



Web programming I

Section 4

Eng/ Mona AbdelAziz

HTML Multimedia

- Multimedia elements (like audio or video) are stored in media files.

Format	File	Description
MPEG	.mpg .mpeg	The most popular video format on the web. Supported by all browsers, but not supported in HTML5.
Flash	.swf .flv	Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Supported by HTML5.
WebM	.webm	Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
MPEG-4 or MP4	.mp4	Supported by all HTML5 browsers. Recommended by YouTube.

HTML Multimedia cont

[Table 6-1] Web Audio Formats

Format	File	Description
WAV	.wav	Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5.
Ogg	.ogg	Supported by HTML5.
MP3	.mp3	MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. MP4 audio format is supported by all browsers.

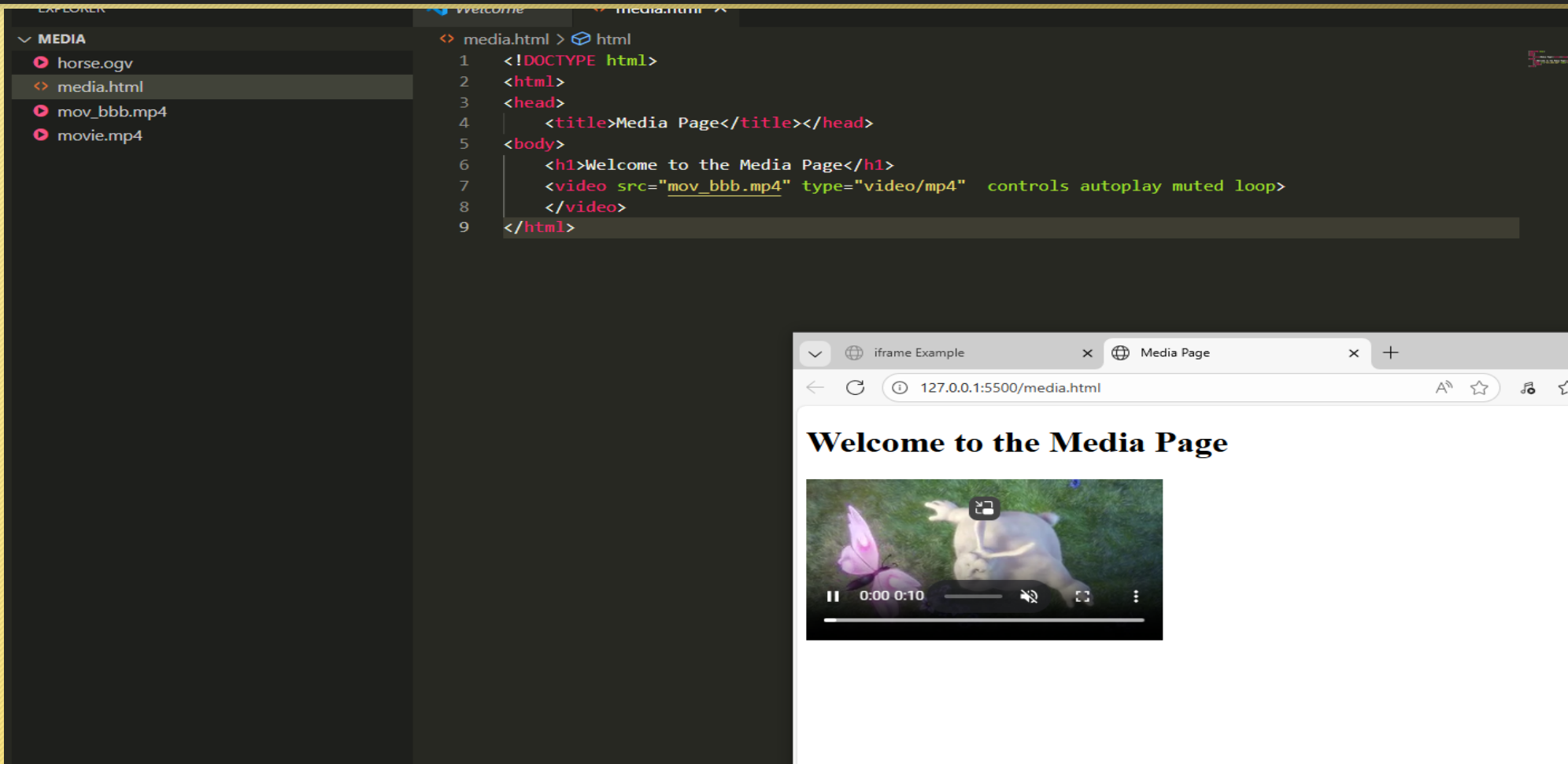
<video> element

- The HTML <video> element is used to show a video on a web page.

Video attributes

- **src**: The URL of the video file to embed. However, it's more common to use <source> elements within <video> to support multiple formats.
- **width** and **height**: Sets the display dimensions of the video in pixels.
- **controls**: Adds video controls (play, pause, volume, etc.) for users.
- **autoplay**: Plays the video automatically upon loading. This is often combined with muted, as many browsers only autoplay muted videos.
- **muted**: Mutes the audio of the video by default.
- **loop**: Replays the video continuously when it reaches the end.

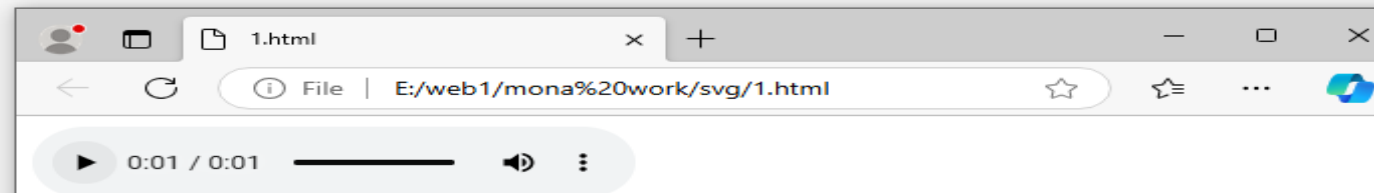
Vedio EX



<audio> Element

- To play an audio file in HTML, use the <audio> element

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <audio controls>
6    <source src="horse.ogg" type="audio/ogg">
7    <source src="horse.mp3" type="audio/mpeg">
8    Your browser does not support the audio element.
9  </audio>
10 </body>
11 </html>
12
13
```



Difference

```

v MEDIA
  horse.ogv
  media.html
  mov_bbb.mp4
  movie.mp4

media.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Media Page</title></head>
5  <body>
6    <h1>Welcome to the Media Page</h1>
7    <video controls autoplay muted loop>
8      <source src="movi_bbb.mp4" type="video/mp4">
9      <source src="movie.mp4" type="video/mp4">
10   </video>
11 </html>

```

Welcome to the Media Page



<iframe> attributes

Attribute	Description
SRC	Specifies the URL of the page or video to display inside the iframe.
width	Sets the width of the iframe (in pixels or %).
height	Sets the height of the iframe (in pixels or %).
title	Provides a text description for accessibility (screen readers).
frameborder	Specifies whether to display a border (use "0" or "1"). (Now replaced by CSS).
allowfullscreen	Allows fullscreen mode for videos.
loading	Controls how the iframe loads: "lazy" (delayed) or "eager" (immediate).

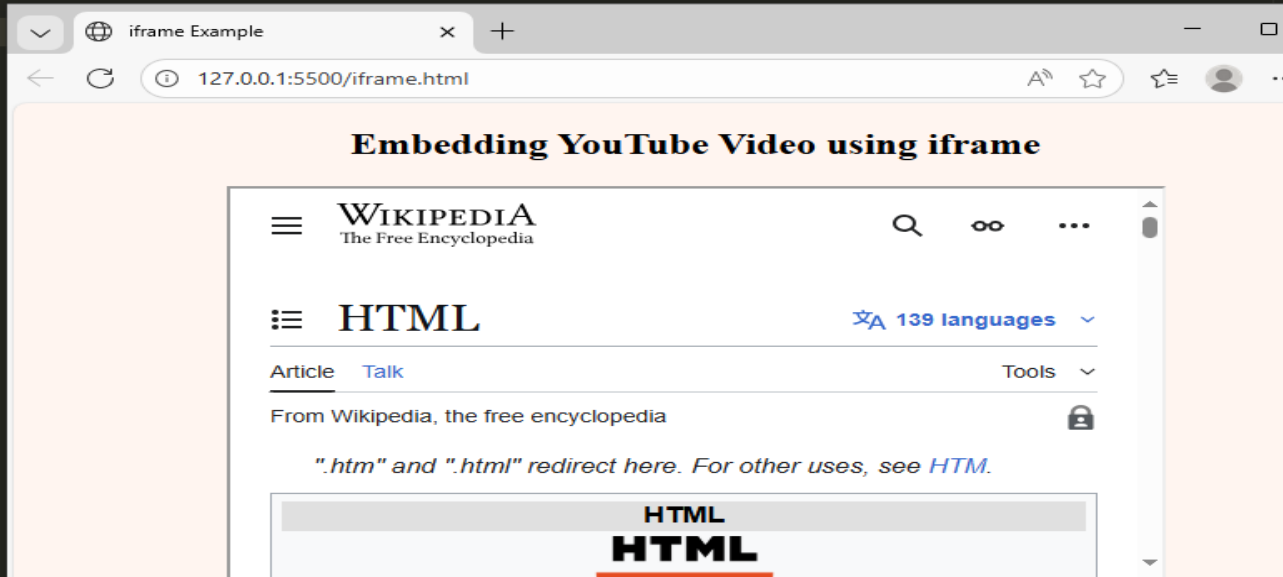
Iframe ex

frame.html > html > body > iframe

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>iframe Example</title>
</head>
<body style="text-align:center; background-color: #fff5f0;">

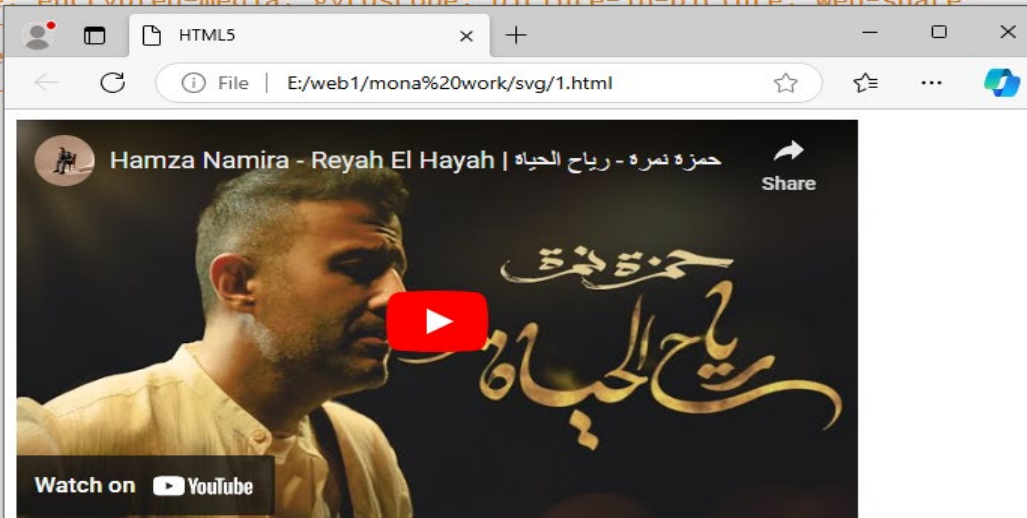
  <h2>Embedding YouTube Video using iframe</h2>

  <iframe width="560" height="315"
    src="https://en.wikipedia.org/wiki/HTML"
    title="YouTube video player"
    frameborder="1"
    allowfullscreen>
  </iframe>
</body>
</html>
```



Iframe element

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>
5   HTML5
6 </title>
7 </head>
8 <body>
9   <iframe width="560" height="315" src="https://www.youtube.com/embed/Apl8h-P0F9Q?
10     si=sy1XUfTisdUG0u2m" title="YouTube video player" frameborder="0" allow="accelerometer;
11     autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture: web-share"
12     referrerpolicy="strict-origin-when-downgrade" >
13   </iframe>
14 </body>
</html>
```



<meta> tag & attributes

- provides metadata about the HTML document, which is information not displayed directly on the page but used by browsers, search engines, and other services.
- appear in search engines
- has many attributes

```
1  <!DOCTYPE html>
2 ▼ <html>
3 ▼   <head>
4     <meta>
5   </head>
6 ▼ <body>
```

<meta> attributes

viewport:

- Controls the page's dimensions and scaling on mobile devices.
- Typically used for responsive design.
- Example: `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

```
1  <!DOCTYPE html>
2  ▼ <html>
3  ▼   <head>
4      <!-- <meta charset="iso-8859-1">-->
5          <meta charset="utf-8">
6          <meta name="description" content="our first website">
7          <meta name="author" content="second year IT ">
8          <meta name="keywords" content="BTU IT Track">
9          <meta name="viewport" content="width=device-width,initial-scale=1.0">
10     </head>
11     <body>
12 </body>
13 </html>
```


<meta attributes>

Charset Declaration

Defines the character encoding for the HTML document.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5  </head>
```

Name

- Specifies the name of the metadata.
- Common values: "description", "keywords", "author", "viewport".

```
3  <head>
4    <meta charset="utf-8">
5    <meta name="description" content="our first website">
6    <meta name="author" content="second year IT ">
7    <meta name="keywords" content="BTU IT Track">
8  </head>
```


Auto refresh

- Auto reload live data pages (like news or stock updates).
- Redirect users after login/logout.
- Create a short “**Welcome Page**” that jumps to the main site.
- **Syntax**

`<meta http-equiv="refresh" content="time; url=target_page">`

```
3 <head>
4   <title>Media Page</title></head>
5   <meta http-equiv="refresh" content="3;url=page2.html">
```

Cascading Style Sheets (CSS) Fundamentals

CSS



- Defines the visual style of web pages
- CSS describes how HTML elements should be displayed.
- With CSS you can quickly and easily restyle any element.
- One way you can add styling to a web page is by inserting the required statements into the head of the page, between the
- <head> and </head> tags.

How to add CSS Code to your webpage

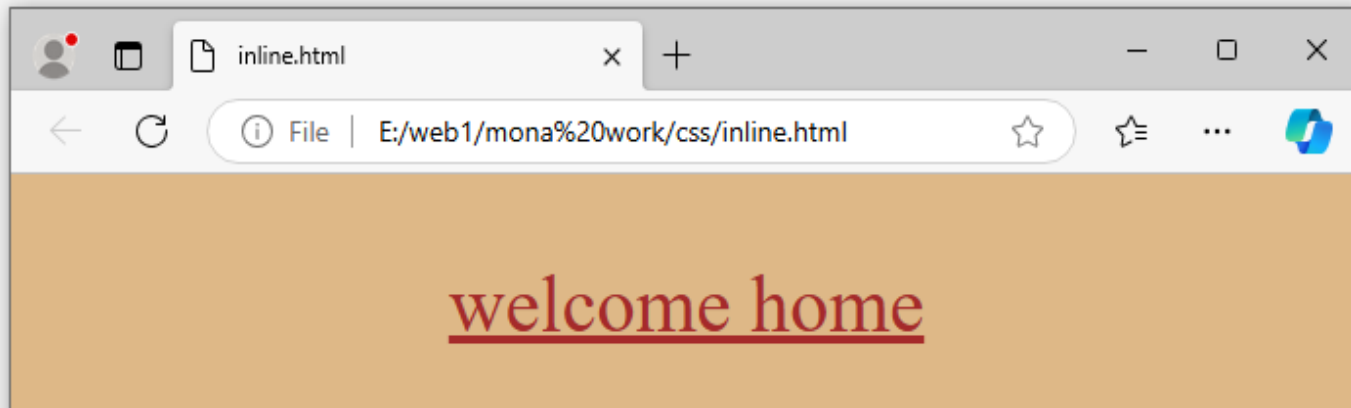
1-Inline CSS: Styles added directly to the HTML element

2-Internal CSS: Styles defined at the head section of the document.

3-External CSS: insert a link tag in the head section to load a CSS file (Styles defined in a separate file)

Inline CSS

```
1 <!doctype html>
2 <html>
3 <head>
4
5     </head>
6 <body style="background-color: burlywood">
7     <p style="color: brown;font-family: serif;font-size: 40px;text-align: center;text-
8       decoration: underline"> welcome home</p>
9 </body>
10 </html>
```



CSS Syntax

Internal CSS

Each CSS rule consists of: Selector

- declaration block
- Selector: specifies the HTML element that we want to apply the styles
- property1 / property2: specifies the attribute of HTML elements that we want to change (color, background, and so on)
- Value : specifies the new value you want to assign to the property (color of the text to the red, background to gray, and so on)

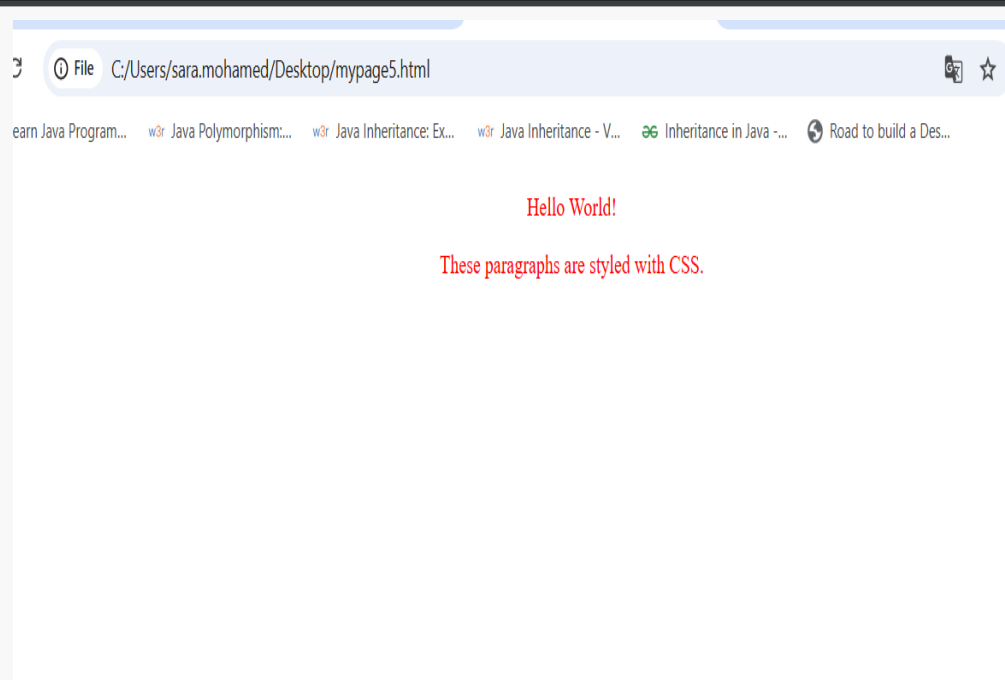


The diagram illustrates the components of a CSS rule. It shows the code: `div { color: blue; font-size: 20px; }`. A blue box labeled "Selector" has a red arrow pointing to the `div` part of the code. A blue bracket on the right side of the code, spanning the lines `color: blue;` and `font-size: 20px;`, has a red arrow pointing to a blue box labeled "Declaration Block".

```
div {  
  color: blue;  
  font-size: 20px;  
}
```


Internal CSS, by inserting a style tag in the head section

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  p {
6    color: red;
7    text-align: center;
8  }
9  </style>
10 </head>
11 <body>
12 <p>Hello World!</p>
13 <p>These paragraphs are styled with CSS.</p>
14 </body>
15 </html>
```



Internal syntax(multiple CSS)

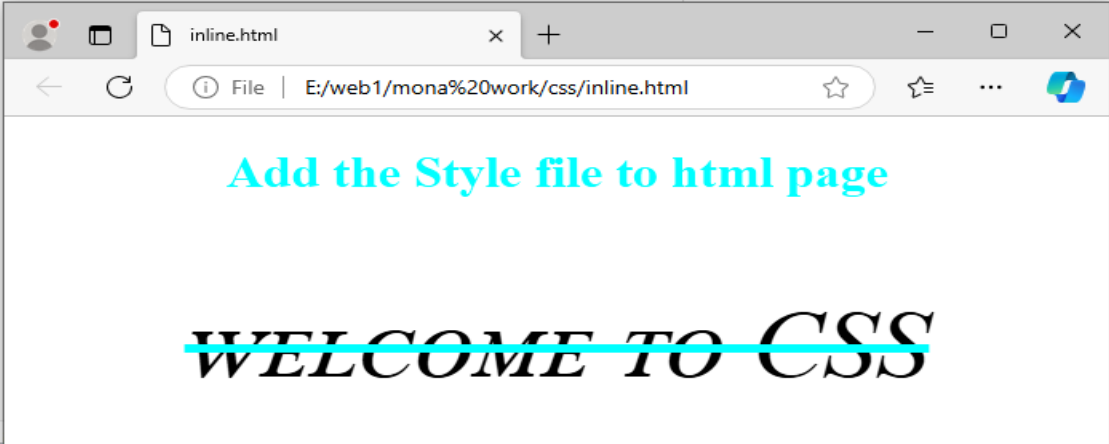
```
1  <!doctype html>
2  <html>
3  <head>
4  <style>
5  body{
6      background-color: blanchedalmond;
7      text-align: center;
8  }
9  }
10 p{
11     color: brown;
12     font-size: 50px
13 }
14
15 </style>
16 </head>
17 <body>
18 <p> welcome to CSS</p>
19 </body>
20 </html>
```



External CSS

```
inline.html
1 <!doctype html>
2 <html>
3 <head>
4   <link rel="stylesheet"
5     href="style.css">
6 </head>
7 <body>
8   <h1> Add the Style file to html
9   page </h1>
10  <p> welcome to CSS</p>
11 </body>
12 </html>

style.css
1 P{
2   text-decoration: line-through;
3   text-decoration-color: aqua;
4   text-justify: auto;
5   text-align: center;
6   font-size: 50pt;
7   font-style: italic;
8   font-variant: small-caps;
9 }
10 h1{
11   color: aqua;
12   text-align: center
13 }
```



Another

```
1 ▼ P{
2   text-decoration: line-through dashed red 2px;
3   text-justify: auto;
4   text-align: center;
5   font-size: 50pt;
6   font-style: italic;
7   font-variant: small-caps;
8 }
9 ▼ h1{
10   color: aqua;
11   text-align: center;
12   text-decoration: overline dotted blue 5px;
13 }
```



Explanation

- text-decoration: line-through;**

Adds a line through the text.

- text-decoration-color: aqua;**

Sets the color of the line-through to aqua.

- text-justify: auto;**

Adjusts the spacing between words. In this case, the default justification behavior is applied.

- text-align: center;**

Centers the text horizontally within its container.

CONT

- font-size: 50pt;**

Sets the font size to 50 points.

- font-style: italic;**

Makes the text italicized.

- font-variant: small-caps;**

Converts the text to small-caps, where lowercase letters are displayed as smaller uppercase letters.

Multiple external CSS

```
m.css
1 ▼ body {
2     background: lightgreen;
3 }
```

```
s.css
1 ▼ p {
2     color: red;
3 }
4 |
```

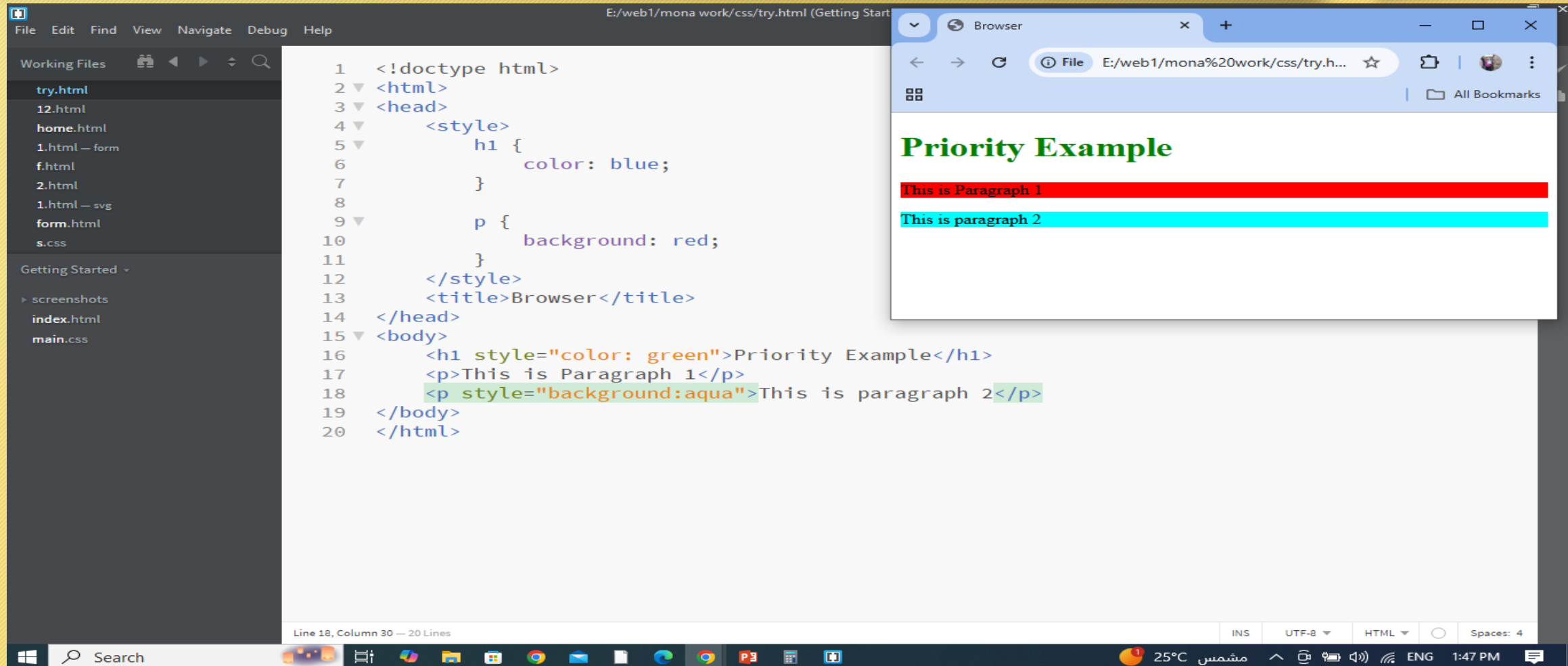
```
1  <!doctype html>
2 ▼ <html>
3 ▼ <head>
4     <link rel="stylesheet" href="s.css">
5     <link rel="stylesheet"
6         href="m.css">
7 ▼ </head>
8     <body>
9         <h1> Add the Style file to html
10        page </h1>
11        <p> welcome to CSS</p>
12        <div> welcome to this page</div>
13    </body>
14 </html>|
```

Add the Style file to html page

welcome to CSS

welcome to this page

Inline Style Override Internal Style



The image shows a code editor on the left and a web browser on the right, illustrating a CSS priority example.

Code Editor (Left):

```
1 <!doctype html>
2 <html>
3 <head>
4 <style>
5     h1 {
6         color: blue;
7     }
8
9     p {
10        background: red;
11    }
12 </style>
13 <title>Browser</title>
14 </head>
15 <body>
16     <h1 style="color: green">Priority Example</h1>
17     <p>This is Paragraph 1</p>
18     <p style="background:aqua">This is paragraph 2</p>
19 </body>
20 </html>
```

Web Browser (Right):

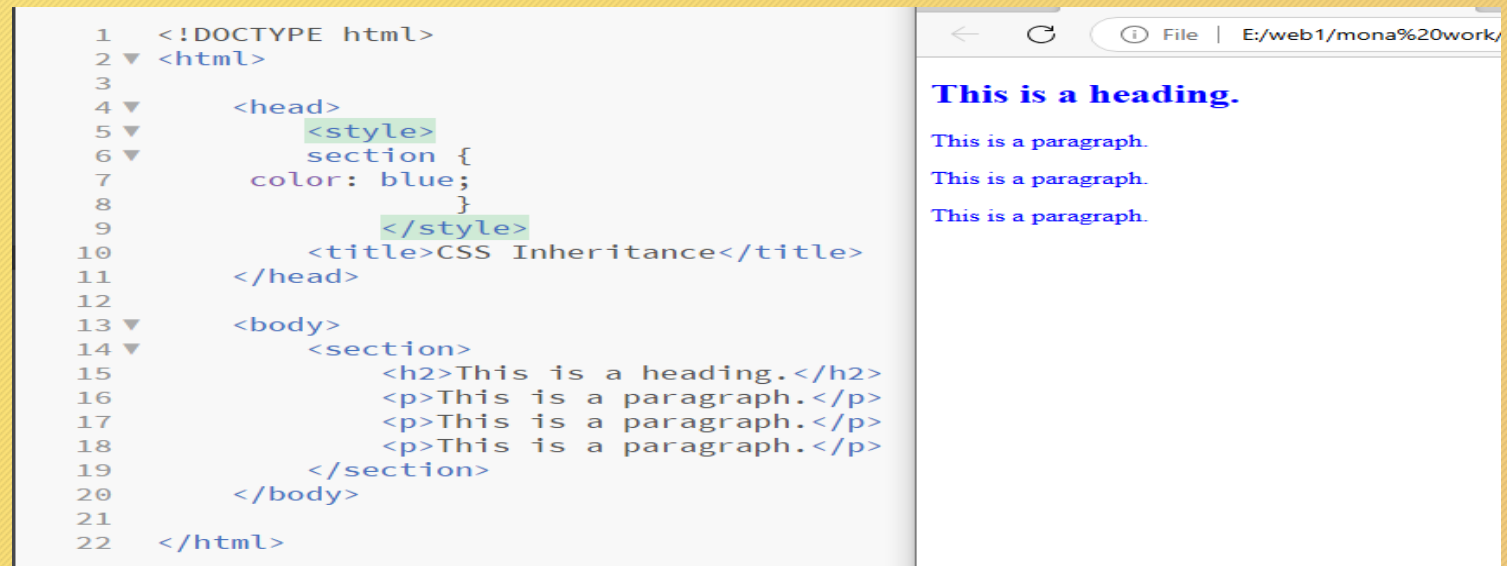
The browser displays the rendered HTML. The title is "Browser". The main heading is "Priority Example" in green. Below it, there are two paragraphs:

- "This is Paragraph 1" (red background)
- "This is paragraph 2" (cyan background)

The browser's address bar shows the file path: `E:/web1/mona%20work/css/try.h...`. The status bar at the bottom indicates the current line and column: "Line 18, Column 30 — 20 Lines".

CSS Essential Concepts

- **Inheritance:** In CSS, inheritance passes the styles directly from the parent element to its child elements.
- The child elements normally take the same styles that have been assigned to the parent, unless they are provided their own styles.



The image shows a side-by-side comparison of an HTML document and its rendered output in a web browser. On the left, the HTML source code is displayed with line numbers 1 through 22. It defines a document type, a head section with a title 'CSS Inheritance', and a body section containing a heading and three paragraphs. A CSS style is defined in the head, targeting a 'section' element and setting its 'color' to 'blue'. On the right, the browser window shows the rendered page. The heading is displayed in blue, bold text, and the three paragraphs below it are also displayed in blue text, demonstrating that the 'color' style is inherited by the child elements of the 'section' parent.

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <style>
6 section {
7   color: blue;
8 }
9 </style>
10 <title>CSS Inheritance</title>
11 </head>
12
13 <body>
14 <section>
15 <h2>This is a heading.</h2>
16 <p>This is a paragraph.</p>
17 <p>This is a paragraph.</p>
18 <p>This is a paragraph.</p>
19 </section>
20 </body>
21
22 </html>
```

Browser output:

This is a heading.

This is a paragraph.

This is a paragraph.

This is a paragraph.

Rule Order

- Rule order refers to the sequence in which CSS rules are applied to the HTML elements

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5  <style>
6  p {
7      color: red;
8  }
9
10     /* overrides color previous color value */
11  p {
12      color: blue;
13  }
14  </style>
15  <title>CSS rule order </title>
16 </head>
17 <body>
18     <p>This is a paragraph.</p>
19
20 </body>
21
22 </html>
```

← ↻ ⓘ File | Edit

This is a paragraph.

CSS Comments

- You can add comment to CSS code by placing your comment within a pair of `/* ... */`,

Simple selectors

- Type Selector: Targets elements by their tag name.
- Class Selector(.): Targets elements by their class attribute.
- ID Selector(#): Targets an element by its id attribute
- Universal Selector (*) :Targets all elements.
- Group Selector: selects all the HTML elements with the same style definitions.

EX

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  p {
6      text-align: center;
7      color: red;
8      font-size: 50pt
9  }
10 #para1{
11     color:blue;
12     font-size: 40pt
13 }
14 .center
15 {
16     color:yellow;
17     font-size: 30pt}
18 </style>
19 </head>
20 <body>
21 <p>Every paragraph will be affected by the style.</p>
22 <p id="para1">Me too!</p>
23 <p class="center">And me!</p>
24
25 </body>
26 </html>
```

Every paragraph
will be affected
the style.

Me too!

And me!

Simple selectors

- in CSS is a selector that targets elements without combining them with other selectors or using complex structures. These are the most basic types of selectors used to apply styles to HTML elements.
- (select elements based on name, id, class) A class name cannot start with a number!

The CSS Universal Selector

- The universal selector selects every single HTML element on the page. It is written using the asterisk(*) character

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  * {
6      color: red;
7      font-size: 30pt
8  }
9  </style>
10 </head>
11 <body>
12 <p>Every paragraph will be affected by the style.</p>
13 <p>Me too!</p>
14 <p>And me!</p>
15
16 </body>
17 </html>
```

Every paragraph will be
affected by the style.

Me too!

And me!

The CSS Grouping Selector

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 h1, h2, p {
6   text-align: center;
7   color: red;
8 }
9 </style>
10 </head>
11 <body>
12
13 <h1>Hello World!</h1>
14 <h2>Smaller heading!</h2>
15 <p>This is a paragraph.</p>
16
17 </body>
18 </html>
```

← ↻ ⓘ File | E:/web1/mona%20wo... ☆ ⋮

Hello World!

Smaller heading!

This is a paragraph.

CSS priority

- !important rules.
- Inline styles.
- (ID > Class > Element).
- Source order (later rules override earlier ones if specificity is the same)

Style Rule Hierarchy

- The style rule hierarchy determines the priority of CSS rules when multiple rules target the same element.

In CSS, the following hierarchy of style priorities applies:

- Inline styling: Styles applied directly within HTML element.
- ID selectors: Styles elements with specific ID.
- Class and attribute selectors: Styles elements with certain class or attribute.
- Element selectors: Styles elements with specific tag name.

!important attribute

The screenshot displays the Brackets code editor on the left and a web browser on the right. The Brackets editor shows a file named `inhertance.html` (note the typo) with the following HTML and CSS code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 /* class selector */
6 .paragraph {
7     color: green;
8 }
9 /* id selector */
10 #unique {
11     color: purple;
12 }
13 /* using !important on element selector */
14 p {
15     color: red !important;
16 }
17 </style>
18 </head>
19 <body>
20     <h1>Heading</h1>
21     <p>This is the first paragraph.</p>
22     <p class="paragraph">This is the second paragraph.</p>
23     <p id="unique">This is the third paragraph.</p>
24 </body>
25 </html>
```

The web browser on the right shows the rendered output of the `inhertance.html` file. It displays a heading and three paragraphs, all of which are red due to the `!important` rule in the CSS.

Heading

This is the first paragraph.

This is the second paragraph.

This is the third paragraph.

Priority example

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5    /* class selector */
6    .paragraph {
7      color: green;
8    }
9    /* id selector */
10   #unique {
11     color: purple;
12   }
13   p {
14     color: red;
15   }
16 </style>
17 </head>
18 <body>
19   <h1>Heading</h1>
20   <p>This is the first paragraph.</p>
21   <p class="paragraph">This is the second paragraph.</p>
22   <p id="unique" class="paragraph">This is the third paragraph.</p>
23 </body>
24 </html>
```

