HTML NOTES

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
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

The <!DOCTYPE> Declaration

The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The `<!DOCTYPE>` declaration is not case sensitive.

The `<!DOCTYPE>` declaration for HTML5 is:

```
<!DOCTYPE html>
```

HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag:

```
<p>This is a paragraph.</p>
```

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>
```

HTML Links

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

The link's destination is specified in the `href` attribute.

Attributes are used to provide additional information about HTML elements.

HTML Images

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

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

HTML Attributes

- 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.google.com">Visit W3Schools</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.jpg">
```

There are two ways to specify the URL in the `src` attribute:

1. Absolute URL - Links to an external image that is hosted on another website.

Notes: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

2. 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.

Tip: It is almost always best to use relative URLs. They will not break if you change domain.

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.jpg" width="500" height="600">
```

The alt Attribute

The required `alt` attribute for the `<img>` tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the `src` attribute, or if the user uses a screen reader.

Example

```
<img src="img.jpg" alt="">
```

The style Attribute

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

Example

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

The lang Attribute

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

The following example specifies English as the language:

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

Country codes can also be added to the language code in the `lang` attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.

The following example specifies English as the language and United States as the country:

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

The title Attribute

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

The value of the title attribute will be displayed as a tooltip when you mouse over the element:

Example

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

HTML Formatting Elements

Formatting elements were designed to display special types of text:

- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

HTML STYLES

- 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

HTML Image Maps

Image Maps

The HTML `<map>` tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more `<area>` tags.

Example:

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">


<map name="workmap">

  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">

  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">

  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">

</map>
```

HTML Tables

HTML tables allow web developers to arrange data into rows and columns.

A simple HTML table:

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

HTML Lists

HTML lists allow web developers to group a set of related items in lists.

Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Description Lists

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

HTML class Attribute

The HTML `class` attribute is used to specify a class for an HTML element.

Multiple HTML elements can share the same class.

Using The class Attribute

The `class` attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.

In the following example we have three `<div>` elements with a `class` attribute with the value of "city". All of the three `<div>` elements will be styled equally according to the `.city` style definition in the head section:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
```

```
</div>

<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

HTML id Attribute

The HTML `id` attribute is used to specify a unique id for an HTML element.

You cannot have more than one element with the same id in an HTML document.

Using The id Attribute

The `id` attribute specifies a unique id for an HTML element. The value of the `id` attribute must be unique within the HTML document.

The `id` attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

The syntax for id is: write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

In the following example we have an `<h1>` element that points to the id name "myHeader". This `<h1>` element will be styled according to the `#myHeader` style definition in the head section:

Example

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

```
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

Difference Between Class and ID

A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page:

Example

```
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<!-- An element with a unique id -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple elements with same class -->
```

```
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

HTML Forms

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

The <form> Element

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

```
<form>
.
form elements
.
</form>
```

The <input> Element

The HTML `<input>` element is the most used form element.

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

Text Fields

The `<input type="text">` defines a single-line input field for text input.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

The <label> Element

Notice the use of the <label> element in the example above.

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 user 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.

Radio Buttons

The `<input type="radio">` defines a radio button.

A form with radio buttons:

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaSc
ript">
  <label for="javascript">JavaScript</label>
</form>
```

Checkboxes

The `<input type="checkbox">` defines a **checkbox**.

The Submit Button

The `<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's `action` attribute.

Example

A form with a submit button:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

CSS NOTES

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

A CSS rule 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;
}
```

CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

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

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The CSS rule below will be applied to the HTML element with id="para1":

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

The CSS class Selector

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

To select elements with a specific class, write a period (.) character, followed by the class name.

In this example all HTML elements with class="center" will be red and center-aligned:

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

The CSS Universal Selector

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

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

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

Basic rule syntax

Style rule syntax

```
style-rule ::=

    selectors-list {

      properties-list

    }
```

Where :

```
selectors-list ::=

    selector[:pseudo-class] [::pseudo-element]

    [, selectors-list]



properties-list ::=[property : value] [; properties-list]
```

Style rule examples

```
strong {
  color: red;
}

div.menu-bar li:hover > ul {
  display: block;
}
```

At-rule syntax

As the structure of at-rules varies widely, please see At-rule to find the syntax of the specific one you want.

Index

Note: The property names in this index do not include the JavaScript names where they differ from the CSS standard names.

--*

-webkit-line-clamp

A

- accent-color
- :active
- additive-symbols (@counter-style)
- ::after (:after)
- align-content
- align-items
- align-self
- align-tracks
- all
- <an-plus-b>
- <angle>
- <angle-percentage>
- animation
- animation-delay
- animation-direction
- animation-duration
- animation-fill-mode
- animation-iteration-count
- animation-name
- animation-play-state
- animation-timeline
- animation-timing-function
- @annotation
- annotation()

- :any-link
- appearance
- ascent-override (@font-face)
- aspect-ratio
- attr()

B

- ::backdrop
- backdrop-filter
- backface-visibility
- background
- background-attachment
- background-blend-mode
- background-clip
- background-color
- background-image
- background-origin
- background-position
- background-position-x
- background-position-y
- background-repeat
- background-size
- <basic-shape>
- ::before (:before)
- :blank
- bleed (@page)
- <blend-mode>
- block-overflow
- block-size
- blur()
- border
- border-block
- border-block-color
- border-block-end
- border-block-end-color
- border-block-end-style
- border-block-end-width
- border-block-start
- border-block-start-color
- border-block-start-style
- border-block-start-width
- border-block-style
- border-block-width
- border-bottom
- border-bottom-color
- border-bottom-left-radius
- border-bottom-right-radius
- border-bottom-style
- border-bottom-width
- border-collapse
- border-color
- border-end-end-radius
- border-end-start-radius
- border-image
- border-image-outset
- border-image-repeat
- border-image-slice
- border-image-source

- border-image-width
- border-inline
- border-inline-color
- border-inline-end
- border-inline-end-color
- border-inline-end-style
- border-inline-end-width
- border-inline-start
- border-inline-start-color
- border-inline-start-style
- border-inline-start-width
- border-inline-style
- border-inline-width
- border-left
- border-left-color
- border-left-style
- border-left-width
- border-radius
- border-right
- border-right-color
- border-right-style
- border-right-width
- border-spacing
- border-start-end-radius
- border-start-start-radius
- border-style
- border-top
- border-top-color
- border-top-left-radius
- border-top-right-radius
- border-top-style
- border-top-width
- border-width
- bottom
- @bottom-center
- box-decoration-break
- box-shadow
- box-sizing
- break-after
- break-before
- break-inside
- brightness()

C

- calc()
- caption-side
- caret-color
- @character-variant
- character-variant()
- @charset
- :checked
- circle()
- clamp()
- clear
- clip
- clip-path
- <color>
- color

- color-scheme
- column-count
- column-fill
- column-gap
- column-rule
- column-rule-color
- column-rule-style
- column-rule-width
- column-span
- column-width
- columns
- conic-gradient()
- contain
- content
- content-visibility
- contrast()
- <counter>
- counter-increment
- counter-reset
- counter-set
- @counter-style
- counters()
- cross-fade()
- cubic-bezier()
- ::cue
- ::cue-region
- :current
- cursor
- <custom-ident>
- length#cap
- length#ch
- length#cm

D

- angle#deg
- :default
- :defined
- descent-override (@font-face)
- <dimension>
- :dir
- direction
- :disabled
- display
- <display-box>
- <display-inside>
- <display-internal>
- <display-legacy>
- <display-listitem>
- <display-outside>
- drop-shadow()
- resolution#dpcm
- resolution#dpi
- resolution#dppx

E

- element()
- ellipse()

- :empty
- empty-cells
- :enabled
- env()
- length#em
- length#ex

F

- fallback (@counter-style)
- filter
- <filter-function>
- :first
- :first-child
- ::first-letter (:first-letter)
- ::first-line (:first-line)
- :first-of-type
- fit-content()
- <flex>
- flex
- flex-basis
- flex-direction
- flex-flow
- flex-grow
- flex-shrink
- flex-wrap
- flex_value#fr
- float
- :focus
- :focus-visible
- :focus-within
- font
- font-display (@font-face)
- @font-face
- font-family
- font-family (@font-face)
- font-feature-settings
- font-feature-settings (@font-face)
- @font-feature-values
- font-kerning
- font-language-override
- font-optical-sizing
- font-size
- font-size-adjust
- font-stretch
- font-stretch (@font-face)
- font-style
- font-style (@font-face)
- font-synthesis
- font-variant
- font-variant (@font-face)
- font-variant-alternates
- font-variant-caps
- font-variant-east-asian
- font-variant-ligatures
- font-variant-numeric
- font-variant-position
- font-variation-settings
- font-variation-settings (@font-face)

- image-set()
- @import
- :in-range
- :indeterminate
- inherit
- inherits (@property)
- initial
- initial-letter
- initial-letter-align
- initial-value (@property)
- inline-size
- input-security
- inset
- inset()
- inset-block
- inset-block-end
- inset-block-start
- inset-inline
- inset-inline-end
- inset-inline-start
- <integer>
- :invalid
- invert()
- :is
- isolation
- length#ic
- length#in

J

- justify-content
- justify-items
- justify-self
- justify-tracks

K

- frequency#kHz
- @keyframes

L

- :lang
- :last-child
- :last-of-type
- @layer
- layer()
- layer() (@import)
- leader()
- :left
- left
- @left-bottom
- <length>
- <length-percentage>
- letter-spacing
- line-break
- line-clamp
- line-gap-override (@font-face)

- line-height
- line-height-step
- linear-gradient()
- :link
- list-style
- list-style-image
- list-style-position
- list-style-type
- local()
- :local-link

M

- length#mm
- margin
- margin-block
- margin-block-end
- margin-block-start
- margin-bottom
- margin-inline
- margin-inline-end
- margin-inline-start
- margin-left
- margin-right
- margin-top
- margin-trim
- ::marker
- marks (@page)
- mask
- mask-border
- mask-border-mode
- mask-border-outset
- mask-border-repeat
- mask-border-slice
- mask-border-source
- mask-border-width
- mask-clip
- mask-composite
- mask-image
- mask-mode
- mask-origin
- mask-position
- mask-repeat
- mask-size
- mask-type
- masonry-auto-flow
- math-style
- matrix()
- matrix3d()
- max()
- max-block-size
- max-height
- max-height (@viewport)
- max-inline-size
- max-lines
- max-width
- max-width (@viewport)
- max-zoom (@viewport)
- @media

- overscroll-behavior
- overscroll-behavior-block
- overscroll-behavior-inline
- overscroll-behavior-x
- overscroll-behavior-y

P

- Pseudo-classes
- Pseudo-elements
- length#pc
- length#pt
- length#px
- pad (@counter-style)
- padding
- padding-block
- padding-block-end
- padding-block-start
- padding-bottom
- padding-inline
- padding-inline-end
- padding-inline-start
- padding-left
- padding-right
- padding-top
- @page
- page-break-after
- page-break-before
- page-break-inside
- paint()

etc.

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

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

</body>
</html>
```

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.

```
<!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.

```
<!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>
```

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

Thankyou