the page size.

- Properties for Sizing Elements: width, height

- Value: pt, px, cm, mm, in, pc, ex, em, or percentage

- Example: `img {width: 167px; height: 145px;}`

- Result: Resizes all image tags to dimensions of 167 x 145 pixels.

15.17 - Text Wrapping:

The WHITE-SPACE property is employed to ensure text wraps within the designated element. By default, text automatically wraps inside elements. The values for the WHITE-SPACE property are "normal" (default) and "nowrap," which disables text wrapping inside the element. You typically only

need to use the WHITE-SPACE property when you want to turn off text wrapping, as the default behavior is to wrap text inside HTML elements.

- Property for Text Wrapping: white-space

- Value: "normal" or "nowrap"

- Example: `div {white-space: nowrap;}`

- Result: Sets the CSS rule to disable text wrapping for all div elements.

15.18 - Text Shadowing:

To add more visual impact to your text, you can utilize the TEXT-SHADOW property. Specify values for horizontal shadow, vertical shadow, blur distance, and shadow color. The value can be expressed in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em. The shadow color can be specified using a color name or hexadecimal value.

- Property for Text Shadowing: text-shadow

- Value: pt, px, cm, mm, in, pc, ex, em, and a color name or hexadecimal

- Example: `h1 {text-shadow: 5pt; color: blue;}`

- Result: Sets the CSS rule to display all level 1 headings with a 5pt blue shadow.

Chapter 16:

Creating Backgrounds in CSS

16.1 - Background Colors:

To define the background color on your webpage, you can utilize the BACKGROUND-COLOR property. Similar to other color selection properties, you can specify the color using either a hexadecimal value or a color name. The background color for your entire page is typically set using the body selector.

- Property for Background Color: background-color
- Value: Color name or hexadecimal value
- Example: `body {background-color: #ADD8E6;}`
- Result: Specifies the CSS rule to establish a light blue background color for your webpage.

16.2 - Background Images:

When you want to set an image as the background of your webpage, you can employ the BACKGROUND-IMAGE property. This property can be used to set the background for your entire page using the body selector or for specific elements like tables or paragraphs. By default, background images repeat to fill the available space, but you can control this behavior using the BACKGROUND-REPEAT property with values such as repeat-x (horizontal repetition), repeat-y (vertical repetition), or no-repeat. It's crucial to choose an image that complements your webpage's content and doesn't distract from it.

- Property for Background Images: background-image
- Value: URL("X"), where "X" is the link to your image file.
- Supporting Property for Background Image Repetition: background-repeat
- Value: repeat-x, repeat-y, or no-repeat.
- Example: `body {background-image: url("background.jpg"); background-repeat: repeat-y;}`
- Result: Applies the CSS rule to set your background as the chosen image and repeat it vertically only.

A website for downloading background images for webpages:

<https://www.freepik.com/free-photos-vectors/website-background>

16.3 - Fixed Background Images:

When using an image as a background, you can also control whether it scrolls along with other elements on your webpage. This can be managed using the BACKGROUND-ATTACHMENT property with values like "scroll" (the default) or "fixed." In "scroll" mode, the background image moves as users navigate your webpage, while "fixed" mode keeps the background image stationary, regardless of the part of the page your users are viewing.

- Property for Background Attachment: background-attachment
- Value: scroll or fixed
- Example: `body {background-image: url("background.jpg"); background-attachment: fixed;}`
- Result: Sets the CSS rule to use the selected image as the background with a fixed position on the page.

Chapter 17:

Images in CSS

17.1 - Adjusting Image Opacity:

You can control the transparency of an image or element using CSS by employing the `OPACITY` property and specifying a value ranging from 0.0 to 1.0. Lower values make the image or element more transparent. To adjust the opacity of specific images, you need to assign IDs to your images, as discussed in Chapter 14.

- Property for Image Opacity: `opacity`
- Value: A range from 0.0 to 1.0, where lower values indicate increased transparency.
- Example: `img {opacity: 0.5;}`
- Result: Defines the CSS rule to display all images with 50% opacity.

17.2 - Floating Images:

When you want to position an image to the left or right of another element, you can utilize the `FLOAT` property. The `FLOAT` property offers two values: "left" and "right." The floated image will move as far left or right as possible, allowing any subsequent elements to wrap around it. For instance, floating an image to the right will cause text in the following element to wrap around the left side of the image, without affecting elements that appear before the floated image.

- Property for Floating Images: `float`
- Value: left or right
- Example: `img {float: left;}`
- Result: Establishes the CSS rule for all images to align on the left side of the page, with any subsequent elements wrapping around the image's right side.

17.3 - Centring Images:

To position an image in the center of its containing element using CSS, you can use a combination of CSS properties. Here's a CSS code snippet to achieve this:

```
``css
.center-image {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

In this code:

1. `.center-image`` is a class selector that you can apply to the container element that holds the image. You can replace it with your own class name or use any other selector that matches your HTML structure.
2. ``display: flex;`` is used to create a flex container. This allows you to easily center the image both horizontally and vertically within its parent container.
3. ``justify-content: center;`` centers the content (in this case, the image) horizontally within the flex container.
4. ``align-items: center;`` centers the content vertically within the flex container.

Now, when you apply the `.center-image`` class to a container element that wraps your image, the image will be centered both horizontally and vertically within that container. Here's an example of how you would use this in HTML:

```
``html
<div class="center-image">
  
</div>
```

This HTML structure will center the image within its parent `<div>`` element.