

Introduction to Basic Technology Web

This course studies the basic concepts of technologies that can be used in developing a functional website. The basic technology for website creation that will be studied includes **HTML, CSS, Javascript, and other latest technologies**. Furthermore, students are also expected to be able to develop a website properly and correctly, namely by applying the technologies that have been learned both in the classroom and hands on.



Calandra Alencia Haryani,S.E.,S.SI.,M.T.I.,MCE

I earned Bachelor's Degree in Accounting with an Auditing concentration, Bachelor's Degree in Information systems with Project Management from Universitas Pelita Harapan, and a Master's Degree in Information Technology with a Business Data Analytics major from Universitas Indonesia. You can reach me at:

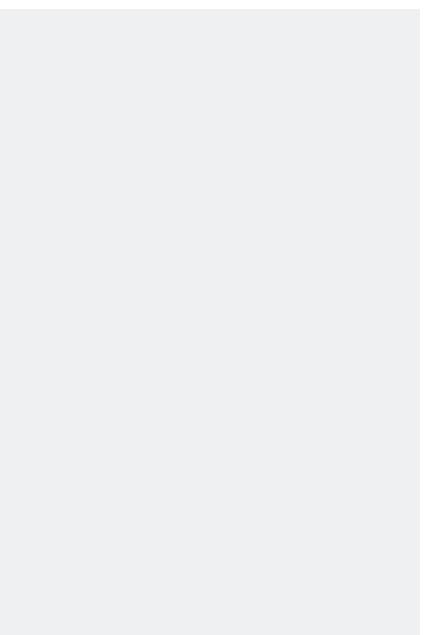
-  calandra.haryani@uph.edu
-  <https://wa.link/imrk9r>
-  <https://www.linkedin.com/in/calandraalencia/>

Objective.



- Describe the evolution of style
- List advantages of using Cascading Style Sheets
- Configure background and text color on web pages
- Create style sheets
- Apply inline styles
- Use embedded style sheets
- Use external style sheets
- Configure element, class, id, and descendant selectors
- Utilize the “cascade” in CSS
- Validate CSS

CSS



As HTML5 is introduced, examples and exercises encourage students to create sample pages and gain useful experience. Students use a variety of structural, grouping, and text-level HTML elements to create web pages with hyperlinks

WHY CSS

- HTML is designed to describe a web content
- As it develops, more tags to style content like `` and `<center>` are added
- These tags greatly complicates web development, making it very bloated and complex (“nightmare for web developers”)
- New standard is created to specifically style web content: Cascading Style Sheet (**CSS**), **separate content and styling**

— Most HTML tags to style contents are considered **deprecated**

- CSS can control layout of **multiple web pages at once**
- A webpage or website can have multiple different style sheets (themes), can let user choose
(example: https://www.w3schools.com/css/demo_default.htm)

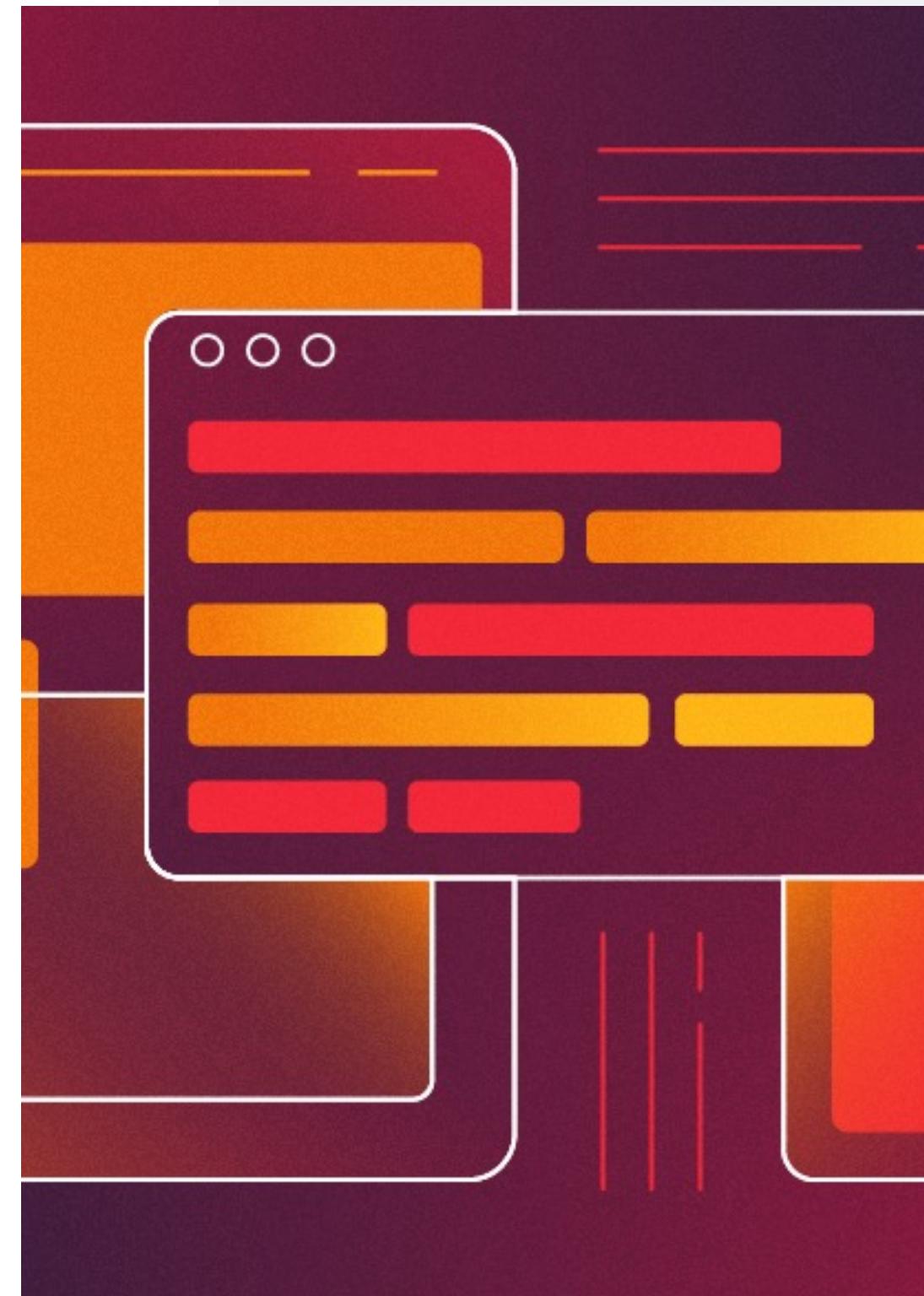
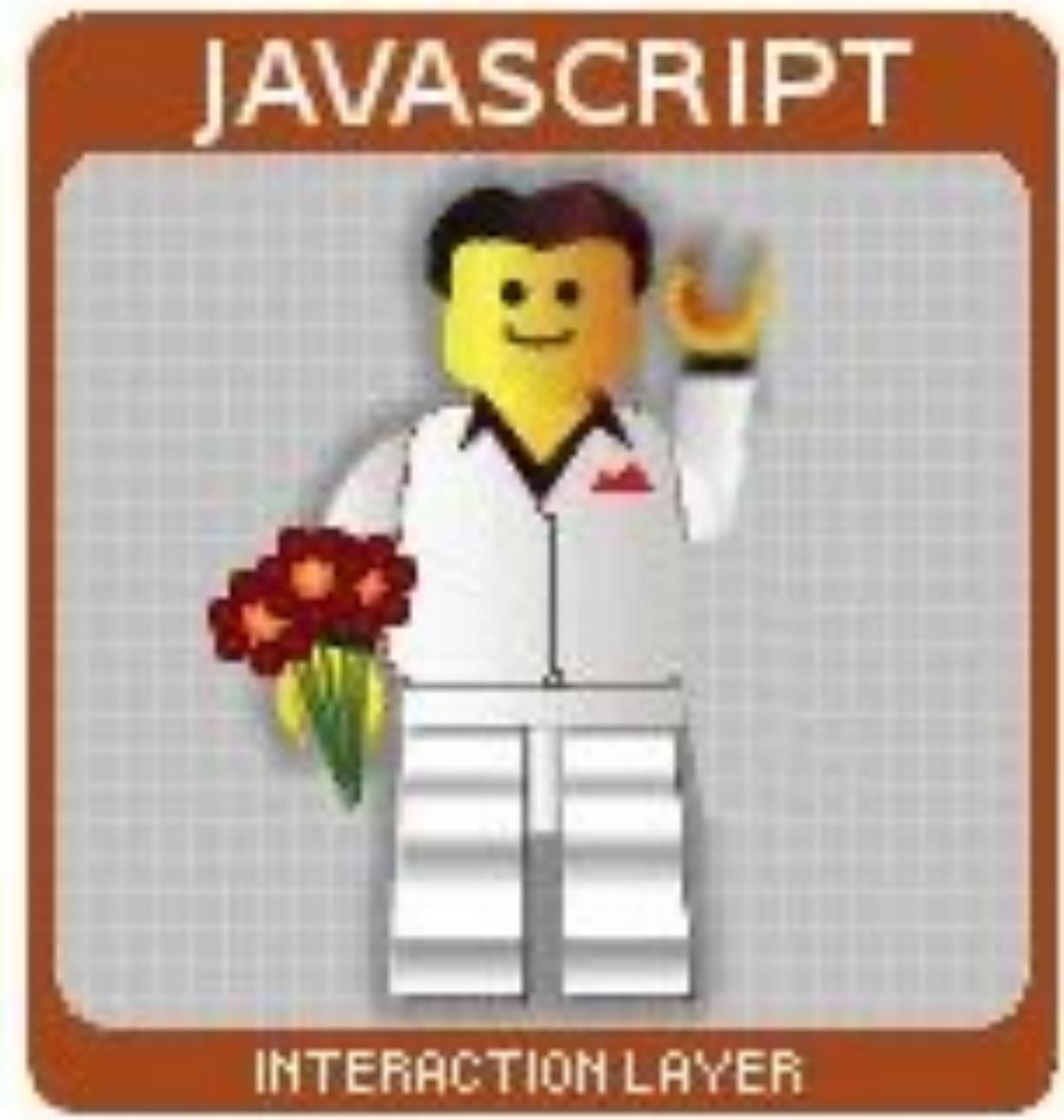


Foto Ini oleh Penulis tidak diketahui dilisensikan di bawah [CC BY-SA](#).

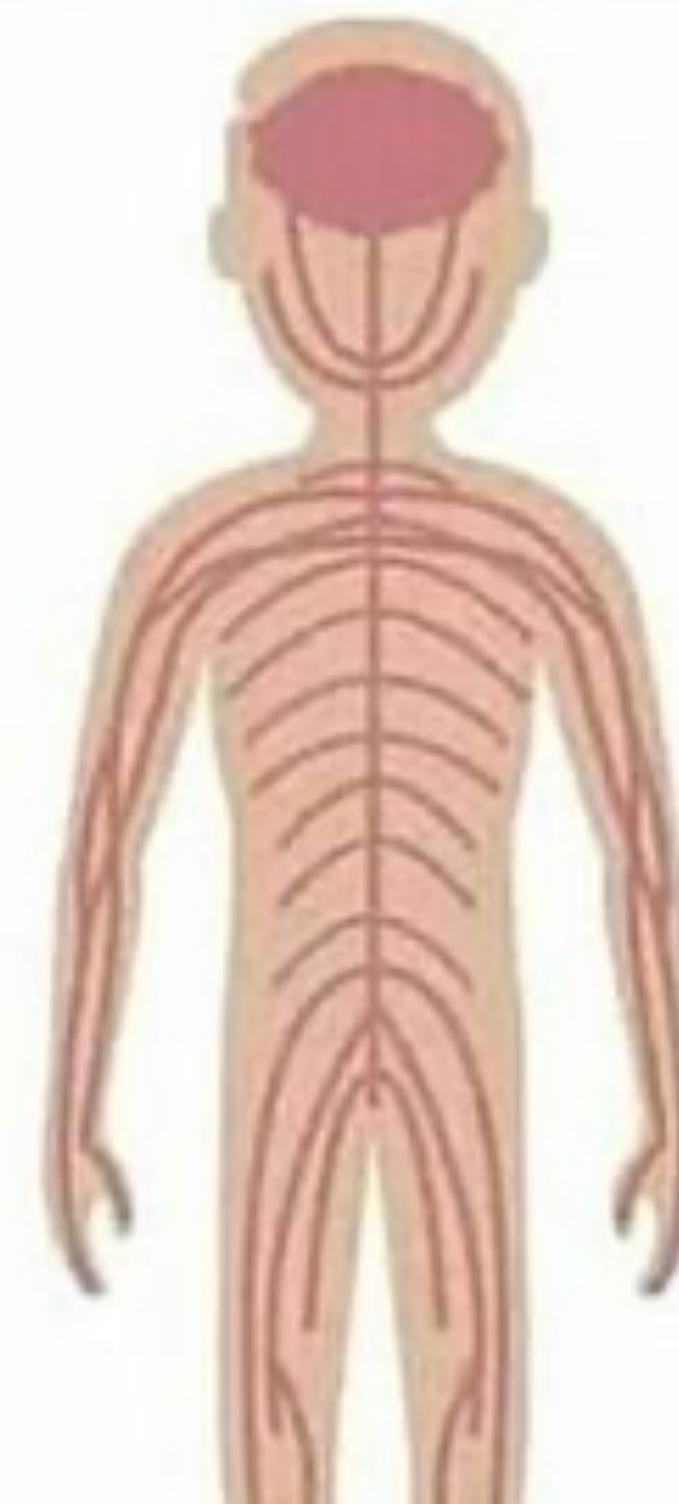


[Foto Ini](#) oleh Penulis tidak diketahui dilisensikan di bawah [CC BY-NC](#).

HTML



JS



CSS

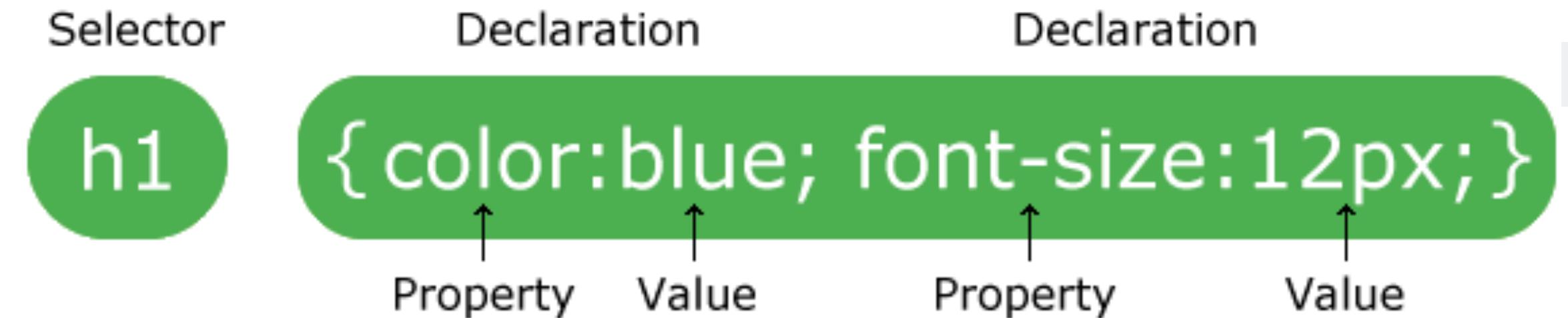


WHAT IS CSS

- Cascading Style Sheet, used to define how to style a website (positioning, colors, font size, alignment, margins, etc)
- Most recent version is CSS 2.1
- Can be implemented in a website (HTML document) in three different ways:
 - **Inline CSS**: embedded directly to HTML tags by using **style** attribute. Example: `<body style="..>`
 - **Internal CSS**: included in `<style>` element inside `<head>` section of HTML document to be styled
 - **External CSS**: Exists in a separate file with .css extension outside of HTML document, linked to with a `<link>` element inside `<head>` section of HTML document
- If multiple style sheets exist, style sheet in higher priority overrides the lower:
 1. Inline
 2. External and internal style sheets
 3. Browser default

```
    <li><a href="index.html">Home</a>
    <li><a href="home-events.html">Events</a>
    <li><a href="multi-col-menu.html">Multi Col</a>
    <li class="has-children"> <a href="#">More Options</a>
        <ul>
            <li><a href="#">Tall Buttons</a>
            <li><a href="#">Image Links</a>
            <li class="active"> <a href="#">Variables</a>
        </ul>
    </li>
    <li class="has-children"> <a href="#">More Options</a>
        <ul>
            <li><a href="#">Variables</a>
        </ul>
    </li>
</ul>
```

CSS SYNTAX



- All three CSS “versions” uses similar syntax and components
- CSS rule set comprises of **selector** and **declaration block**; each declaration comprises of **property:value pair**, declarations separated by semicolons (;
- Declaration blocks surrounded by curly braces { }
- **Selector**: select HTML element to be formatted (not necessary in inline CSS)
- **Property**: style component to be formatted
- **Value**: value of style component to be formatted
- Property and value separated by colon (:)

Inline CSS - Example

- Add style attribute to the HTML element to be formatted
 - The name of said attribute is style, and value of said attribute is a declaration block (property:value pair)
 - Can add more than one declaration in a style attribute, use semicolon to separate declarations
 - No need to declare selector since embedded directly to a specified HTML element
 - Curly braces not used to surround declaration block, instead use quotes (" ") to follow HTML attribute syntax
- Example (will format all elements inside <body> with red colored text):
 - <body style="color:Red;">
 - <h1>This is a heading</h1>
 - <p>This is a text</p>
 - </body>

```
ht:100%}body{margin:0;pa  
, sans-serif; font-size:1em;  
-weight:600;color:#404040;  
{line-height:180%}a{color:  
:0 auto; width:1200px}for  
}form .submit{margin-top:  
form select,form textarea  
;border:0;background:#f  
dius:.50em; margin:1em 0  
,0,0,0.05);border:solid  
t;-webkit-transition:all  
-transition:all .35s eas  
outline:0}form input.text  
0 0 2px 1px #e0e0e0;back  
e-placeholder{color:#555  
!important}form : -moz-p  
color:#555 !important}fo
```

Foto Ini oleh Penulis tidak diketahui dilisensikan di bawah CC BY-SA-NC.

Internal CSS - Example

- Insert CSS inside <style> tag
 - <style> must be placed inside <head>
 - Use complete CSS Syntax (selector + declaration block)
- Example:
- <head>
 - <style>
 - body {background-color:yellow;}
 - p {color:red;}
 - </style>
 - </head>

```
ht:100%}body{margin:0;pa  
, sans-serif;font-size:1em  
-weight:600;color:#404040  
{line-height:180%}a{color:  
:0 auto;width:1200px}for  
}form .submit{margin-top:  
orm select,form textarea  
;border:0;background:#f  
dius:.50em;margin:1em 0  
,0,0,0.05);border:solid  
t;-webkit-transition:all  
-transition:all .35s eas  
utline:0}form input.text  
0 0 2px 1px #e0e0e0;back  
e-placeholder{color:#555  
!important}form : -moz-p  
color:#555 !important}fo
```

Foto Ini oleh Penulis tidak diketahui dilisensikan di bawah CC BY-SA-NC.

EXTERNAL CSS - Example

- Create a separate .css file, put all CSS styling syntaxes on said file
 - CSS file should not contain any HTML element
- Link to said file using `<link>` element, place `<link>` inside `<head>`
- Syntax: `<link rel="stylesheet" href="filename">`
- Example: main.html file:
- `<head>`
- `<link rel="stylesheet" href="style.css">`
- `</head>`

```
ht:100%}body{margin:0;pa  
, sans-serif; font-size:1em;  
-weight:600;color:#404040;  
{line-height:180%}a{color:  
:0 auto; width:1200px}for  
}form .submit{margin-top:  
orm select,form textarea  
;border:0;background:#f  
dius:.50em; margin:1em 0  
,0,0,0.05);border:solid  
t;-webkit-transition:all  
-transition:all .35s eas  
utline:0}form input.text  
0 0 2px 1px #e0e0e0;back  
e-placeholder{color:#555  
!important}form : -moz-p  
color:#555 !important}fo
```

CSS COMMENTS

- Like in HTML or many other programming languages, can add comments to a CSS code
- Use /* ... */ symbols to add comment to CSS code
 - Anything inside the /* ... */ symbols treated as comment (not run as part of code)
 - Comment can be put anywhere in the code

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: red; /* Set text color to red */
}
</style>
</head>
<body>
```

CSS Sample Usage – Colors

Basic

- Colors in CSS can be specified using predefined color names, RGB, HEX, HSL, RGBA, or HSLA values
- There are 140 standard color names, supported by all browsers: https://www.w3schools.com/colors/colors_names.asp
- Components that can be colored:
 - Background color of HTML element (using **background-color** property)
 - Text color (using **color** property)
 - Border color (using **border** property)
- Example:
 - `<p style="color:red; background-color:black;">This is a text.</p>`

CSS Colors – RGB & RGBA

- Can specify a color by using RGB value (combination of red, green and blue)
 - Syntax: `rgb(red,green,blue)`
 - Value of red, green blue ranges between 0-255, defines intensity of said color
 - Example: `rgb(255,0,0)` gives red color (red=255,green=0,blue=0)
 - Example: `rgb(255,255,255)` gives white color
 - Example CSS syntax: `<p style="color:rgb(255,60,5);>This is a text.</p>`
- RGBA is rgb with added alpha/transparency level
 - Syntax: `rgba(red, green, blue, alpha)`
 - Alpha ranges from 0.0 (fully transparent) to 1.0 (not transparent at all)
 - Example CSS syntax: `<p style="color:rgba(255,60,5,0.5);>This is a text.</p>`
- RGB&A Color mixer: https://www.w3schools.com/css/css_colors_rgb.asp

CSS Colors - HEX

- Can specify color using hexadecimal value in form: **#rrggb**
- rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (=decimal 0-255)
 - Example: #ff0000 = red (rr=ff(255), gg=00, bb=00)
- Example CSS Syntax: <p style="color:#FF0000;">This is a text.</p>
- HEX Color Mixer: https://www.w3schools.com/css/css_colors_hex.asp

CSS Colors - HSL & HSLA

- Can specify color using HSL (hue, saturation, and lightness)
 - Syntax: `hsl(hue, saturation, lightness)`
 - Hue: degree on color wheel (from 0 to 360 - 0 is red, 120 is green, 240 is blue)
 - Saturation: from 0% (shade of gray) to 100% (full color)
 - Lightness: from 0% (black) to 100% (white)
 - Example: `hsl(0, 100%, 50%)` = red color
 - CSS Syntax: `<p style="color:hsl(0,100%,50%);>This is a text.</p>`
- HSLA is HSL with added transparency/alpha element from 0.0 (fully transparent) to 1.0 (not transparent at all)
 - Example CSS syntax: `<p style="color:hsla(0,100%,50%,0.5);>This is a text.</p>`
- HSL & HSLA Color Mixer: https://www.w3schools.com/css/css_colors_hsl.asp

CSS Selector

- Select the specific HTML elements to be styled
 - Several different types
 - **Element selector:** All HTML elements with the same name will be styled
 - Example: `p {background-color: green; color: Red;}` (style all `<p>` elements)
 - **id selector:** Select only HTML element with a specific id, defined in said element's id attribute
 - id attribute is unique within a page (cannot have same id value in a page)
 - To select element with specific id, write hash (#) character, followed by id of element
 - Example:
 - HTML: `<p id="para1">`
 - CSS: `#para1 {color: Red;}` (only element with id para1 will be styled)

CSS Selector

- **Class selector:** HTML elements with the same class will be styled, defined in said element's class attribute
 - Unlike id, class is not unique, multiple HTML elements can have same class
 - To select element with specific class, write **dot (.)** character, followed by class of element
 - Can also select only specific HTML elements that should be affected by class selector
 - Example:
 - HTML: <p class="redpara">
 - CSS: p.redpara {color:Red;} (only <p> element with class redpara will be styled)
- **Universal selector:** Use **star (*)** to style all HTML elements in a page
 - Example: * {color:Red;} (all HTML elements will be styled with color red)
- **Grouping selector:** Select multiple HTML elements to be styled with the same style
 - Separate each selector with a **comma (,)**
 - Example: p,h1,h2 {color:Red;} (style p,h1,h2 elements with color Red)

CSS Backgrounds

- There are several CSS properties that can be used to style background in CSS:
 - **background-color**: select a specific color, can use colorname, RGB, HEX or HSL. Example: `style="background-color:green;"`
 - **background-image**: Use an image as a background. Value is `url()` with the image's URL inside bracket.
Example: `style="background-image:url('photo.jpg')"`
 - By default, image is repeated so it covers the entire element; To adjust how the image repeats, use **background-repeat** property, with following values:
 - **repeat**: default value, repeats both horizontally (x-direction) and vertically (y-direction)
 - **repeat-x/repeat-y**: only repeats in x/y direction
 - **no-repeat**: background does not repeat
 - Example: `p {background-image:url('photo.jpg'); background-repeat:repeat-y;}`
 - Web design tip: when using background image, use image that does not disturb text

CSS Backgrounds

- Use **background-position** property to adjust background image position
 - Default value is top left; can adjust to be positioned in **top/center/bottom, left/center/right**
 - Can use specific position using **xpos ypos** (pixel values or units, 0 0 is top left) or **x% y%** (percentages; 0% 0% is top left, 100% 100% is bottom right)
 - Example: `background-image:url("photo.jpg"); background-position:top left;`
- **background-attachment** property specifies whether background image should **scroll** (follow along as page is scrolled) or **fixed** (stay in specified position)
 - Example: `background-image:url("photo.jpg"); background-attachment:fixed;`
- To shorten the code, can use **background** to specify all five background properties at once
 - Order: **background-color** **background-image** **background-repeat** **background-attachment** **background-position**
 - Does not matter if some properties are missing **if other ones are in order**
 - Example: `background #ff0000 url('photo.jpg') top right`

CSS Units

- Many CSS properties uses length values (e.g. background-position)
 - CSS has several units to express length (number/value followed by unit)
 - There are two types of length units:
 - **Absolute**: fixed units, length expressed will appear exactly that size
 - Not recommended for use on screen because screen sizes vary, can be used if output medium is known (e.g. print)
 - Examples: cm (centimetres), mm (millimetres), in (inches), px (pixels)
 - **Relative**: specify length relative to another length property
 - Examples: em (relative to font size), % (relative to parent element), vw & vh (percentage relative to width/height of viewport (browser window size))
 - Ref: https://www.w3schools.com/css/css_units.asp

References.

Terry Felke, Morris, "Web Development and Design Foundation with HTML 5", Pearson, 10th Edition:2020