CS3201: Computer Network Technology TY Div A n B AY 2020-21

Study Material for Section-II-Part-I- HTML n CSS

(**Resource:** www.w3schools.com)

Learning HTML and CSS

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

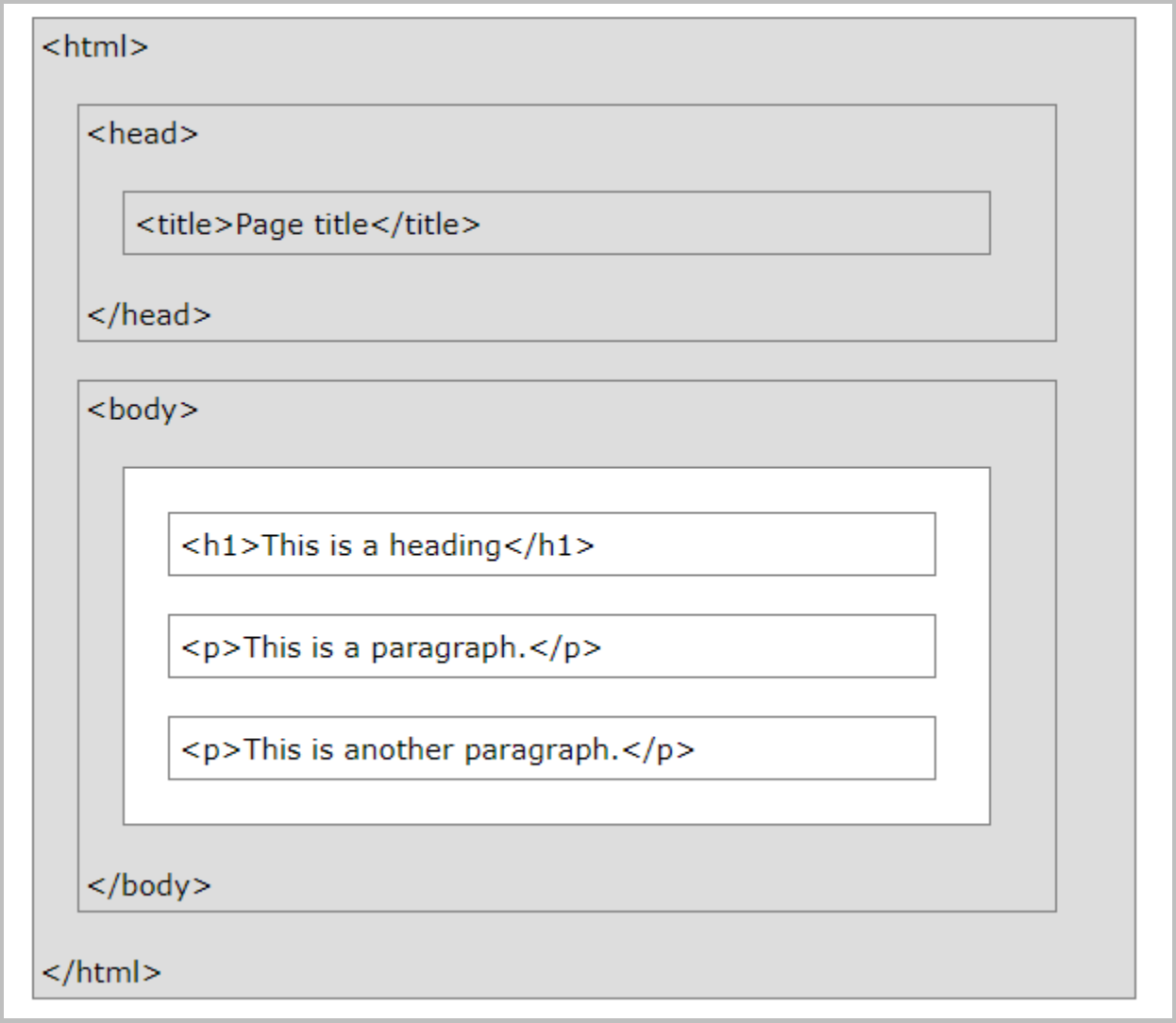
A Simple HTML Document

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph



What is an HTML Element?

An HTML element is defined by a start tag, some content, and an end tag:

<tagname>Content goes here...</tagname>

The HTML **element** is everything from the start tag to the end tag:

<h1>My First Heading</h1>
My first paragraph.

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	
 	none	*none*

HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+

1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

Editor – Notepad

HTML Links

HTML links are defined with the `<a>` tag:

Example

```
<a href="https://www.vit.edu">This is a link</a>
```

HTML Images

HTML images are defined with the `<img>` tag.

The source file (`src`), alternative text (`alt`), `width`, and `height` are provided as attributes:

Example

```
<img src="mld.jpg" alt="W3Schools.com" width="104" height="142">
```

Empty HTML Elements

HTML elements with no content are called empty elements.

The `<br>` tag defines a line break, and is an empty element without a closing tag:

Example

```
<p>This is a <br> paragraph with a line break.</p>
```

HTML is Not Case Sensitive

HTML tags are not case sensitive: `<P>` means the same as `<p>`.

What we know

1. HTML Document Structure
2. HTML Elements

Tag	Added	Short description
THE ROOT ELEMENT		
<html>		Contains the whole document
DOCUMENT METADATA		
<head>		Defines the document's header block
<title>		Specifies the document's title
<base>		Sets base URI to solve relative URIs
<link>		Gives relational information for documents
<meta>		Provides information for the document
<style>		Contains presentational attributes

SECTIONS

<body>	Contains all renderable elements
<article>	Contains distributable content
<section>	Defines a section of the document
<nav>	Defines a navigation section
<aside>	Holds content only slightly related
<h1>	Inserts a level 1 heading
<h2>	Inserts a level 2 heading
<h3>	Inserts a level 3 heading
<h4>	Inserts a level 4 heading
<h5>	Inserts a level 5 heading
<h6>	Inserts a level 6 heading
<hgroup>	Groups consecutive headings
<header>	Contains the header of a section
<footer>	Contains the footer of a section
<address>	Provides contact information

GROUPING CONTENT

<p>	Inserts a paragraph
<hr>	Draws a horizontal line or rule
<pre>	Defines a block of preformatted text
<blockquote>	Block level quotation
	Inserts an ordered list
	Inserts an unordered list
<menu>	Inserts a toolbar menu
	Defines a list item
<dl>	Inserts a definitions list
<dt>	Inserts a term in a list
<dd>	Provides descriptions in a list
<figure>	Marks its content as a reference
<figcaption>	Provides a caption for a figure
<main>	Acts as a main container for elements
	Defines a block of content

TEXT-LEVEL SEMANTICS

<a>	Inserts links or bookmarks
	Indicates emphasis
	Indicates strong emphasis
<small>	Renders text in "small" font
<s>	Content no longer accurate or relevant
<cite>	Inserts a citation or reference
<q>	Inserts an inline quotation
<dfn>	Provides a definition for a term
<abbr>	Explains abbreviations
<ruby>	Inserts ruby annotated text
<rt>	Provides a ruby annotation
<rp>	Makes text to be ignored in ruby
<data>	Provides a machine-readable version
<time>	Represents a date and/or time
<code>	Represents computer code

<var>	Indicates an instance of a variable
<samp>	Contains a program's sample output
<kbd>	Represents text entered by users
<sub>	Defines subscript text
<sup>	Defines superscript text
<i>	Renders italic text
<b>	Text in bold style
<u>	Represents non-textual annotations
<mark>	Marks text in another document
<bdi>	Isolates for bidirectional formatting
<bdo>	Overrides the bidirectional algorithm
	Assings attributes to text (inline)
 	Forces a line break
<wbr>	Represents a line break opportunity

EDITS

<ins>	Indicates inserted text

<del>	Indicates deleted text

EMBEDDED CONTENT

<picture>	Inserts a multi-source image
<source>	Specifies alternative media resources
<img>	Inserts an image
<iframe>	Inserts a frame inside a document
<embed>	Integrates external applications
<object>	Runs external applications
<param>	Sets a parameter for an object
<video>	Inserts videos in the document
<audio>	Inserts audio files in the document
<track>	Provides text tracks for a video
<map>	Defines a client-side image map
<area>	Defines sectors for image maps

TABULAR DATA

<table>	Inserts a table
<caption>	Provides a caption for a table

<colgroup>	Groups columns in a table
<col>	Sets attributes for a table's columns
<tbody>	Defines the body of a table
<thead>	Defines the header of a table
<tfoot>	Defines the footer of a table
<tr>	Inserts a row in a table
<td>	Inserts a regular cell in a table
<th>	Inserts a header cell in a table

FORMS

<form>	Inserts a form
<label>	Sets a label for a control
<input>	Displays an input control
<button>	Creates a button control
<select>	Creates a select control
<datalist>	Provides suggestions for input fields
<optgroup>	Groups options in a select control

<option>	Inserts an option in a select control
<textarea>	Creates a multiline text input
<output>	Shows the output of a process
<progress>	Shows a task's completion progress
<meter>	Represents a measurement
<fieldset>	Groups controls in a form
<legend>	Assigns a caption for a fieldset

INTERACTIVE ELEMENTS

<details>	Provides collapsable information
<summary>	Provides a summary for a details element
<dialog>	Inserts a dialog box

SCRIPTING

<script>	Contains scripts
<noscript>	Provides alternative content for scripts
<template>	Defines a template for data to come
<slot>	Placeholder for data in components

<canvas>	Renders dynamic bitmap graphics

ATTRIBUTES

Attributes are the way authors have to define properties for an element. These properties usually change the way a browser interpret the element, by changing its meaning or presentation.

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The href Attribute:

The `<a>` tag defines a hyperlink. The `href` attribute specifies the URL of the page the link goes to:

Example:

```
<a href="https://www.vit.edu">Visit VIT Pune </a>
```

The src Attribute

The `<img>` tag is used to embed an image in an HTML page. The `src` attribute specifies the path to the image to be displayed:

Example

```
<img src="img_mld.bmp">
```

Absolute URL - Links to an external image that is hosted on another website. Example: src="https://www.vit.edu/images/img_mld.bmp".

Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: src="img_mld.bmp". If the URL begins with a slash, it will be relative to the domain. Example: src="/images/img_mld.jpg".

The width and height Attributes

The `<img>` tag should also contain the `width` and `height` attributes, which specifies the width and height of the image (in pixels):

Example

```
<img src="img_mld.jpg" width="500" height="600">
```

The alt Attribute

```
<img src="img_mld.jpg" alt="Manik in Ireland">
```

The lang Attribute -- Language code and country code

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page.

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

The title Attribute

The `title` attribute defines some extra information about an element.

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

The style Attribute

The `style` attribute is used to add styles to an element, such as color, font, size, and more.

```
<tagname style="property:value;">
```

- Use the `style` attribute for styling HTML elements
- Use `background-color` for background color
- Use `color` for text colors
- Use `font-family` for text fonts
- Use `font-size` for text sizes
- Use `text-align` for text alignment

Background Color

The CSS `background-color` property defines the background color for an HTML element.

Example: Set the background color for a page to powder blue:

```
<body style="background-color:powderblue;">

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
```

Example: Set background color for two different elements:

```
<body>

<h1 style="background-color:powderblue;">This is a heading</h1>
<p style="background-color:tomato;">This is a paragraph.</p>

</body>
```

Text Color

The CSS `color` property defines the text color for an HTML element:

Example

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

<p style="color: green ;"> This is a green field. </p>
```

Fonts

The CSS `font-family` property defines the font to be used for an HTML element:

Example

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

Text Size

The CSS `font-size` property defines the text size for an HTML element:

Example

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

Text Alignment

The CSS `text-align` property defines the horizontal text alignment for an HTML element:

Example

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

HTML Forms Element: An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

Example

First name:

manik

Last name:

know nunk

Submit

The HTML `<form>` element is used to create an HTML form for user input:

```
<form>
.
form elements: label and input
.
</form>
```

The <label> Element: The `<label>` tag defines a label for many form elements. The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the users focus on the input element. The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.

```
<label>   </label>

<label for="Login">User Login:</label>
```

The <input> Element: The HTML `<input>` element is the most used form element. An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)

<input type="button">	Displays a clickable button

```
<input type="text">
<input type="text" id="ulogin" name="Login">
```

```
<form name="loginform" onsubmit="return validateForm()" action="index.htm" method="post">

  <label for="username">Username:</label>

  <input type="text" placeholder="Enter Username" id="username" name="username" required>

  <label for="pwd">Password :</label>
  <input type="password"  placeholder="Enter Password" id="pwd" name="pwd" required>

  <input type="submit" value="Login">
</form>
```

12 Oct 2020

The HTML <form> Elements

The HTML `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`
- `<option>`
- `<optgroup>`

The `<select>` element defines a drop-down list:
The `<option>` elements defines an option that can be selected.

```
<option value="fiat" selected>Fiat</option>
```

The `<textarea>` element defines a multi-line input field (a text area):

Example

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

The `<button>` element defines a clickable button:

The `<fieldset>` element is used to group related data in a form.

The `<legend>` element defines a caption for the `<fieldset>` element.

The `<datalist>` element specifies a list of pre-defined options for an `<input>` element.

Perform a calculation and show the result in an `<output>` element:

HTML Input Types

Here are the different input types you can use in HTML:

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

HTML Form Attributes

Action Attribute : **The `action` attribute defines the action to be performed when the form is submitted.**

Usually, the form data is sent to a file on the server when the user clicks on the submit button. In the example below, the form data is sent to a file called "action_page.php". This file contains a server-side script that handles the form data:

```
<form name="loginform" onsubmit="return validateForm()" action="index.htm"
        method="post">
<form action="/action_page.php">
```

The Target Attribute

The `target` attribute specifies where to display the response that is received after submitting the form.

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

The Method Attribute

The `method` attribute specifies the HTTP method to use used when submitting the form data.

The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

```
<form action="/action_page.php" method="get">
```

```
<form action="/action_page.php" method="post">
```

13 Oct 2020

CSS Solved a Big Problem:

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to **describe the content** of a web page, like:

<h1>This is a heading</h1>

This is a paragraph.

CSS - Cascading Style Sheets

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Example

```
<style>

body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
```

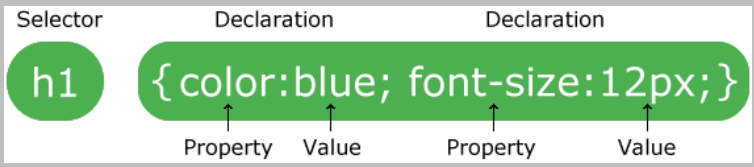
```
  font-family: verdana;
  font-size: 20px;
}

</style>
```

CSS Saves a Lot of Work!

The style definitions are normally saved in external .css files.

CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



Example

In this example all <p> elements will be center-aligned, with a red text color:

```
p {
  color: red;
  text-align: center;
}
```

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

The CSS id Selector

```
#para1 {
  text-align: center;
  color: red;
}
```

Hello World!

This paragraph is not affected by the style.

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

```
.center {
  text-align: center;
  color: red;
}
```

```
<h1 class="center">Red and center-aligned heading</h1>
```

```
<p class="center">Red and center-aligned paragraph.</p>
```

Element specific class

Example

In this example only elements with class="center" will be center-aligned:

```
p.center {
  text-align: center;
  color: red;
}
```

HTML elements can also refer to more than one class.

```
p.center {
  text-align: center;
  color: red;
}

p.large {
  font-size: 300%;
}
```

```
<p class="center large">This paragraph refers to two classes.</p>
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Example

The CSS rule below will affect every HTML element on the page:

```
* {
  text-align: center;
  color: blue;
}
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}
```

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

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

"mystyle.css"

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

Example

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Example

Inline styles are defined within the "style" attribute of the relevant element:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

Comments

```
In HTML          <!.... For Responsive >

IN CSS          /* This is a single-line comment */

p {
  color: red;   /* Set text color to red */
}

/* This is
a multi-line
comment */
```

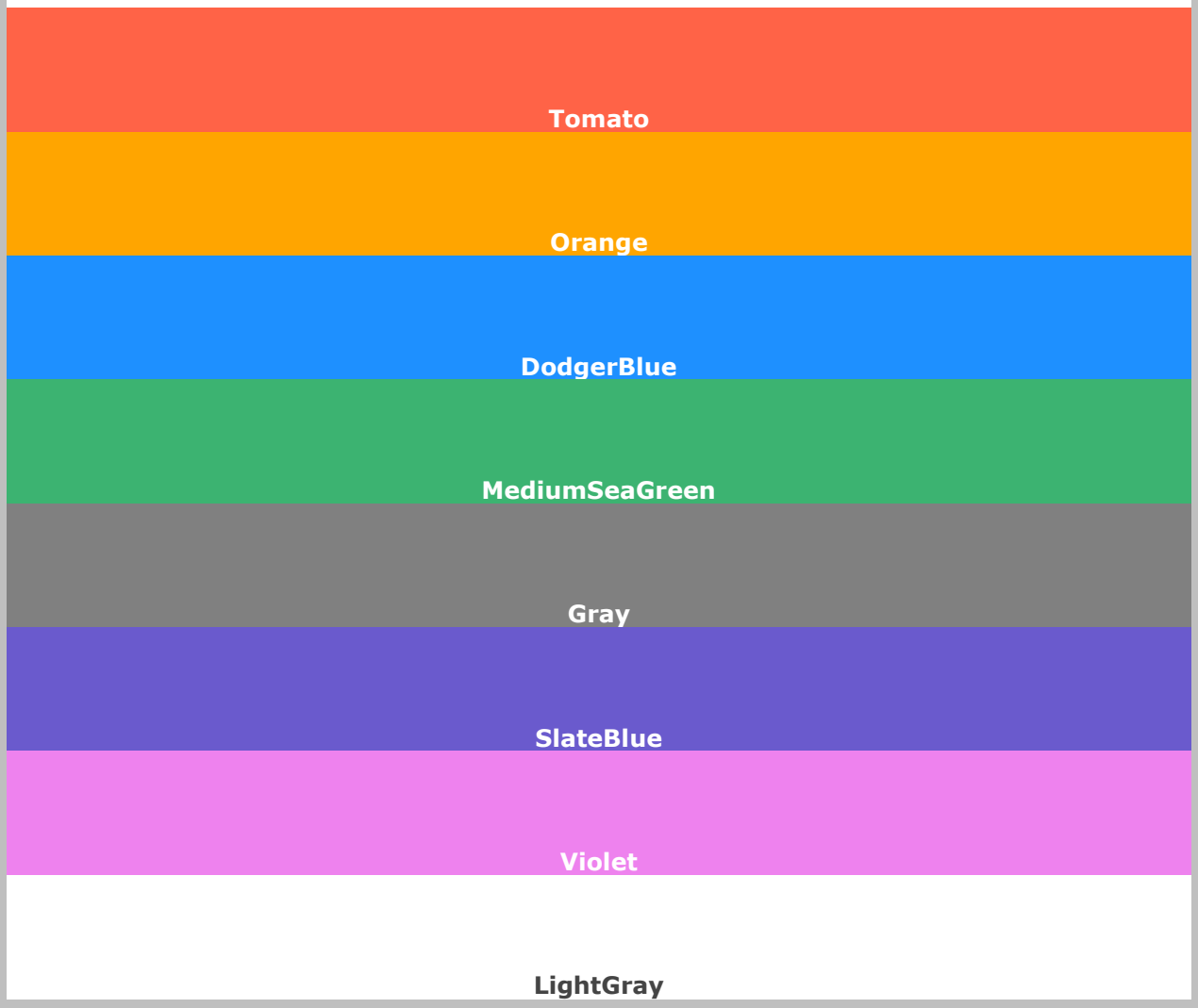
CSS Colors

Colors are specified using predefined color names
Or
RGB, HEX, HSL(hue, saturation, and lightness), RGBA(red, green, blue, alpha), HSLA (hue, saturation, lightness, alpha))values where alpha denotes opacity.

In CSS, a color can be specified by using a color name:

```
<html>

<body>

<h1 style="background-color:Tomato;">Tomato</h1>

<h1 style="background-color:Orange;">Orange</h1>
```

```
<h1 style="background-color:DodgerBlue;">DodgerBlue</h1>

<h1 style="background-color:MediumSeaGreen;">MediumSeaGreen</h1>

<h1 style="background-color:Gray;">Gray</h1>

<h1 style="background-color:SlateBlue;">SlateBlue</h1>

<h1 style="background-color:Violet;">Violet</h1>

<h1 style="background-color:LightGray;">LightGray</h1>

</body>

</html>
```



CSS Text Color

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

CSS Border Color

You can set the color of borders:

Hello World

Hello World

Hello World

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

CSS Color Values

In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato":

rgb(255, 99, 71)

#ff6347

hsl(9, 100%, 64%)

Same as color name "Tomato", but 50% transparent: **rgba(255, 99, 71, 0.5), hsla(9, 100%, 64%, 0.5)**

Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

RGB Value

In CSS, a color can be specified as an RGB value, using this formula:

rgb(*red, green, blue*)

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.

For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.

To display black, set all color parameters to 0, like this: rgb(0, 0, 0).

To display white, set all color parameters to 255, like this: rgb(255, 255, 255).

Example

rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)

rgb(238, 130, 238)

rgb(255, 165, 0)

rgb(106, 90, 205)

Shades of gray are often defined using equal values for all the 3 light sources:

Example

rgb(0, 0, 0)

rgb(60, 60, 60)

rgb(120, 120, 120)

rgb(180, 180, 180)

rgb(240, 240, 240)

rgb(255, 255, 255)

RGBA Value

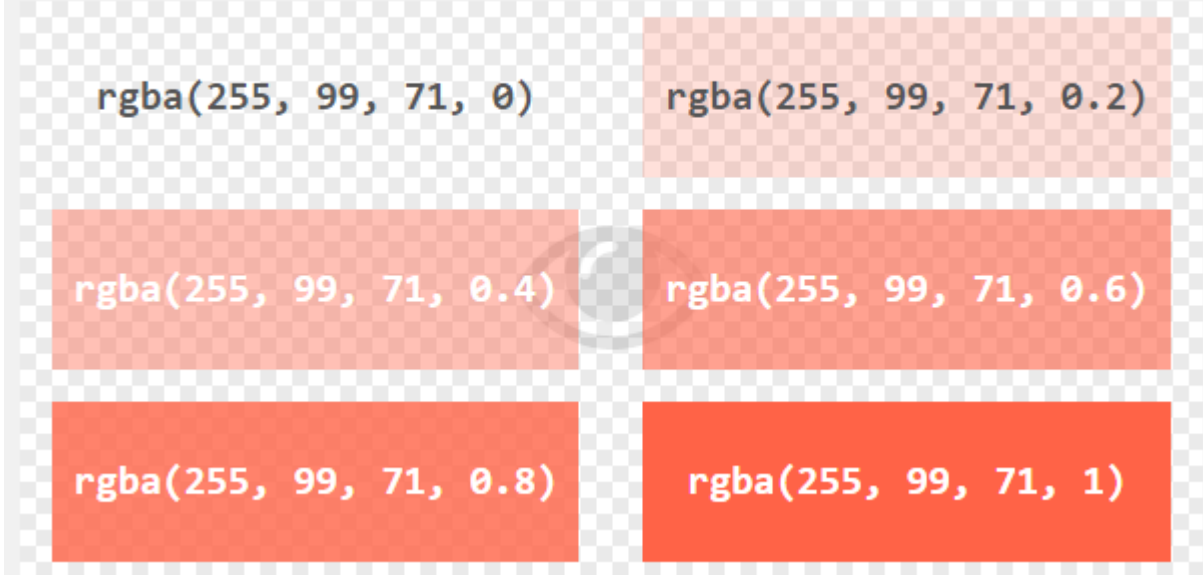
RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with:

rgba(*red, green, blue, alpha*)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example



HEX Value

In CSS, a color can be specified using a hexadecimal value in the form:

#rrggbb

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

Experiment by mixing the HEX values below:

#ff6347

Example

#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

#6a5acd

Shades of gray are often defined using equal values for all the 3 light sources:

Example

#000000

#3c3c3c

#787878

#b4b4b4

#f0f0f0

#ffffff

HSL Value

In CSS, a color can be specified using hue, saturation, and lightness (HSL) in the form:

hsl(*hue, saturation, lightness*)

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

Example

Saturation

Saturation can be described as the intensity of a color.

100% is pure color, no shades of gray

50% is 50% gray, but you can still see the color.

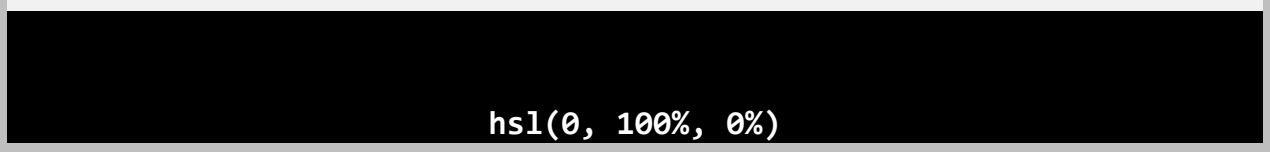
0% is completely gray, you can no longer see the color.

Example

Lightness

The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light) 100% means full lightness (white).

Example



hsl(0, 100%, 25%)

hsl(0, 100%, 50%)

hsl(0, 100%, 75%)

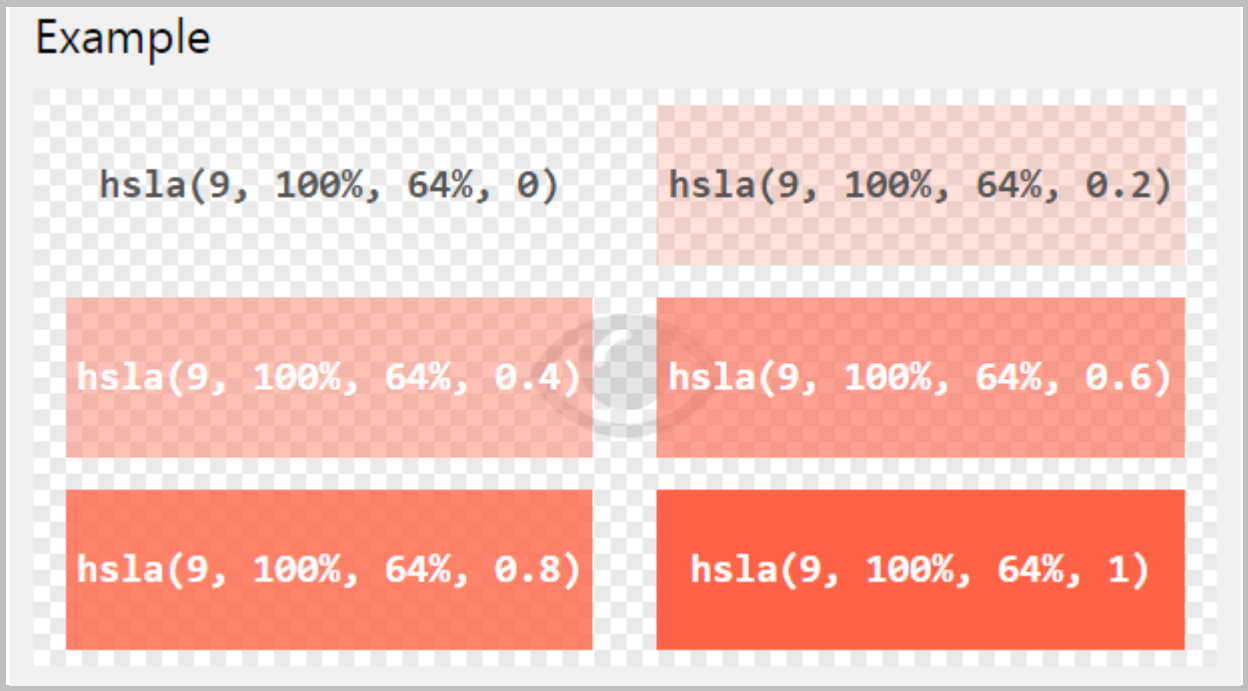
hsl(0, 100%, 90%)

hsl(0, 100%, 100%)

hsla(*hue, saturation, lightness, alpha*)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Experiment by mixing the HSLA values below:



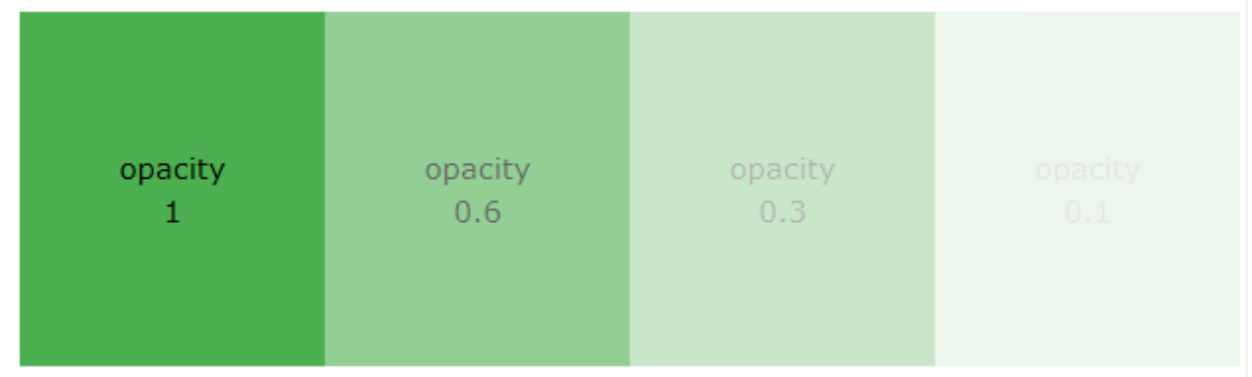
CSS Backgrounds

The CSS background properties are used to define the background effects for elements.

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Example

```
div {
  background-color: green;
  opacity: 0.3;
}
```



Transparency using RGBA

```
div {
  background: rgba(0, 128, 0, 0.3) /* Green background with 30% opacity */
}
```

CSS background-image

```
body {
  background-image: url("paper.gif");
}
```

CSS background - Shorthand property

```
body {
  background-color: #ffffff;
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```