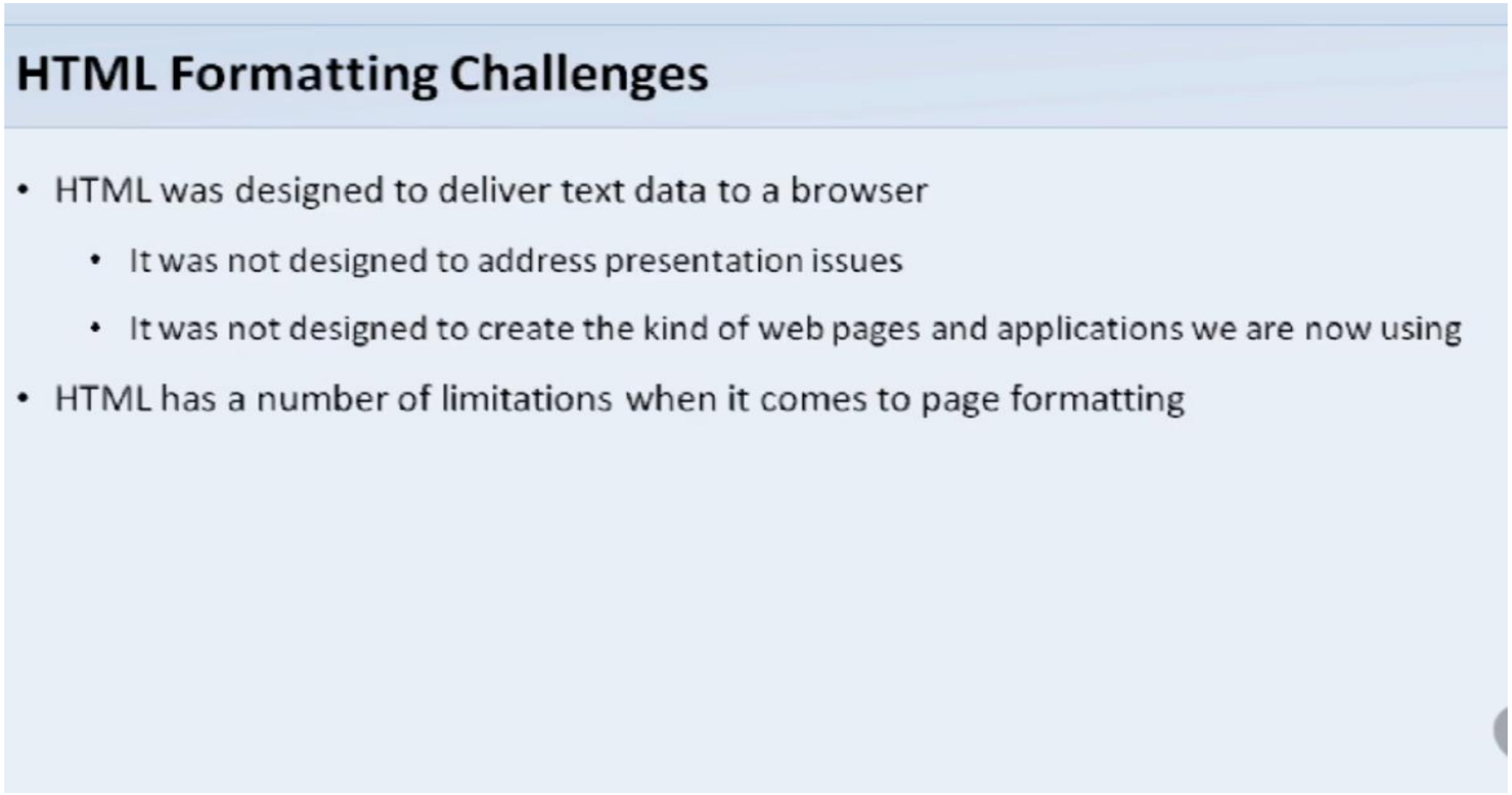
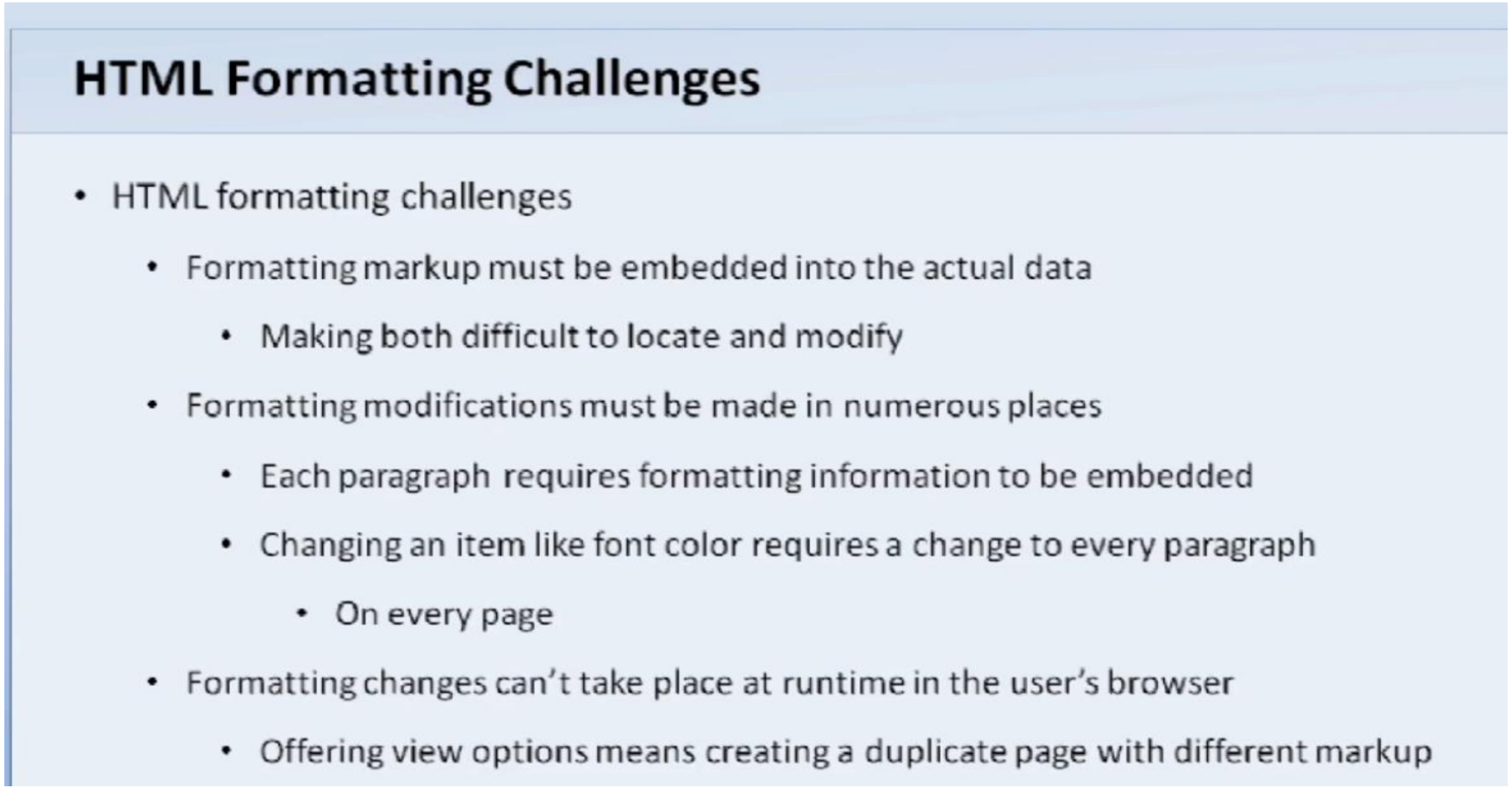
HTML Formatting versus CSS



HTML Formatting Challenges

- HTML was designed to deliver text data to a browser
  - It was not designed to address presentation issues
  - It was not designed to create the kind of web pages and applications we are now using
- HTML has a number of limitations when it comes to page formatting



HTML Formatting Challenges

- HTML formatting challenges
  - Formatting markup must be embedded into the actual data
    - Making both difficult to locate and modify
  - Formatting modifications must be made in numerous places
    - Each paragraph requires formatting information to be embedded
    - Changing an item like font color requires a change to every paragraph
      - On every page
  - Formatting changes can't take place at runtime in the user's browser
    - Offering view options means creating a duplicate page with different markup

```
<p>This is normal text - <b>and this is bold text</b>.</p>
```

HTML

```
<!DOCTYPE HTML>
<html>
<head>
<title> Title of the document</title>
</head>

<body>
The content of the document......
</body>

</html>
```

link

```
<a href="http://www.w3schools.com">Visit W3Schools.com!</a>
```

Use a <figure> element to mark up a photo in a document:

```
<figure>
  <img src="img_pulpit.jpg" alt="The Pulpit Rock" width="304"
height="228">
</figure>
```

Specify the font size, font face and color of ext:

```
<font size="3" color="red">This is some text!</font>
<font size="2" color="blue">This is some text!</font>
<font face="verdana" color="green">This is some text!</font>
```

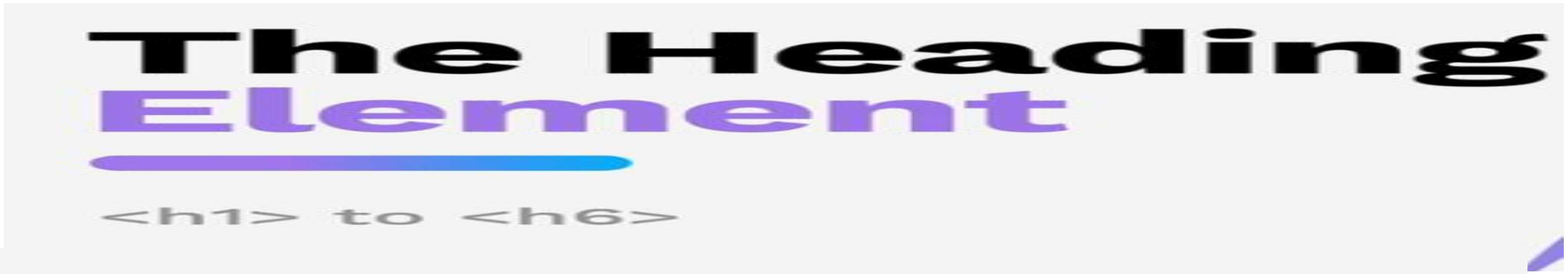
A footer section in a document:

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">
  someone@example.com</a>.</p>
</footer>
```

HTML code syntax Structure

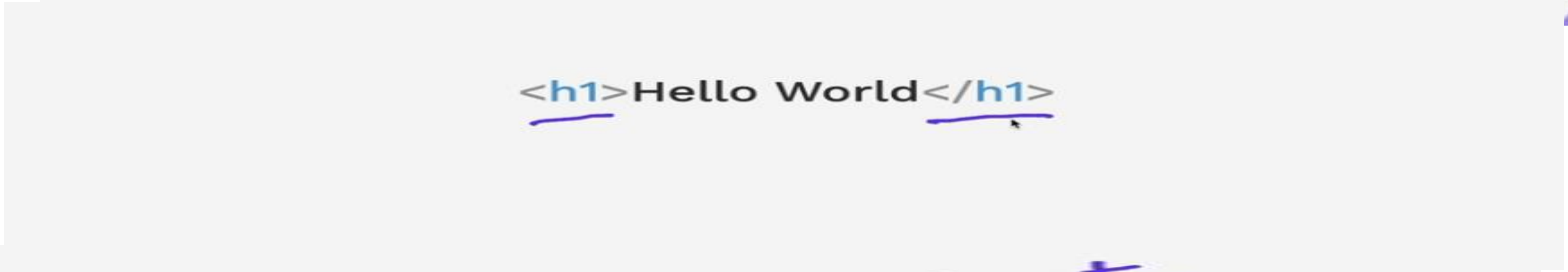
Elements, Contents and Attributes

File Path

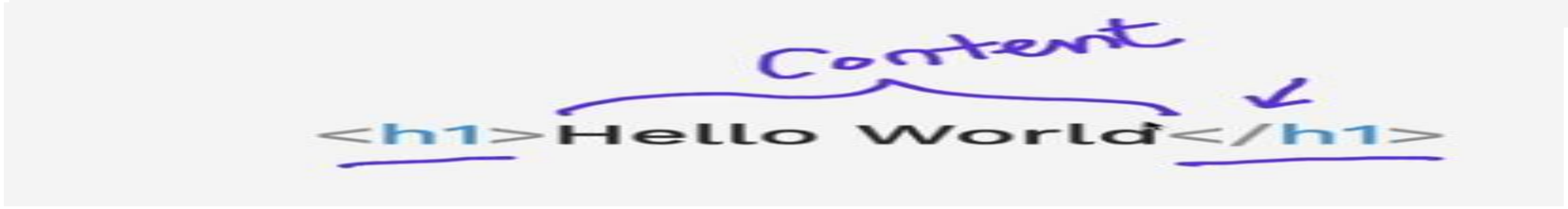


The Heading Element

<h1> to <h6>

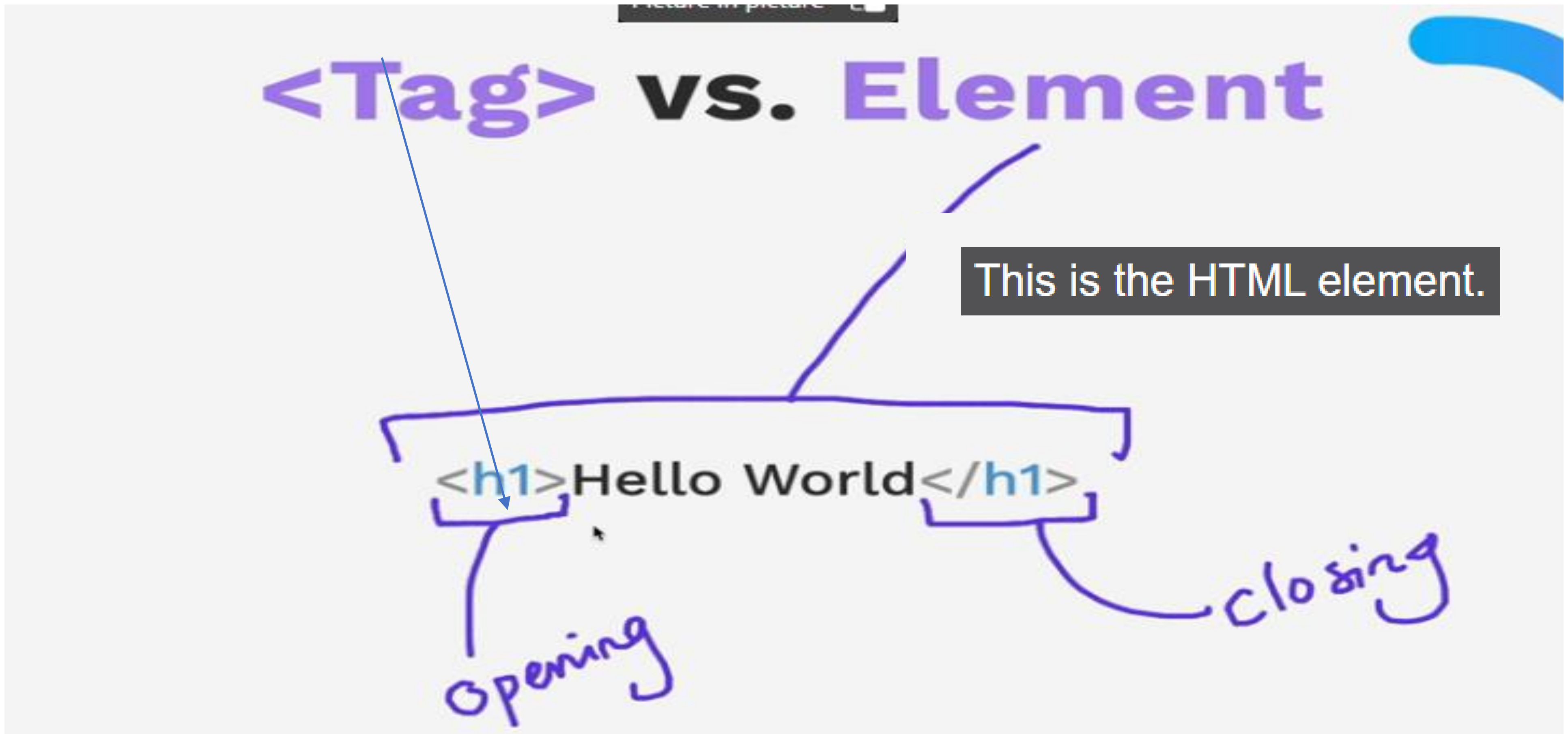


<h1>Hello World</h1>



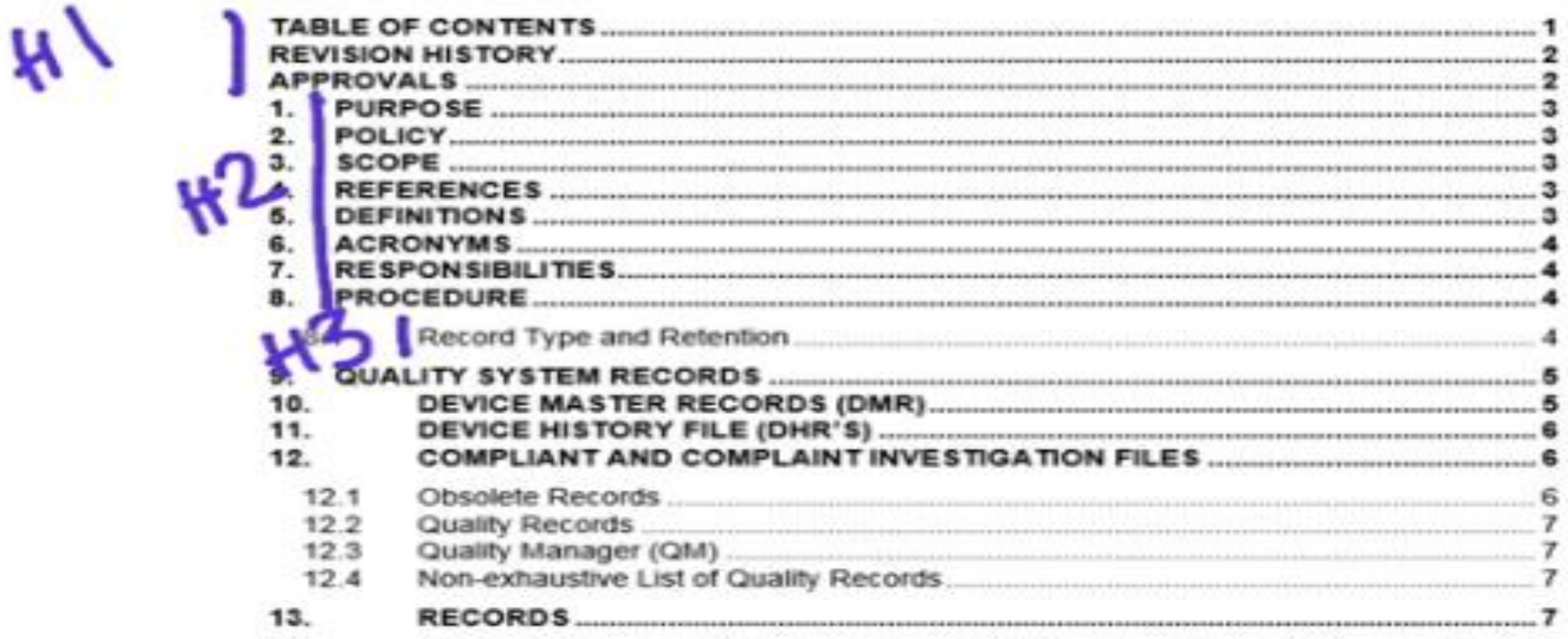
<h1>Hello World</h1>

It starts with opening tag <h1>   and end with closing Tag </h1> in this example



<Tag> vs. Element
This is the HTML element.
<h1>Hello World</h1>
opening
closing

Section Headings



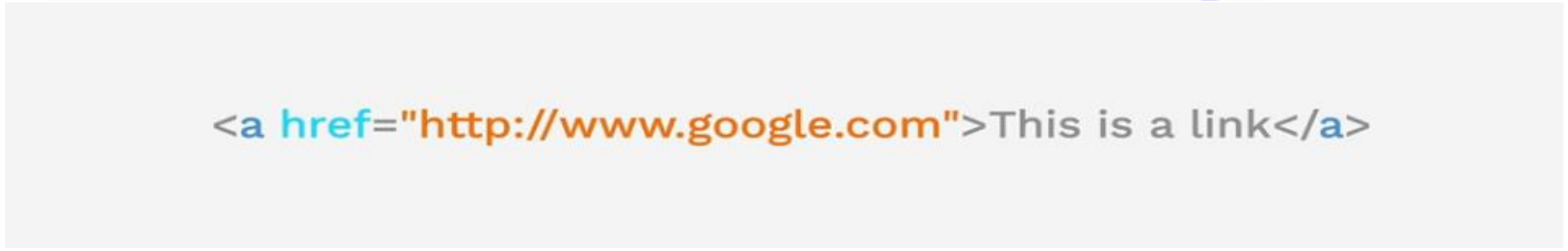
H1

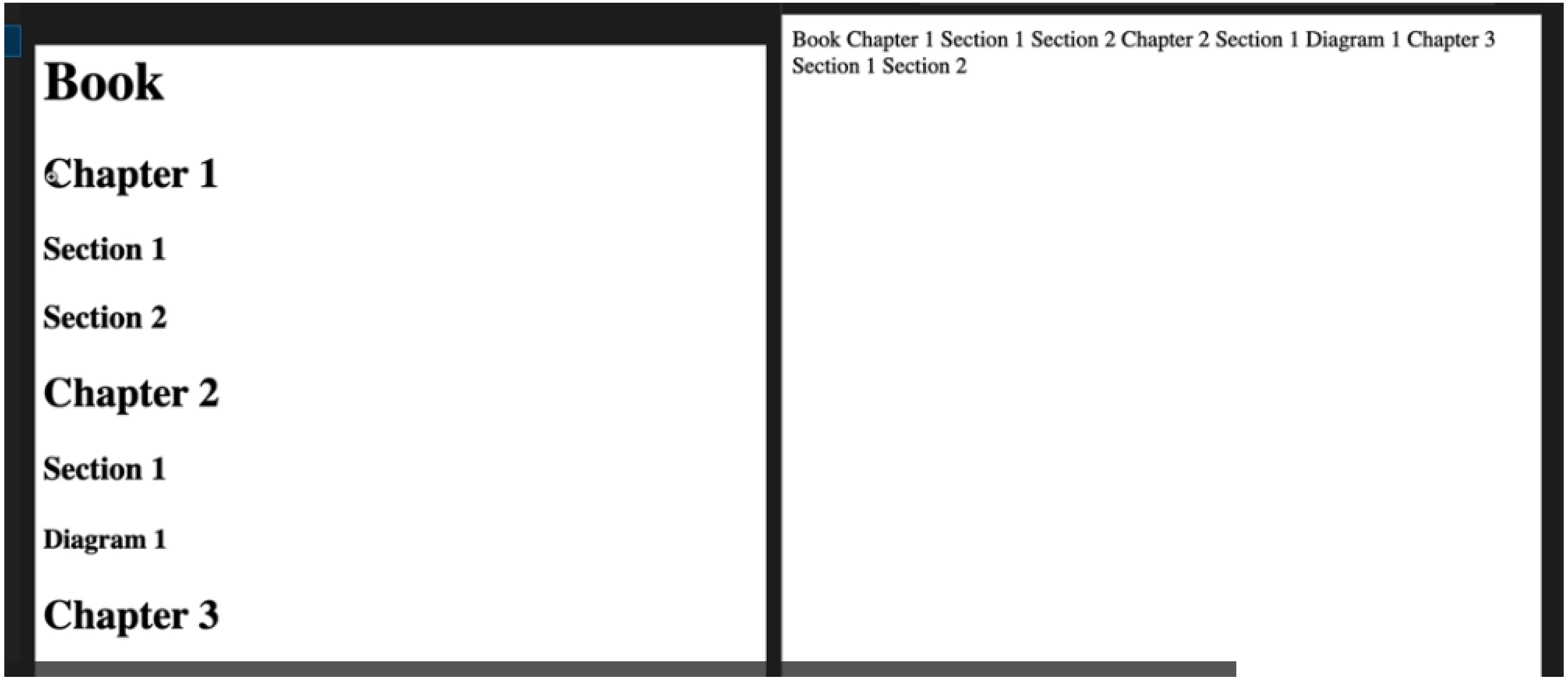
H2

H3

The Anchor Element

Understanding HTML Attributes


```html
<a href="http://www.google.com">This is a link</a>
```



Book

Chapter 1

Section 1

Section 2

Chapter 2

Section 1

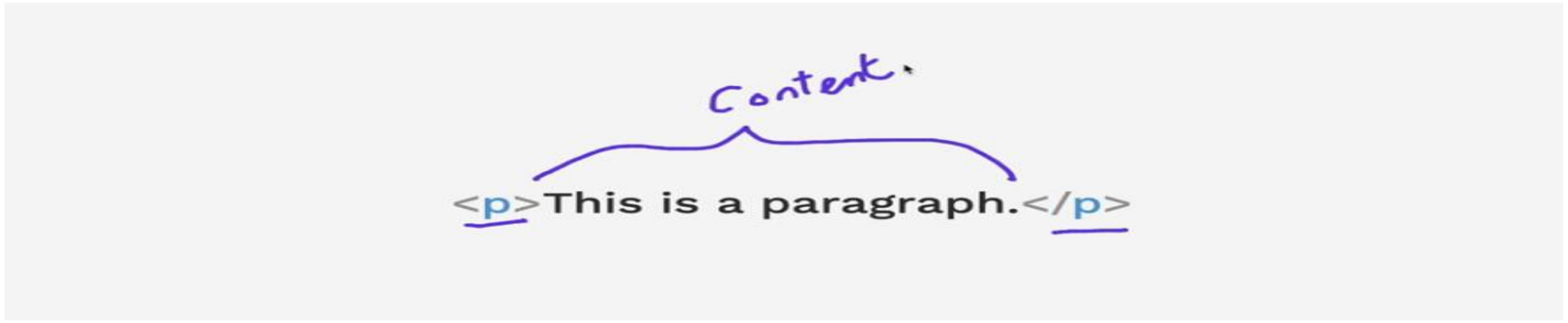
Diagram 1

Chapter 3

```
Book Chapter 1 Section 1 Section 2 Chapter 2 Section 1 Diagram 1 Chapter 3
Section 1 Section 2
```

The Paragraph Element

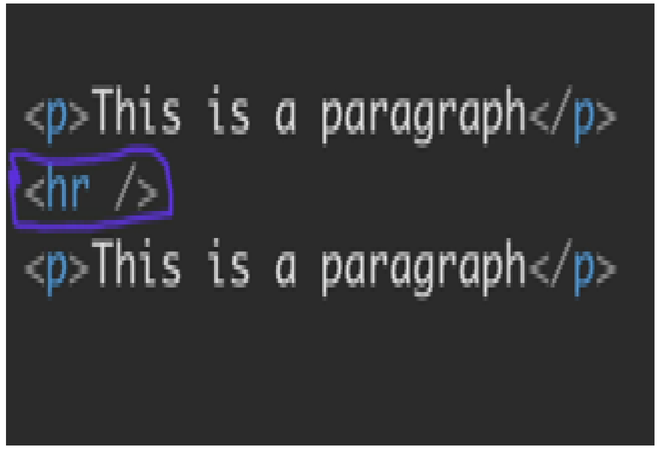
<p>



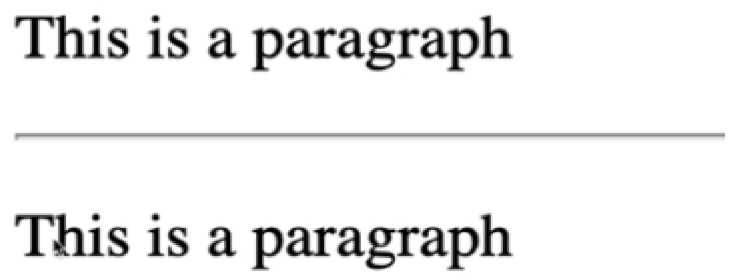
$\underbrace{\texttt{<p>}}\overbrace{\texttt{This is a paragraph.}}^{\text{Content.}}\underbrace{\texttt{</p>}}$

The paragraph will be separated with a line between when we use the <p> tag.

The Void elements such as horizontal rule you are forbidden to put any text inside the tag, Hr will generate a rulers that will divide the contents.

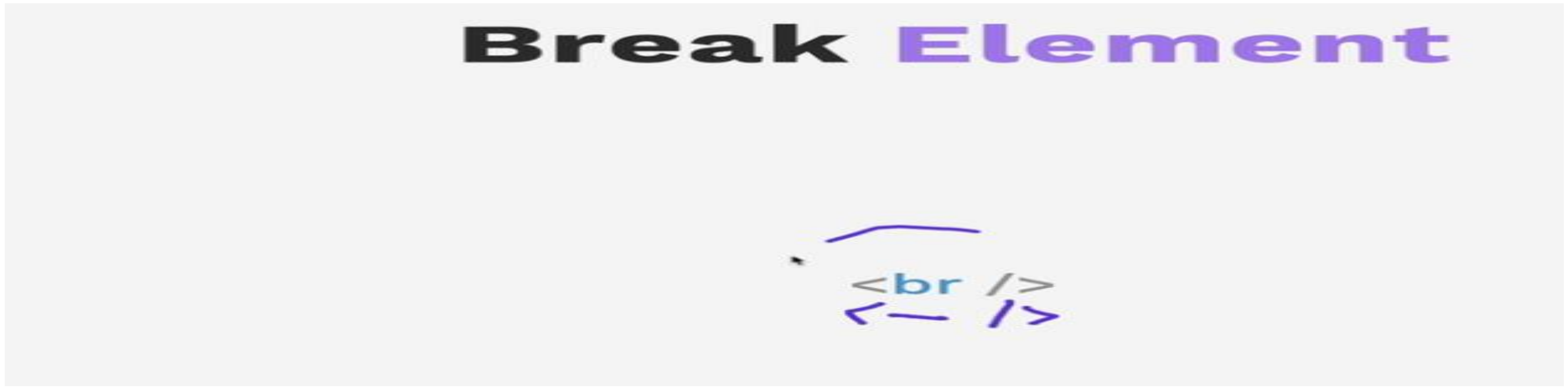


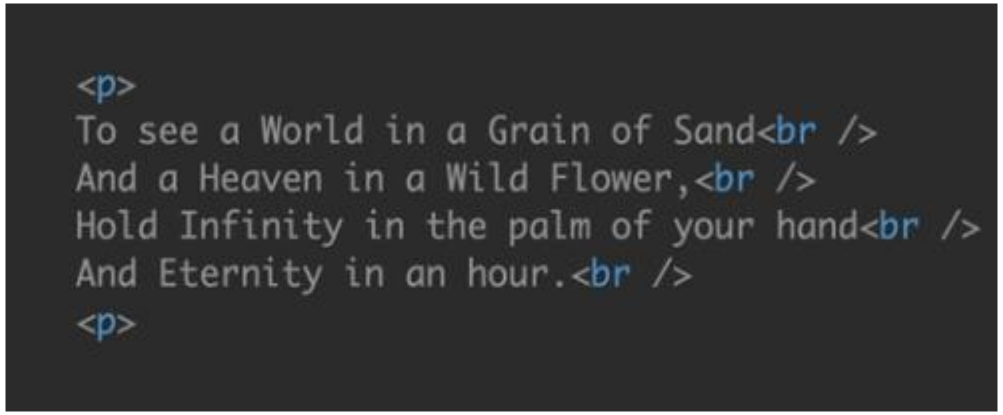
```
<p>This is a paragraph</p>
<hr />
<p>This is a paragraph</p>
```

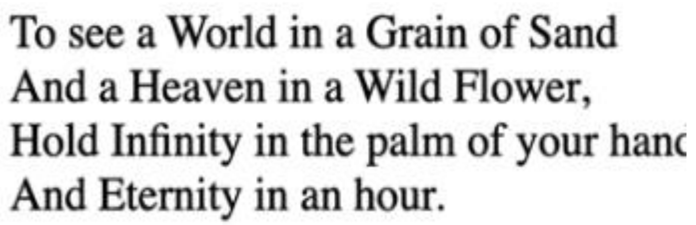


This is a paragraph

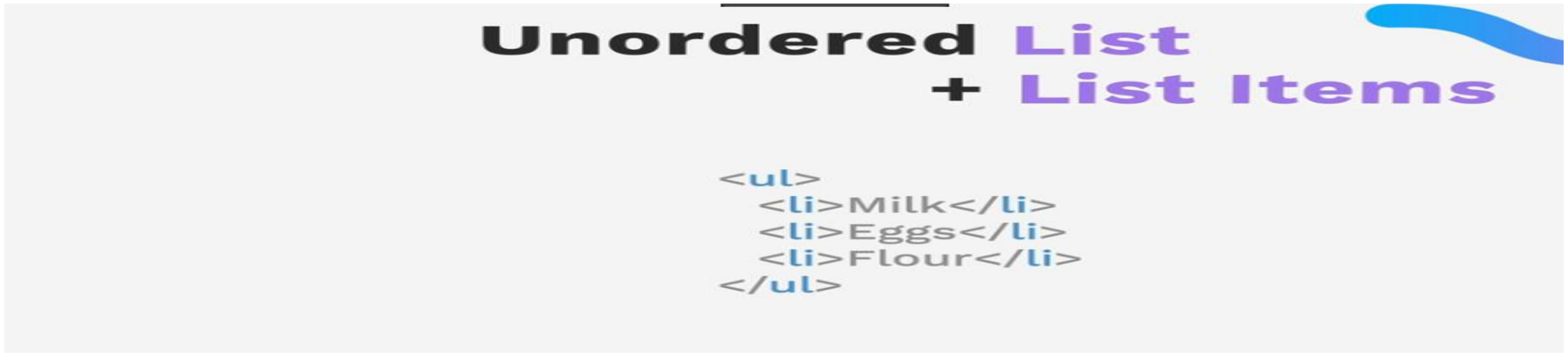
This is a paragraph




```
<p>
To see a World in a Grain of Sand<br />
And a Heaven in a Wild Flower,<br />
Hold Infinity in the palm of your hand<br />
And Eternity in an hour.<br />
<p>
```

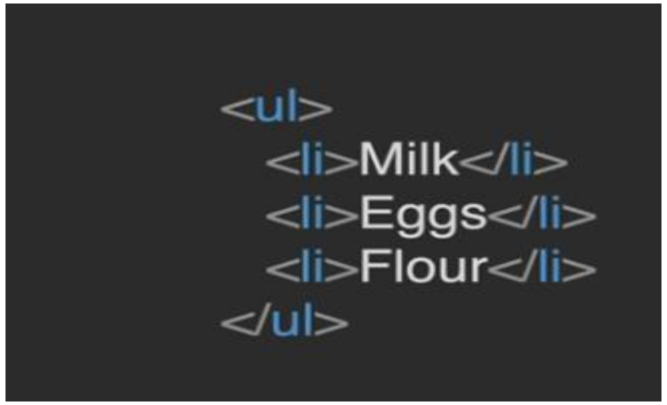

To see a World in a Grain of Sand
And a Heaven in a Wild Flower,
Hold Infinity in the palm of your hanc
And Eternity in an hour.

When the HTML is rendered, this is what you see



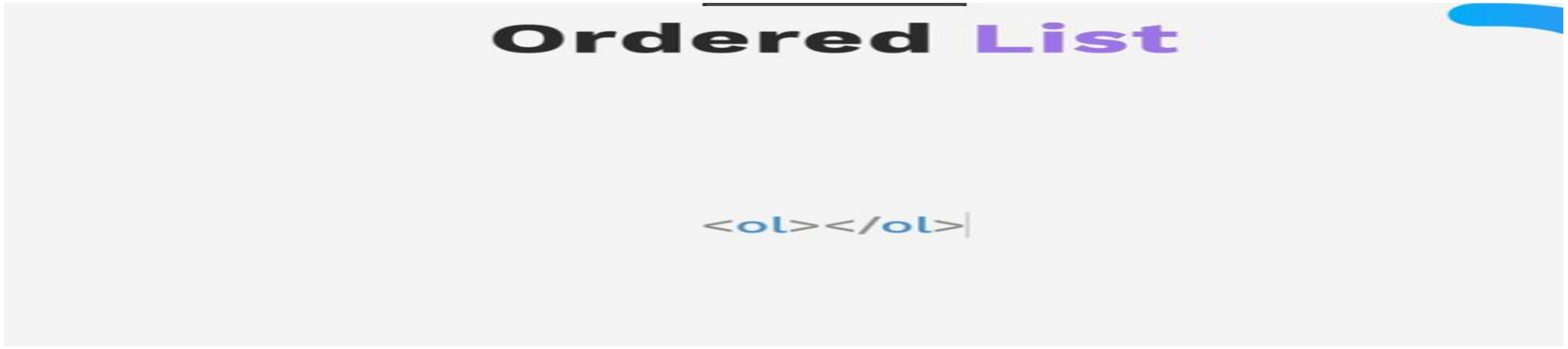
Unordered List + List Items

```
<ul>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Flour</li>
</ul>
```



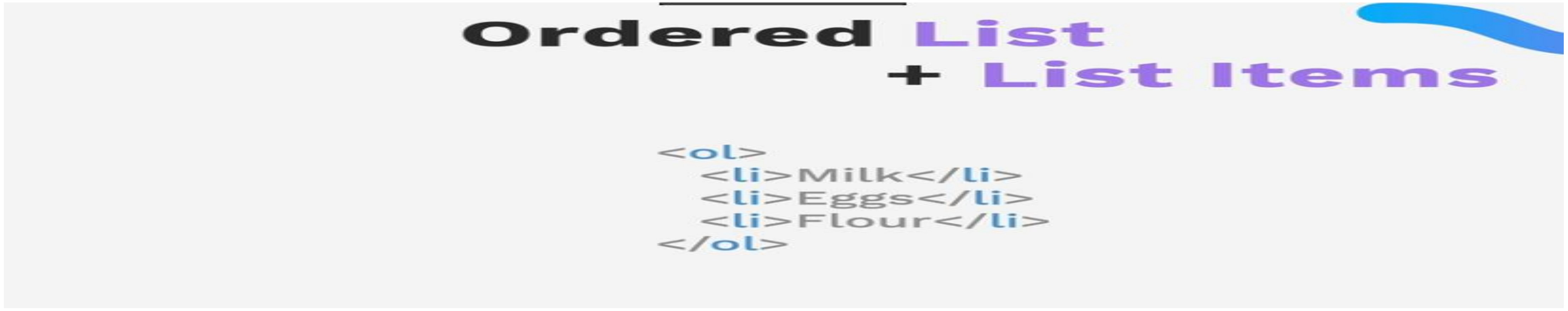
```
<ul>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Flour</li>
</ul>
```

- Milk
- Eggs
- Flour



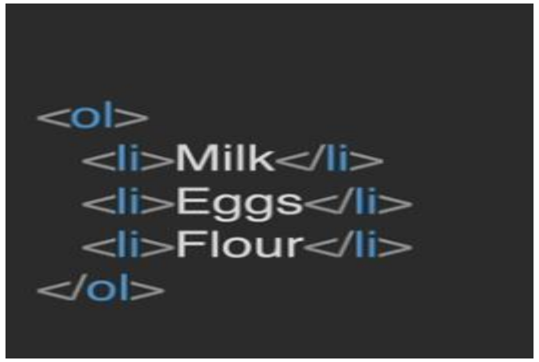
Ordered List

```
<ol></ol>
```

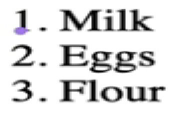


Ordered List + List Items

```
<ol>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Flour</li>
</ol>
```

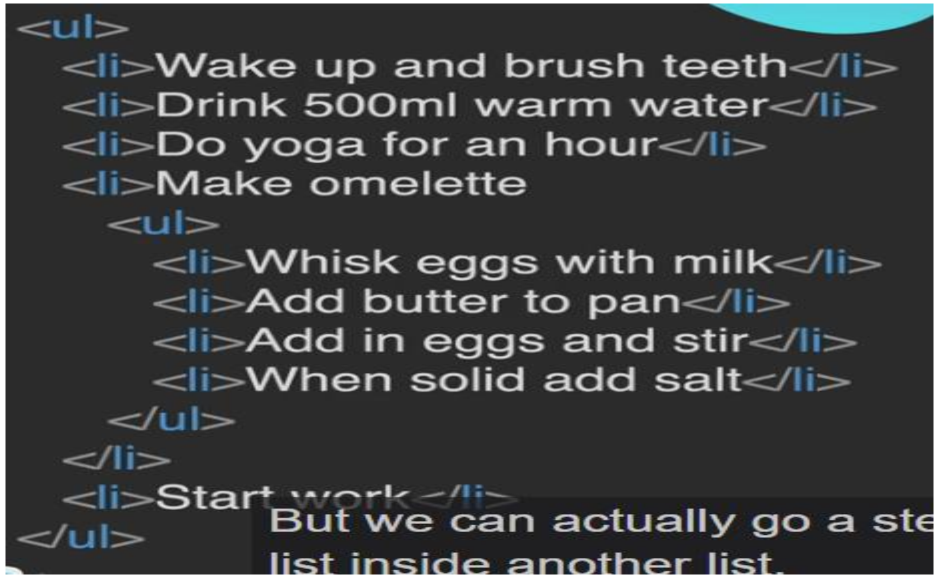


```
<ol>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Flour</li>
</ol>
```



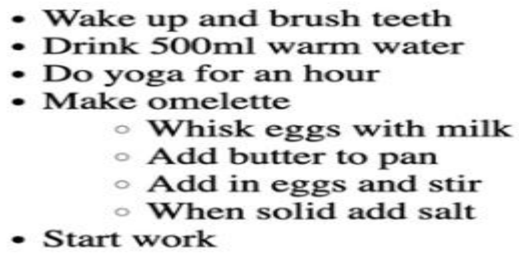
```
1. Milk
2. Eggs
3. Flour
```

We can nest   different tags as in the example   More complicated



```
<ul>
  <li>Wake up and brush teeth</li>
  <li>Drink 500ml warm water</li>
  <li>Do yoga for an hour</li>
  <li>Make omelette
    <ul>
      <li>Whisk eggs with milk</li>
      <li>Add butter to pan</li>
      <li>Add in eggs and stir</li>
      <li>When solid add salt</li>
    </ul>
  </li>
  <li>Start work</li>
</ul>
```

But we can actually go a ste
list inside another list.

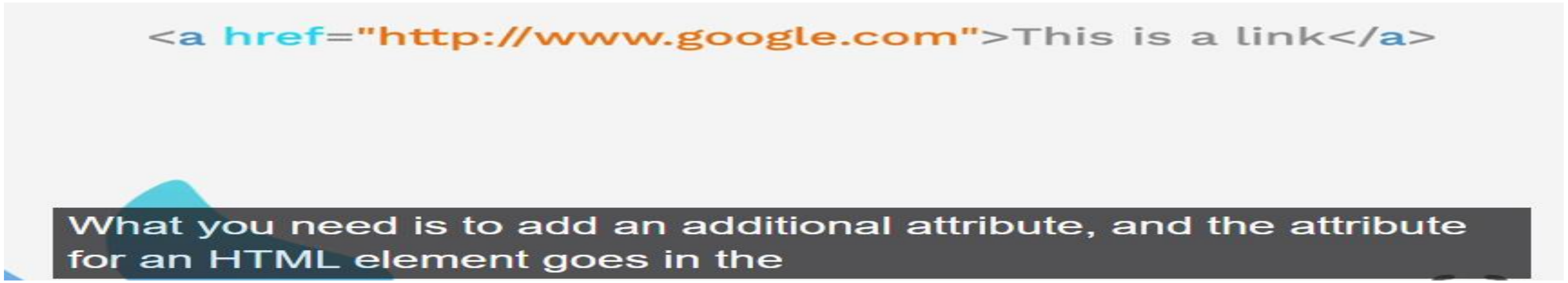


```
• Wake up and brush teeth
• Drink 500ml warm water
• Do yoga for an hour
• Make omelette
      ◦ Whisk eggs with milk
      ◦ Add butter to pan
      ◦ Add in eggs and stir
      ◦ When solid add salt
• Start work
```

The Anchor Element

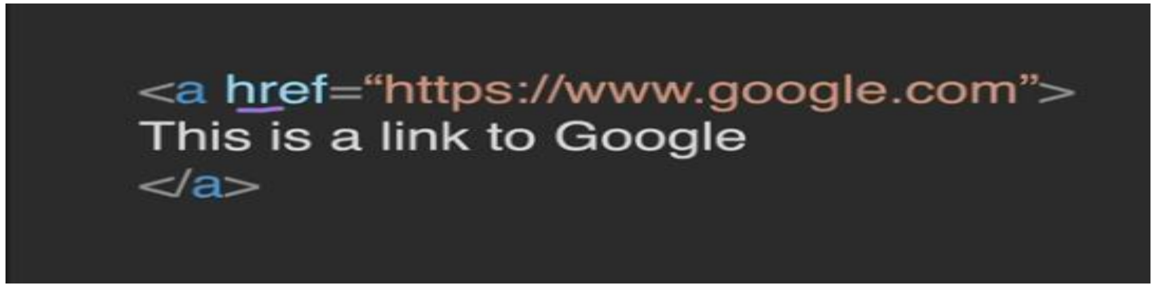
Understanding HTML Attributes

create hyperlinks.



```html
<a href="http://www.google.com">This is a link</a>
```

What you need is to add an additional attribute, and the attribute for an HTML element goes in the



```html
<a href="https://www.google.com">
This is a link to Google
</a>
```

This is a link to Google

The image Element

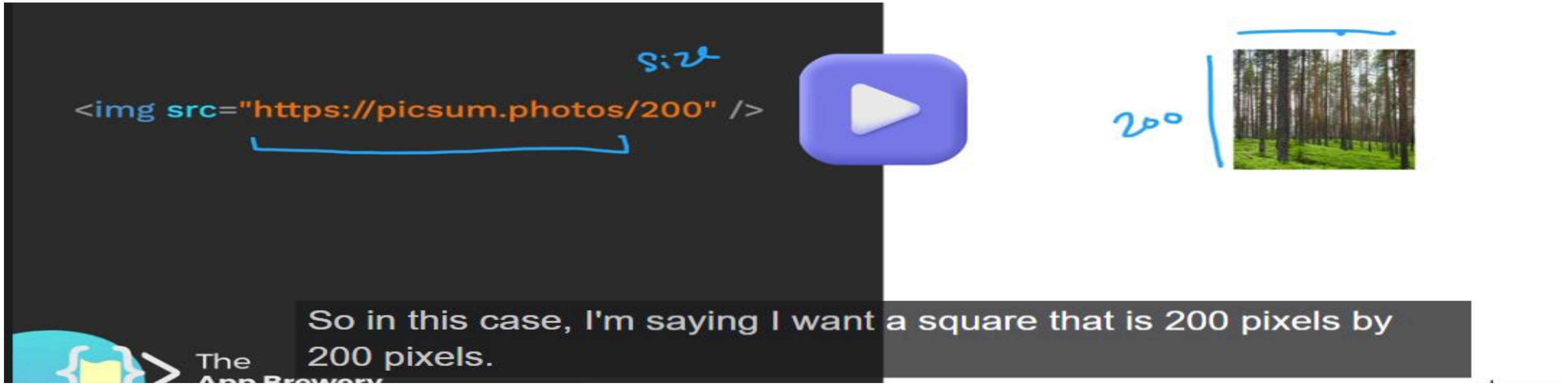
Adding images to our websites

```
<img src="url" />
```

location of image

```
<img src="url" />
```

Source of the image

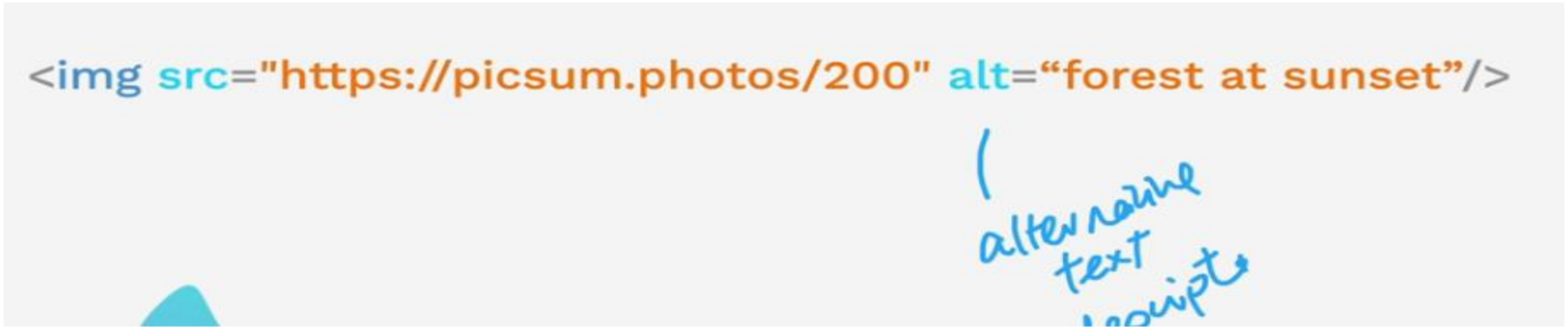


size

```html
<img src="https://picsum.photos/200" />
```

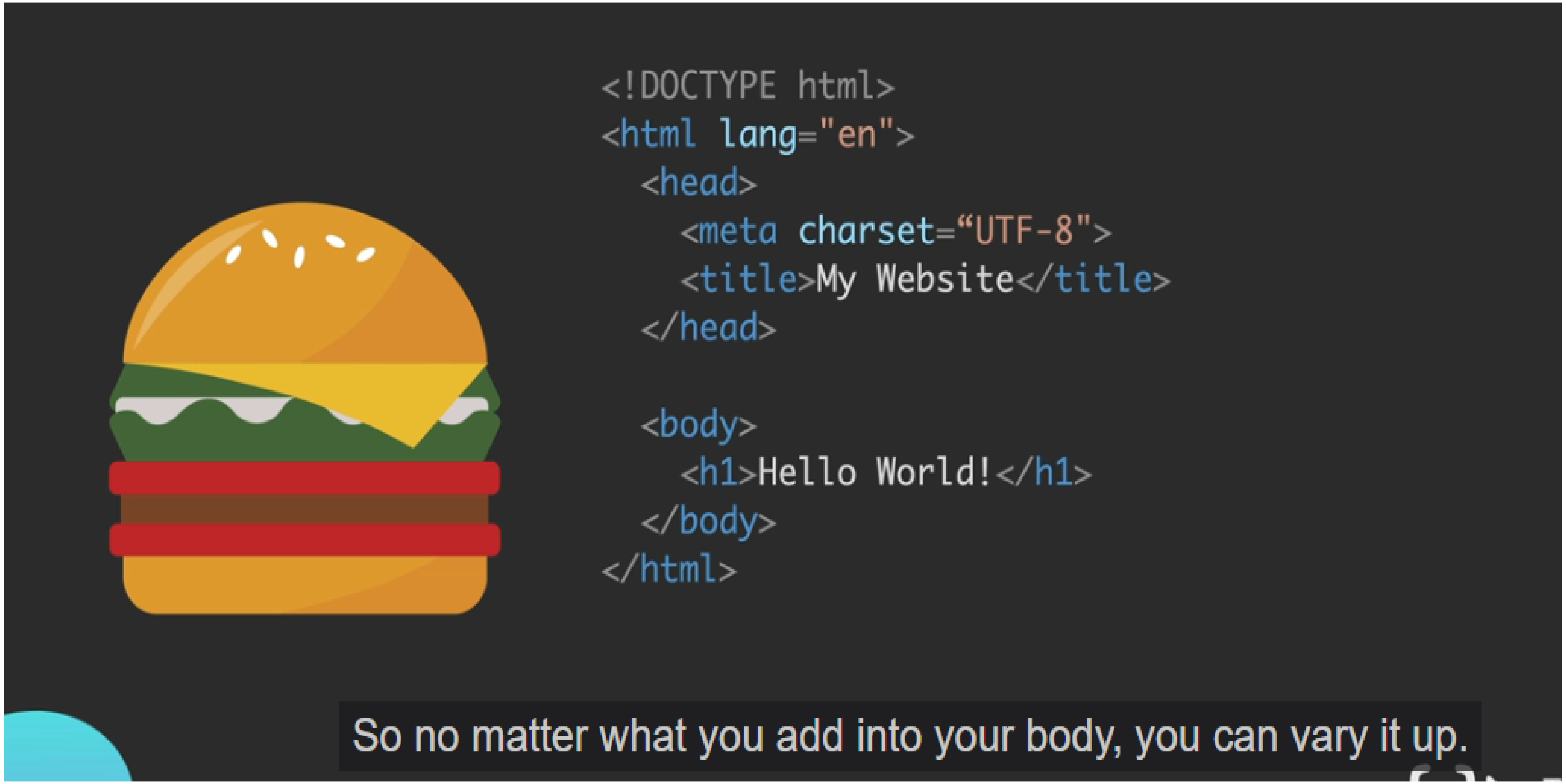
200

So in this case, I'm saying I want a square that is 200 pixels by 200 pixels.



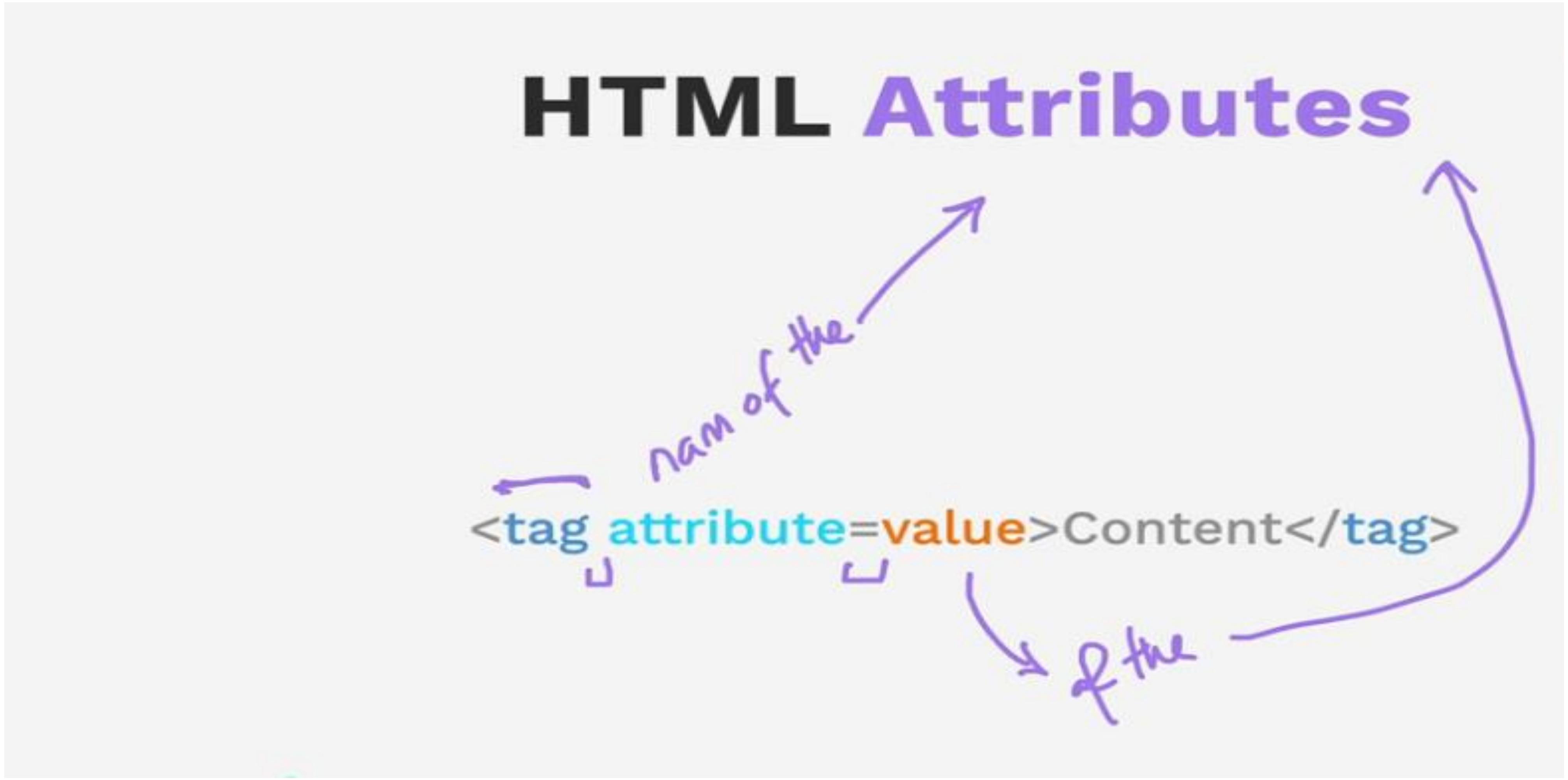
```html
<img src="https://picsum.photos/200" alt="forest at sunset"/>
```

alternative text



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>My Website</title>
  </head>

  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

So no matter what you add into your body, you can vary it up.



HTML Attributes

nam of the

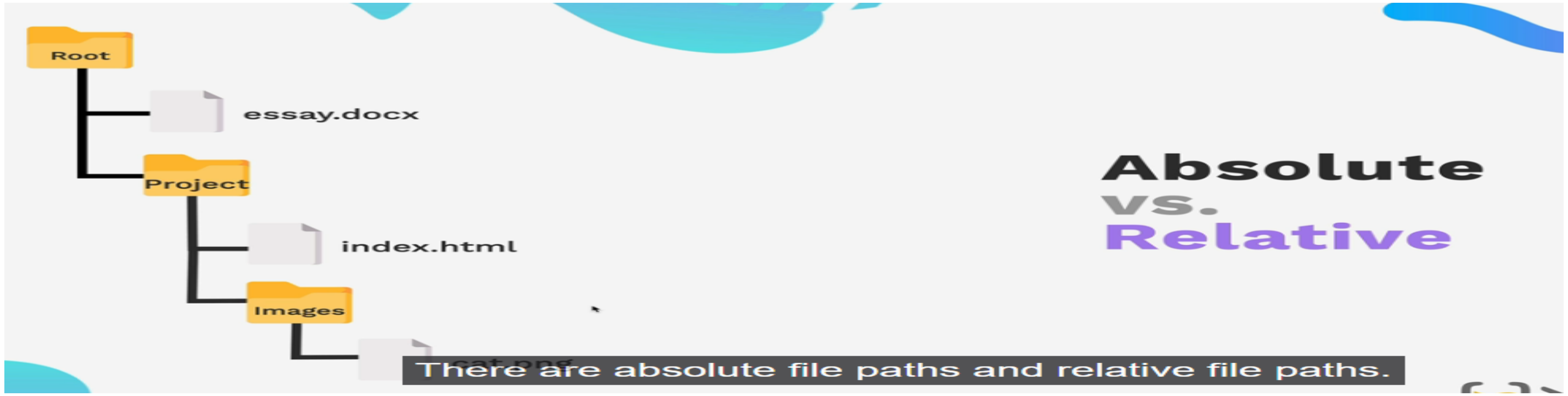
```
<tag attribute=value>Content</tag>
```

f the

File Paths

Absolute and Relative Paths

You can think of a file path as a unique location for a file or folder on your computer.



There are absolute file paths and relative file paths.

Absolute File Path

It's a file path that is relative to the root of the computer.

```
C:/Project/Images/cat.png
```

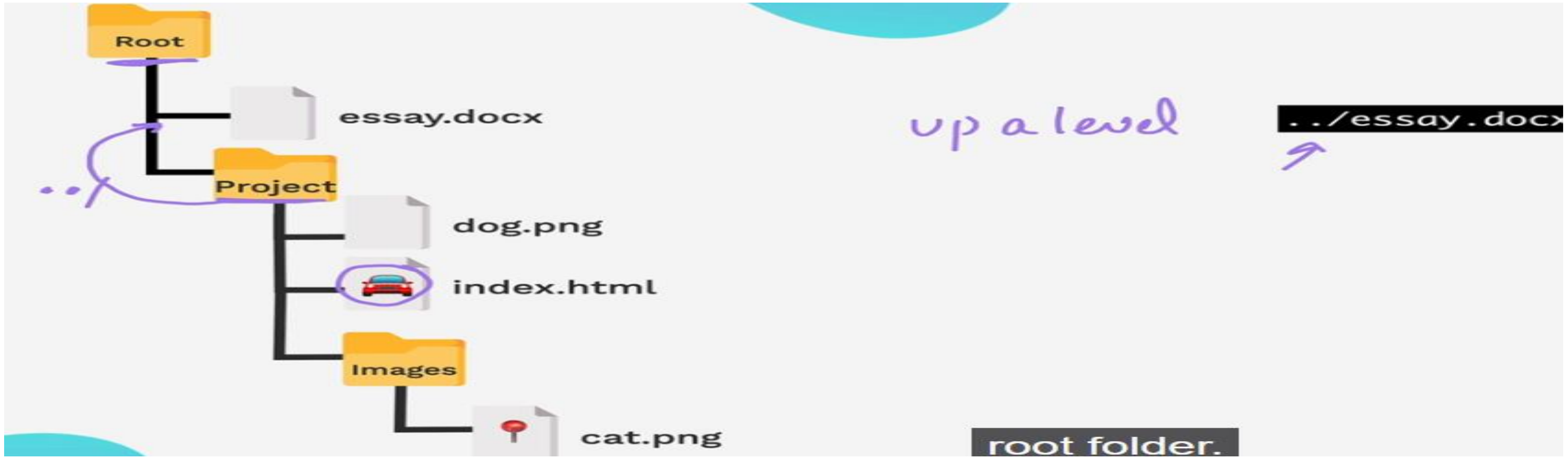
Relative File Path

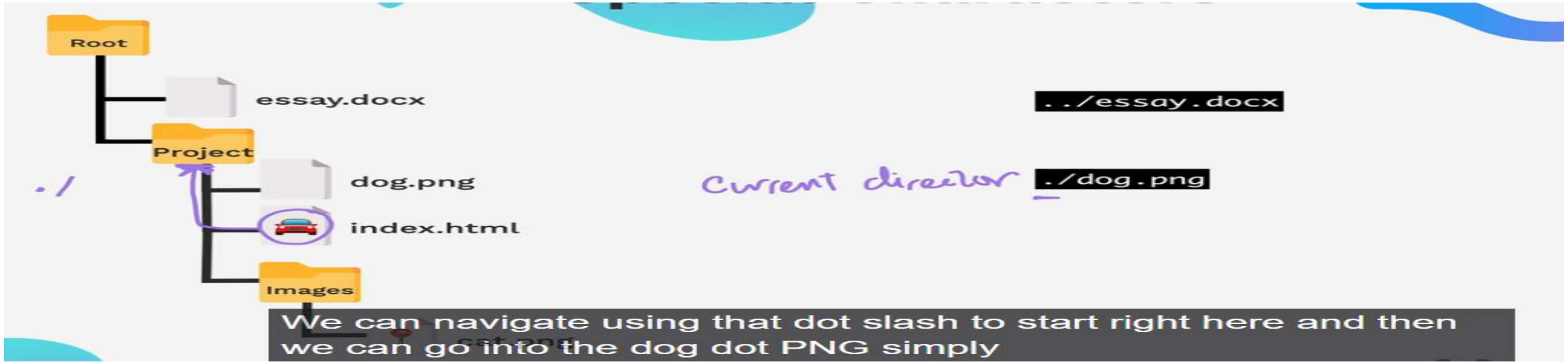
Now, however, for web development, a more useful type of file path is a relative file path, as the

```
../essay.docx
```

relative file path, what it means is to go up a level.

But in this case we can specify it relative to the location where we are writing our code.





We can navigate using that dot slash to start right here and then we can go into the dog dot PNG simply

./ take you to parent ../ take you to grand parent

How to code HTML elements

Two block elements with opening and closing tags

```
<h1>Halloween SuperStore</h1>
<p>Here is a list of links:</p>
```

Two self-closing tags

```
<br>
<img src="logo.gif" alt="Murach Logo">
```

How to code the attributes for HTML elements

How to code an opening tag with attributes

```
<a href="contact.html"
    title="Click to Contact Us"
    class="nav_link">
```

How to code a Boolean attribute

```
<input type="checkbox" name="mailList" checked>
```

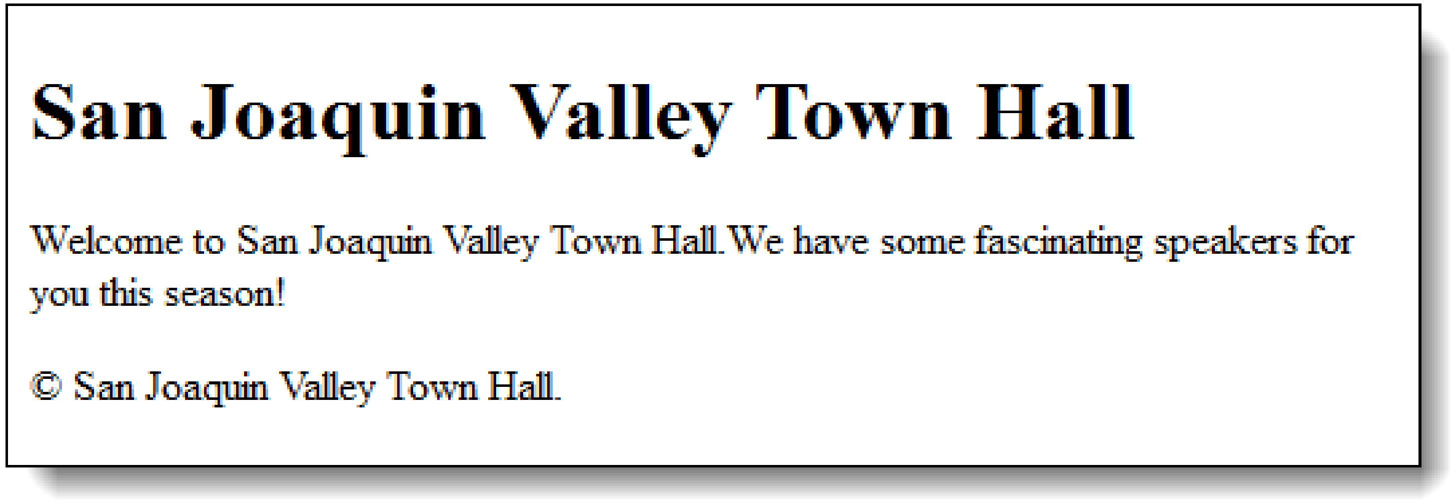
© 2013, Mike Murach & Associates, Inc.

A page that’s structured with header, section, and footer elements

```
<body>
    <header>
        <h1>San Joaquin Valley Town Hall</h1>
    </header>
    <section>
        <p>Welcome to San Joaquin Valley Town Hall. We
           have some fascinating speakers for you this
           season!</p>
    </section>
    <footer>
        <p>&copy; San Joaquin Valley Town Hall.</p>
    </footer>
</body>
```

© 2013, Mike Murach & Associates, Inc.

The page displayed in a web browser



San Joaquin Valley Town Hall

Welcome to San Joaquin Valley Town Hall.We have some fascinating speakers for you this season!

© San Joaquin Valley Town Hall.

The div and span elements

`div`

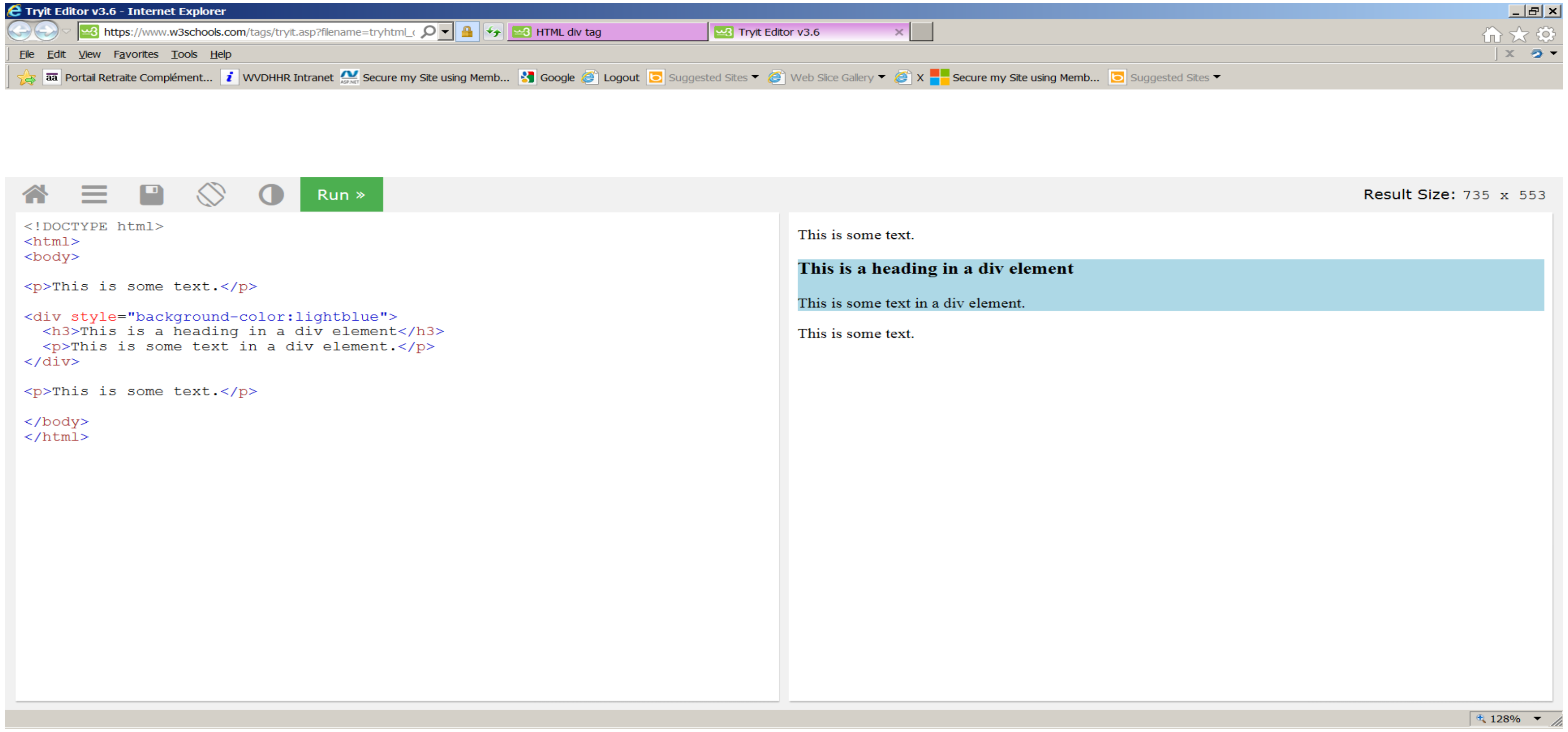
`span`

The <div> tag defines a division or a section in an HTML document.
The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

A <span> element used to color a part of a text:

```
<p>My mother has <span style="color:blue">blue</span> eyes.</p>
```

© 2013, Mike Murach & Associates, Inc.


Tryit Editor v3.6 - Internet Explorer
https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_
HTML div tag
Tryit Editor v3.6
File Edit View Favorites Tools Help
Run »
Result Size: 735 x 553
128%

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

<p>This is some text.</p>

<div style="background-color:lightblue">
  <h3>This is a heading in a div element</h3>
  <p>This is some text in a div element.</p>
</div>

<p>This is some text.</p>

</body>
</html>
```

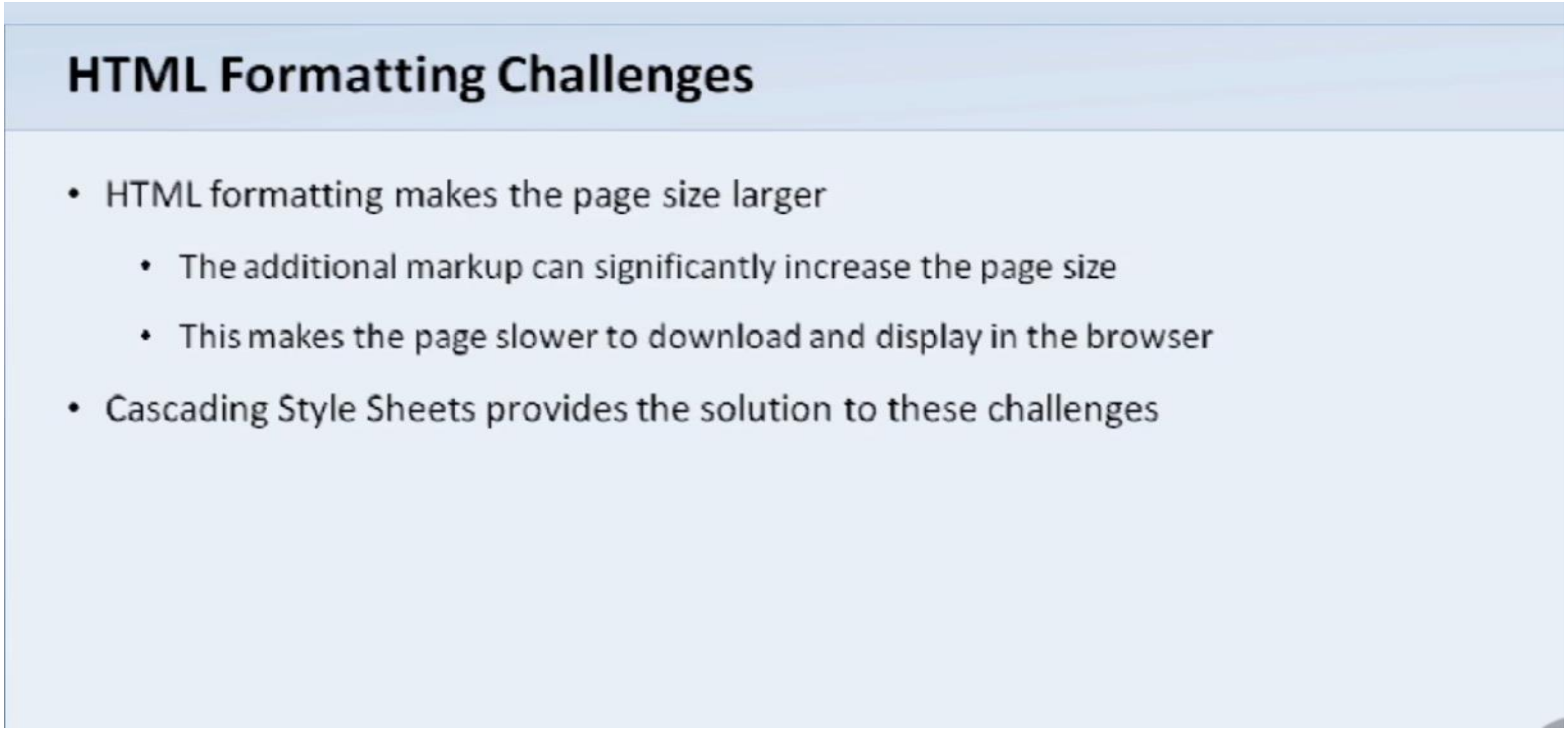
This is some text.

This is a heading in a div element

This is some text in a div element.

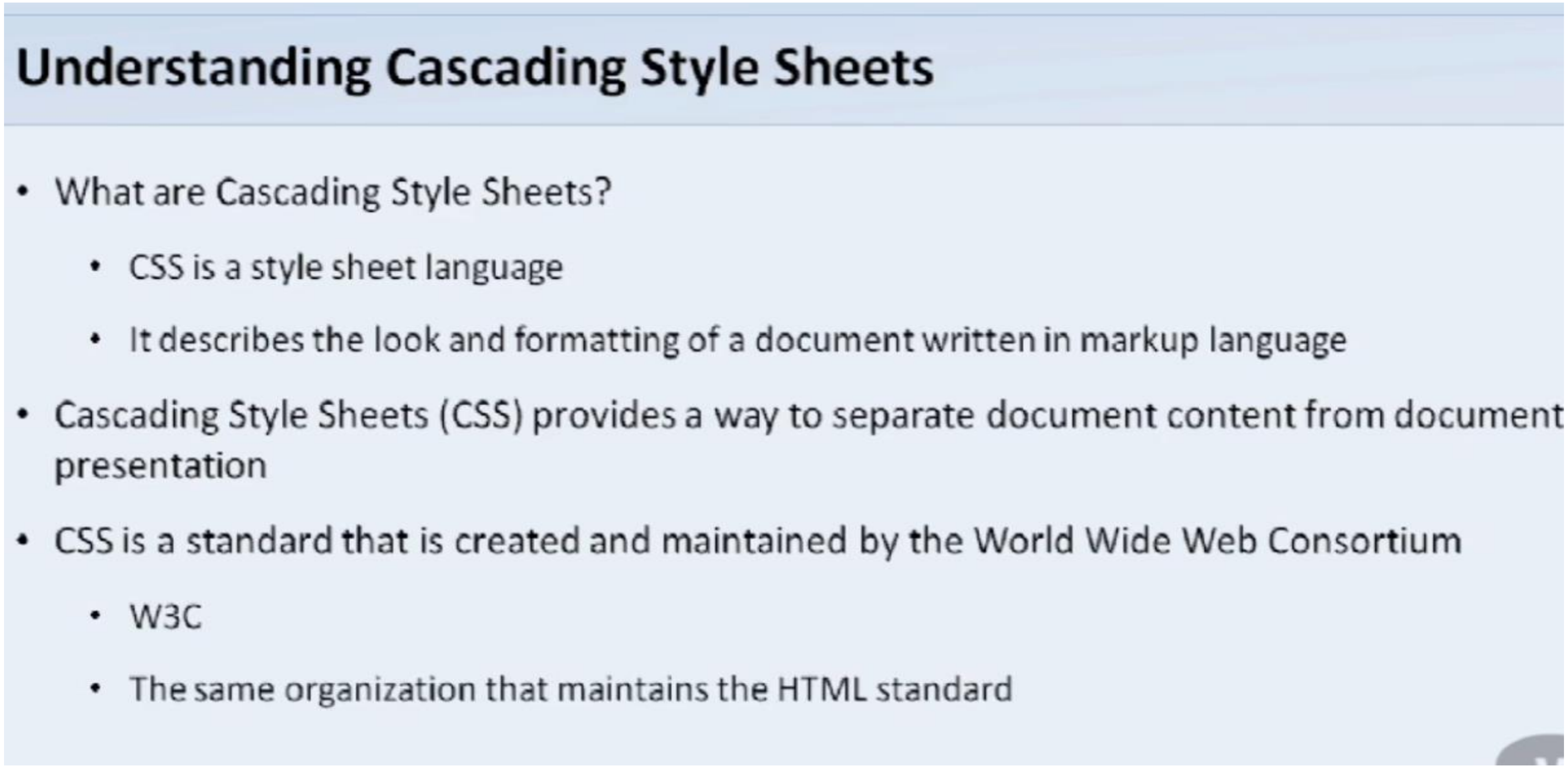
This is some text.

CSS



HTML Formatting Challenges

- HTML formatting makes the page size larger
  - The additional markup can significantly increase the page size
  - This makes the page slower to download and display in the browser
- Cascading Style Sheets provides the solution to these challenges

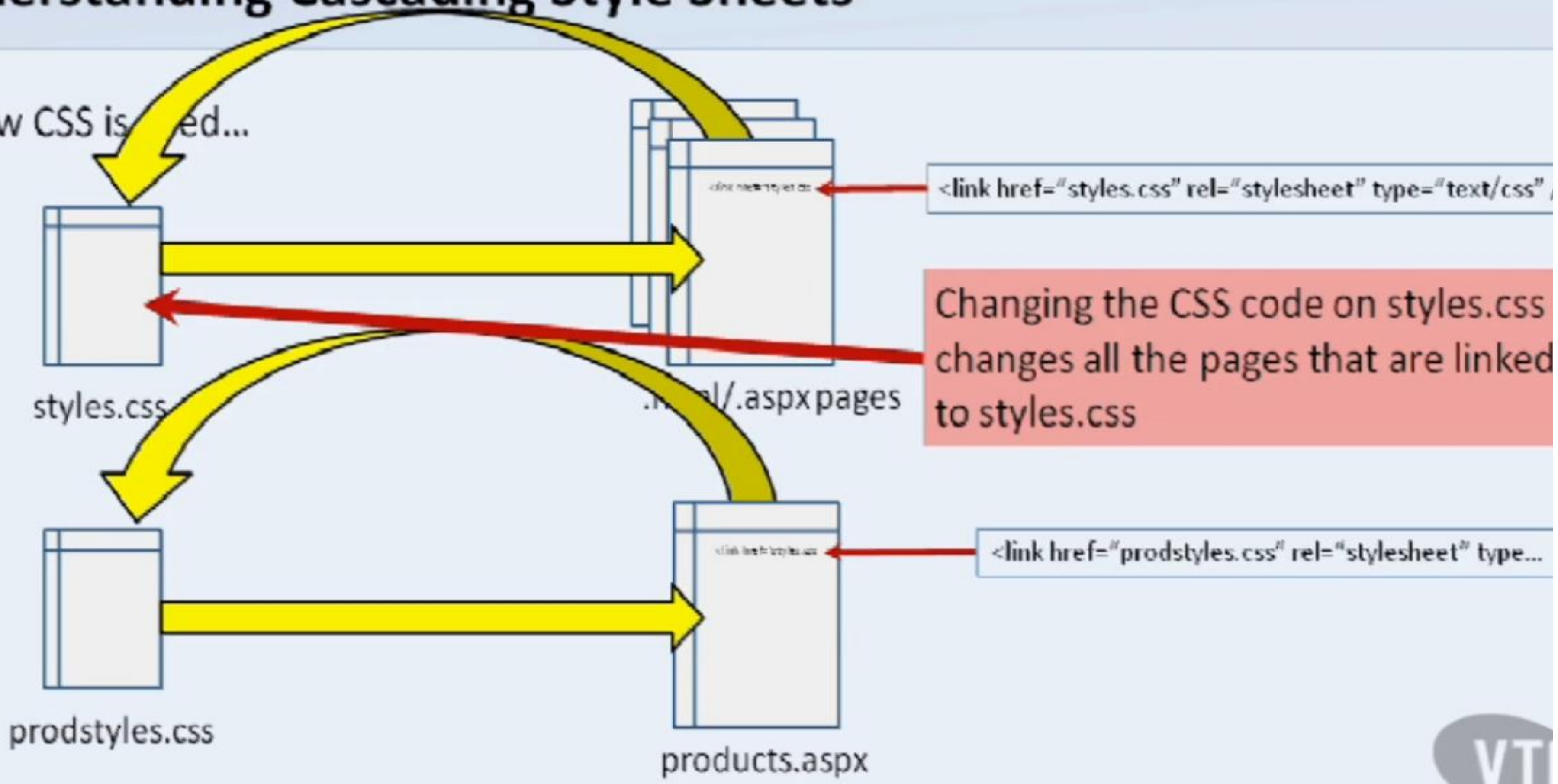


Understanding Cascading Style Sheets

- What are Cascading Style Sheets?
  - CSS is a style sheet language
  - It describes the look and formatting of a document written in markup language
- Cascading Style Sheets (CSS) provides a way to separate document content from document presentation
- CSS is a standard that is created and maintained by the World Wide Web Consortium
  - W3C
  - The same organization that maintains the HTML standard

Understanding Cascading Style Sheets

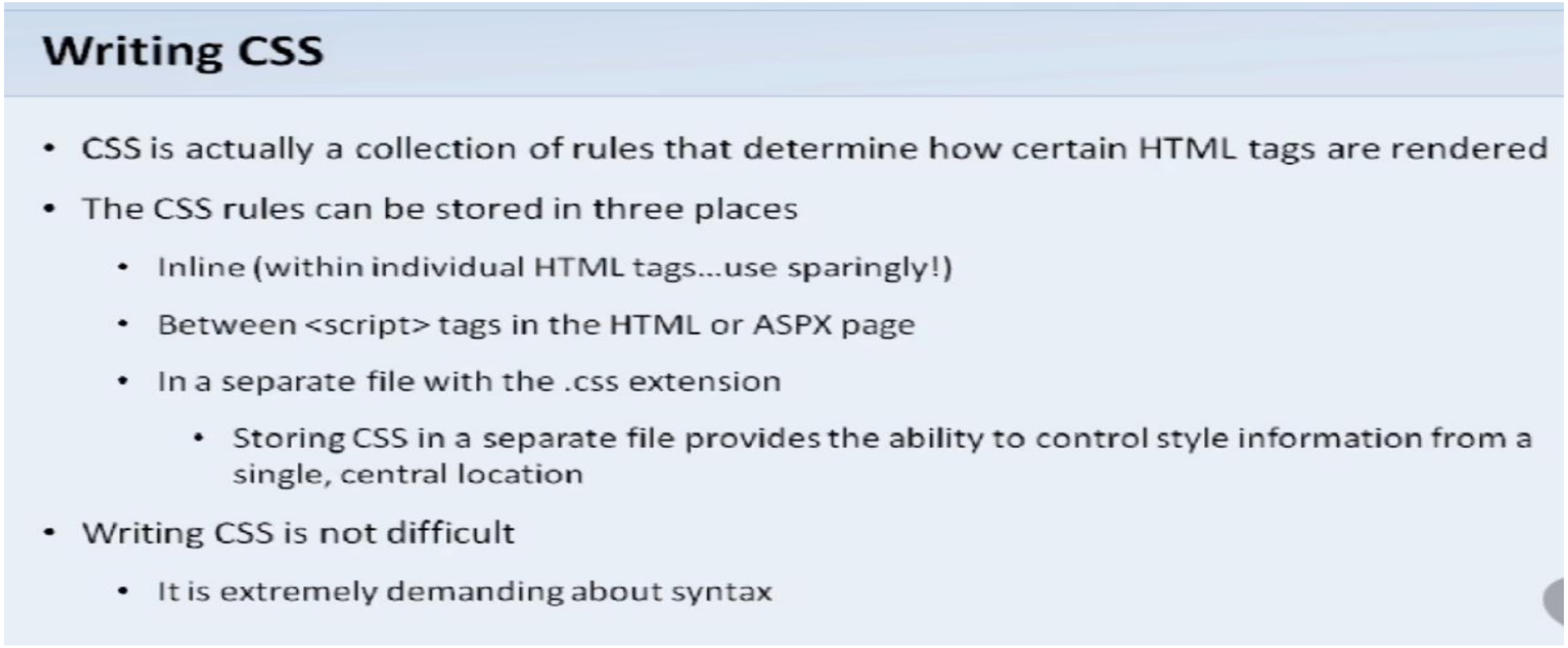
- How CSS is ed...



<link href="styles.css" rel="stylesheet" type="text/css"

Changing the CSS code on styles.css changes all the pages that are linked to styles.css

<link href="prodstyles.css" rel="stylesheet" type...



Writing CSS

- CSS is actually a collection of rules that determine how certain HTML tags are rendered
- The CSS rules can be stored in three places
  - Inline (within individual HTML tags...use sparingly!)
  - Between <script> tags in the HTML or ASPX page
  - In a separate file with the .css extension
    - Storing CSS in a separate file provides the ability to control style information from a single, central location
- Writing CSS is not difficult
  - It is extremely demanding about syntax

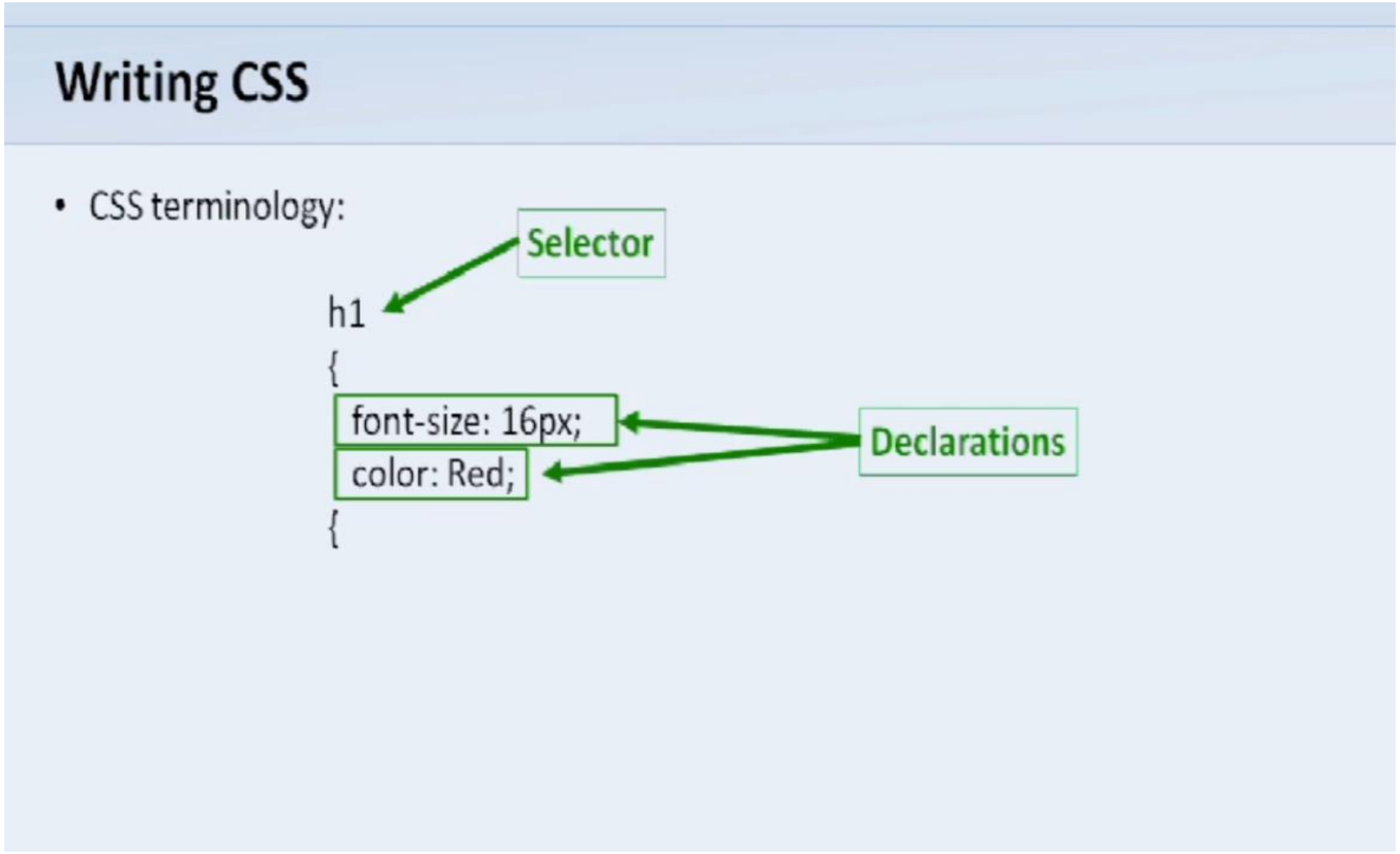
Writing CSS

- The most challenging aspect of CSS is dealing with the differences in which various browsers render it
- CSS coding is based on two main elements
  - Selector
    - Designates the HTML element(s) that will receive the formatting
  - Declaration
    - The styling information for the selector
    - Contained within curly braces {} (there can be multiple declarations)
    - Consists of a property followed by a colon followed by a value and ends with a semicolon

- A CSS rule set consists of a selector and one or more rules within braces. A rule set is called a style rule.
- You code a selector for all elements of a specific type by naming the element. This is element selector.
- Ex: body { width:40px ; }
- Example of elements: header, body, figure, h1, label,…..

- A CSS selector consists of the identifiers that are coded at the beginning of the rule set.

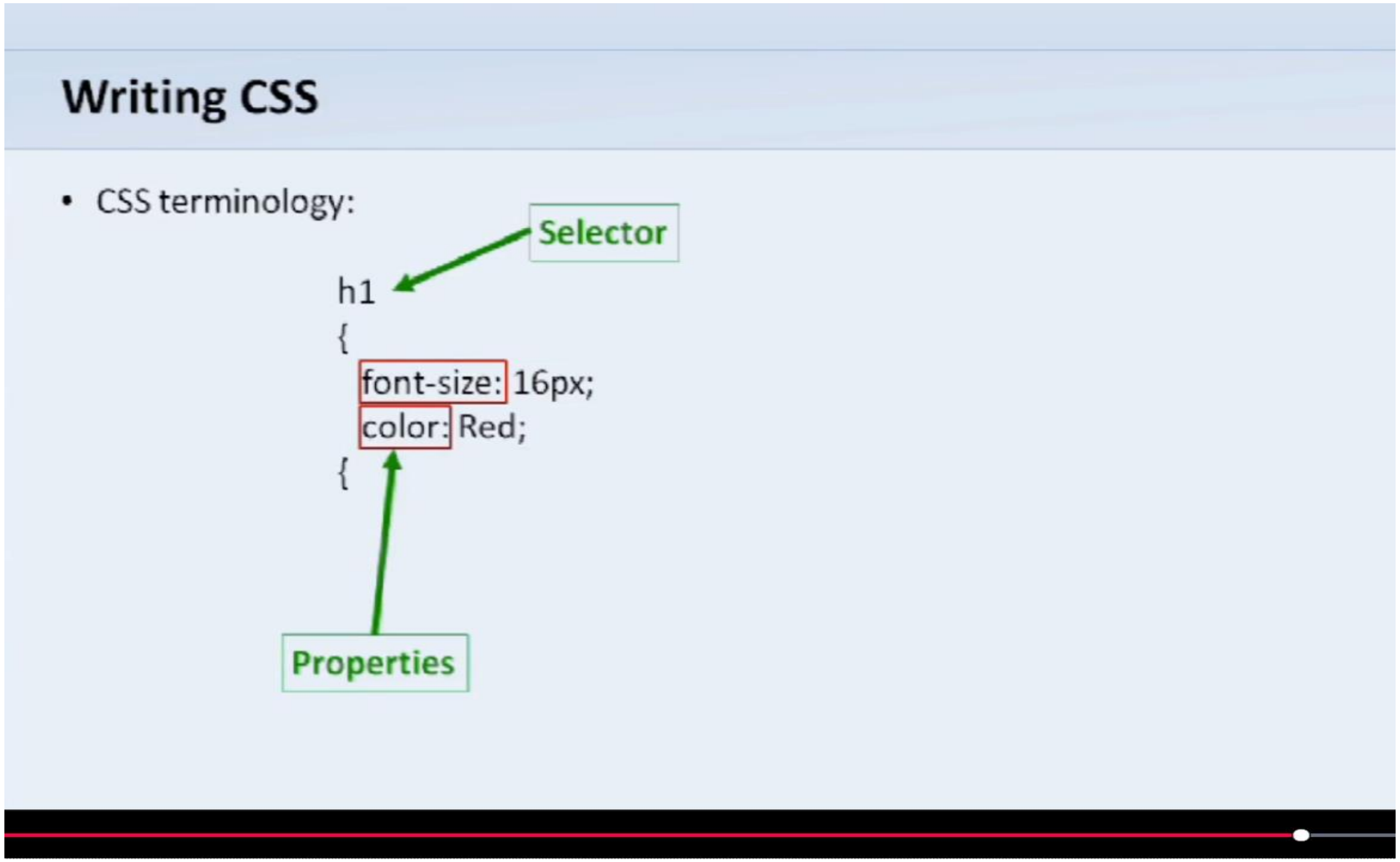
- A CSS rule consists of a property, a colon, a value and a semicolon.
- Ex: label { float:left;   }



Writing CSS

- CSS terminology:

```
h1
{
font-size: 16px;
color: Red;
{
```

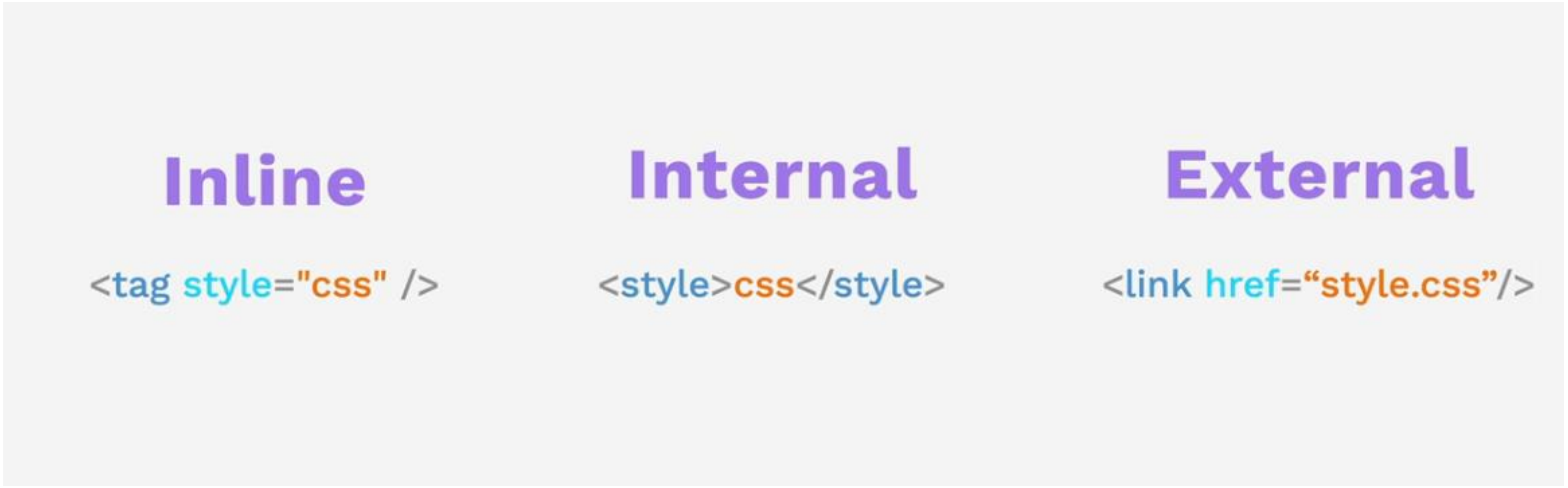


Writing CSS

- CSS terminology:

```
h1
{
  font-size: 16px;
  color: Red;
{
```

Different style in CSS



Inline

```
<tag style="css" />
```

Internal

```
<style>css</style>
```

External

```
<link href="style.css"/>
```

It is a best practice to use external style sheets because it leads to better separation of concerns.

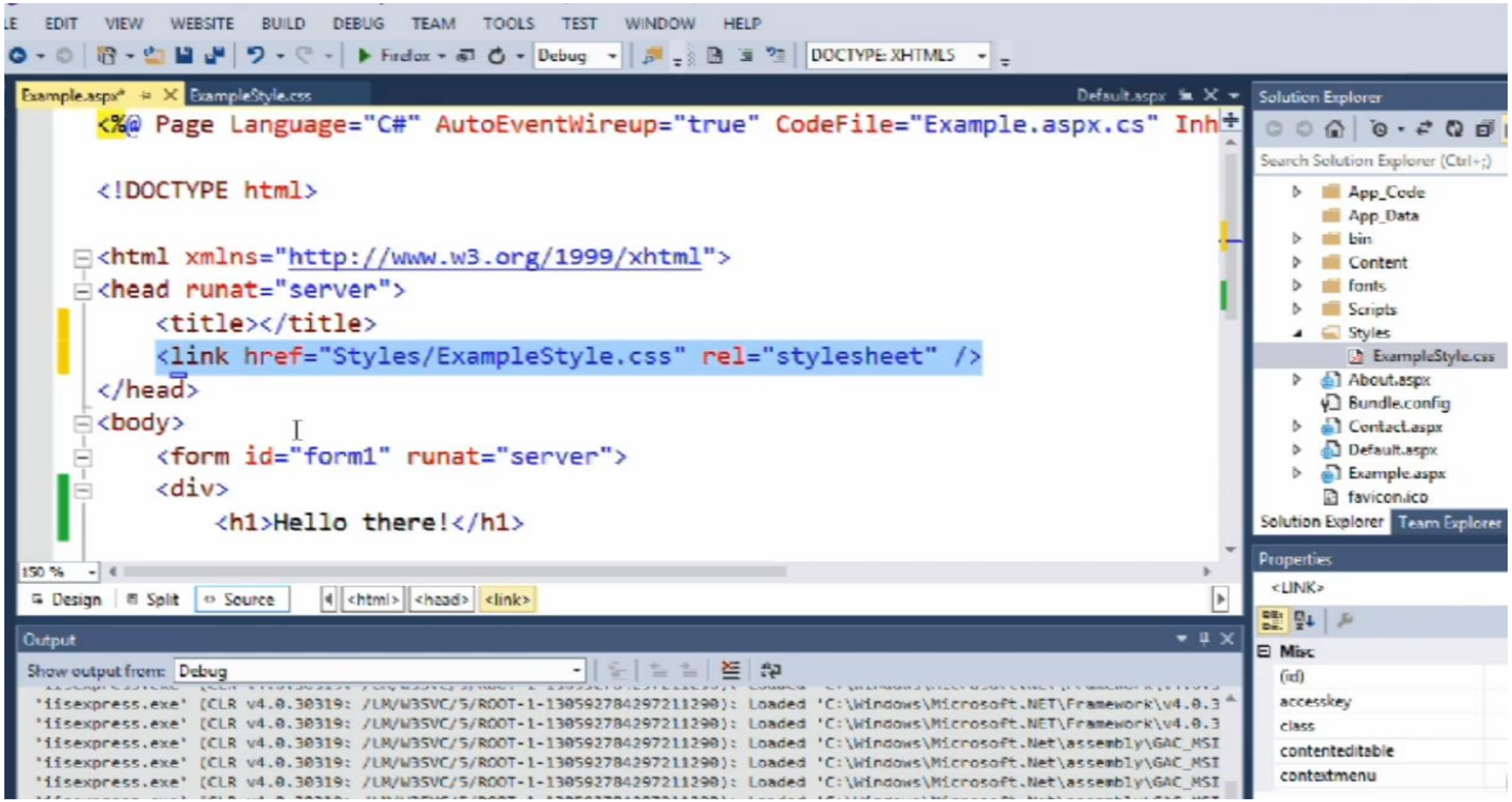
If more than one rule for the same property is applied to the same element the last rule overrides the earlier rules.

It is in the head elements where you can include the link to external style sheet.
You code an Id selector for an element with an I'd attribute by coding a. Pound sign # followed by the Id value.
Ex:

```
#first { margin:0}
```

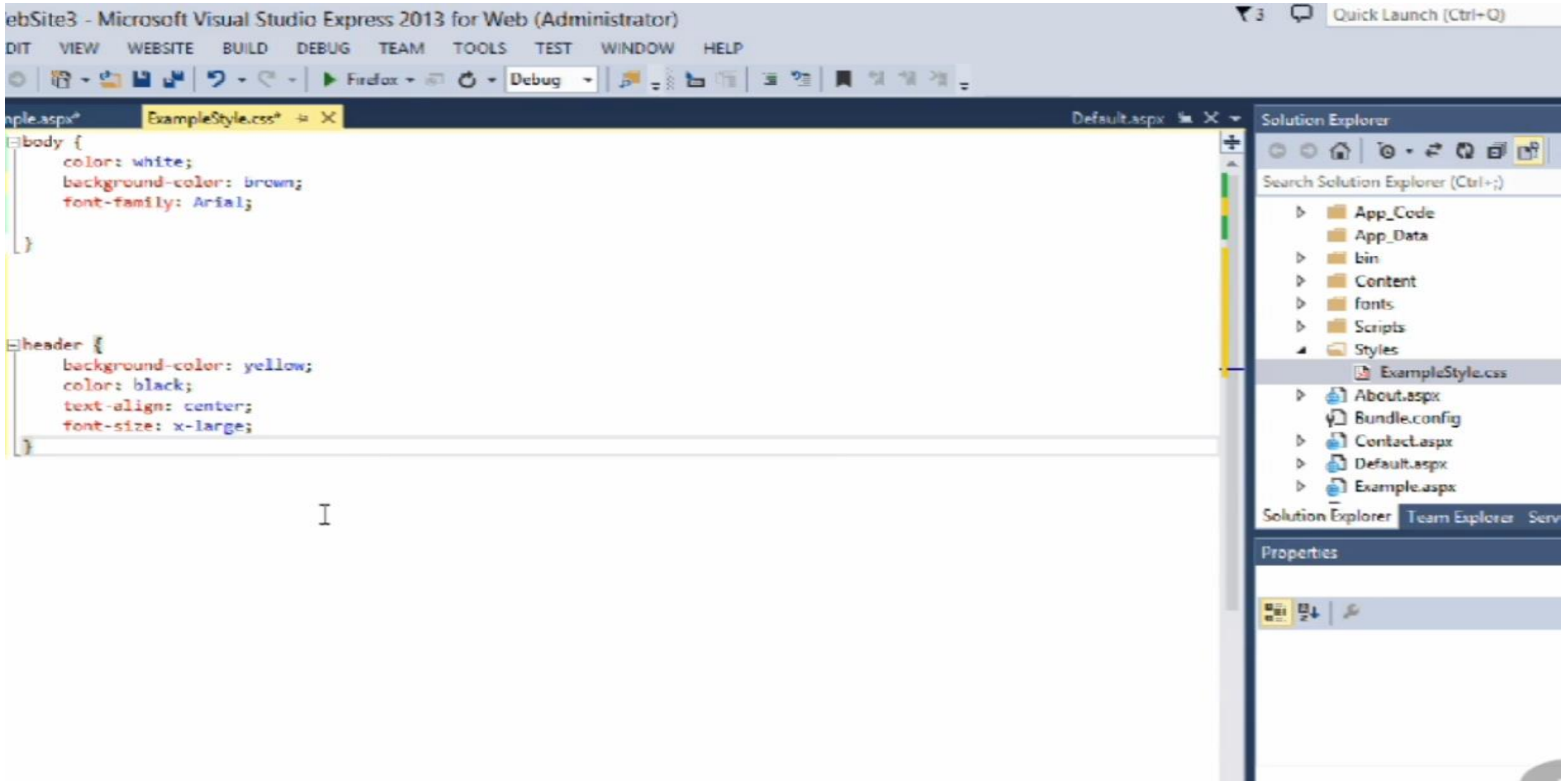
You code a selector for an element with a class attribute by coding a period followed by the class name it is called class selector ex:

```
.classblue   {color:blue}
```



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Example.aspx.cs" Inh

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link href="Styles/ExampleStyle.css" rel="stylesheet" />
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <h1>Hello there!</h1>
```

```
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/5/ROOT-1-130592784297211290): Loaded 'C:\Windows\Microsoft.NET\Framework\v4.0.3
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/5/ROOT-1-130592784297211290): Loaded 'C:\Windows\Microsoft.NET\Framework\v4.0.3
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/5/ROOT-1-130592784297211290): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSI
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/5/ROOT-1-130592784297211290): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSI
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/5/ROOT-1-130592784297211290): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSI
```



```
body {
    color: white;
    background-color: brown;
    font-family: Arial;

}



header {
    background-color: yellow;
    color: black;
    text-align: center;
    font-size: x-large;
}
```

Three ways to provide styles

Use an external style sheet by coding a link element in the head section

```
<link rel="stylesheet" href="styles/main.css">
```

Embed the styles in the head section

```
<style>
    body {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 87.5%;
    }
    h1 {
        font-size: 250%;
    }
</style>
```

Use the style attribute of an element to provide inline styles

```
<span style="color: red; font-size: 14pt;">Warning!
</span>
```

© 2013, Mike Murach & Associates, Inc.

The sequence in which styles are applied

- Styles from an external style sheet
- Embedded styles
- Inline styles

© 2013, Mike Murach & Associates, Inc.

A head element that includes two external style sheets

```
<head>
    <title>The Halloween Store</title>
    <link rel="stylesheet" href="main.css">
    <link rel="stylesheet" href="order.css">
</head>
```

The sequence in which styles are applied

- From the first external style sheet to the last

How to generate a link element for an external style sheet

- To generate a link element in Source view, drag the style sheet from the Solution Explorer into the head element for the page.
- To generate a link element in Design view, choose the FORMAT→Attach Style Sheet command and select the style sheet from the Select Style Sheet dialog box.

How to enter and edit the styles for an external style sheet

- Open the style sheet in the Editor, and enter the styles into the style sheet.
- If necessary, modify the aspx code so it provides the ids and class names that you need for the selectors in the style sheet.
- After you enter a rule set or a series of rule sets, switch to Design view to see whether the styles are working the way you want them to. Or, test the form in a browser.

© 2013, Mike Murach & Associates, Inc.

How to comment out and uncomment CSS rules

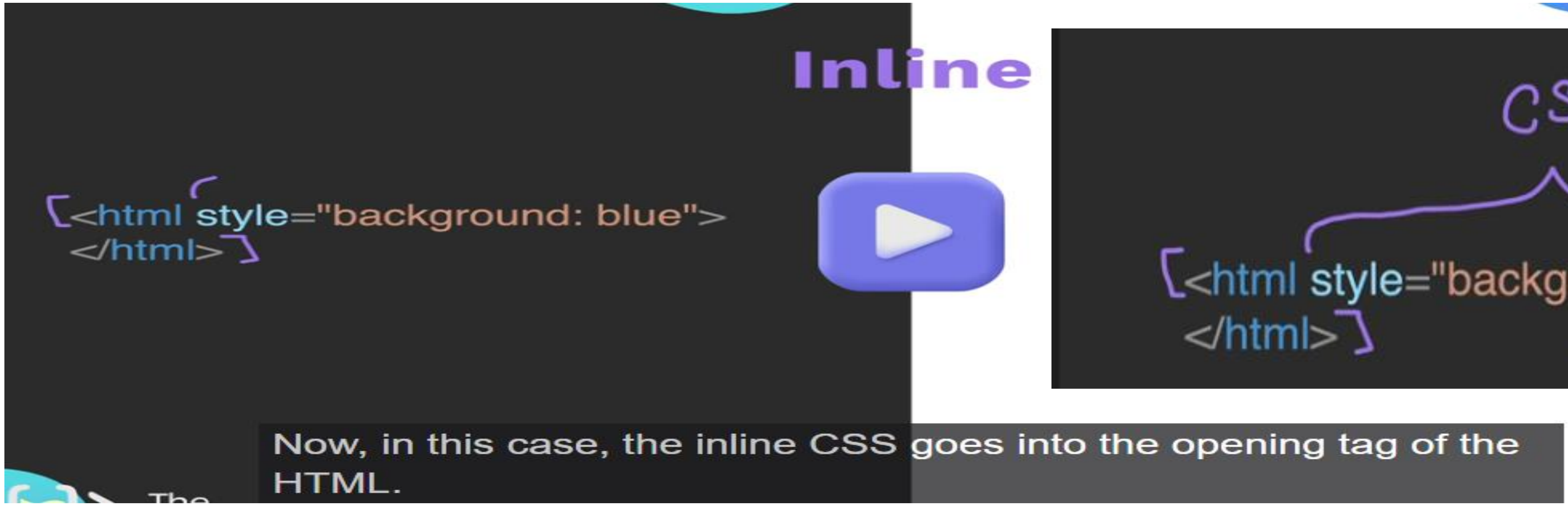
- Press Ctrl+K, Ctrl+C to comment out selected rules, or Ctrl+K , Ctrl+U to uncomment them.
- Or, click the Comment or Uncomment button in the Style Sheet toolbar.

© 2013, Mike Murach & Associates, Inc.

How to use the CSS Outline window

- Use the VIEW→Other Windows→Document Outline command to open this window.
- Then, to navigate to a rule set in the style sheet, click on its selector in this window.

© 2013, Mike Murach & Associates, Inc.

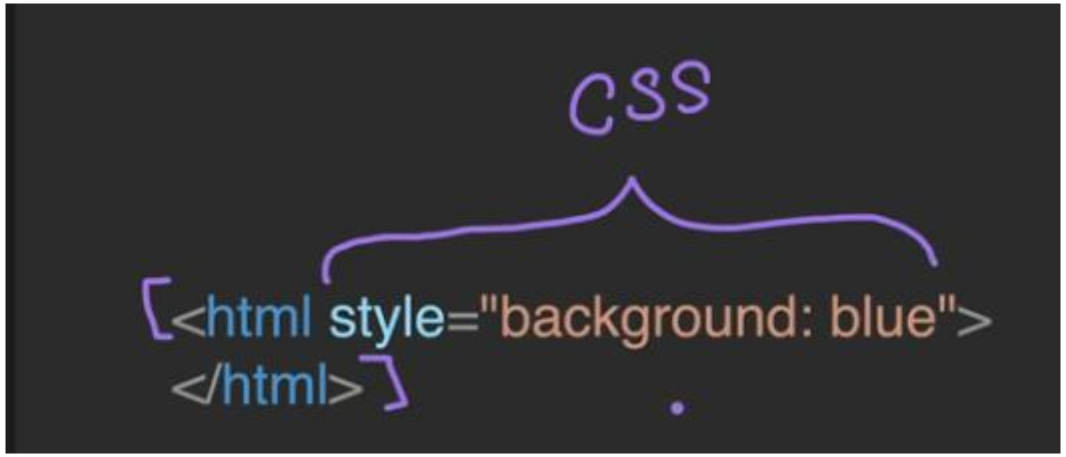


Inline

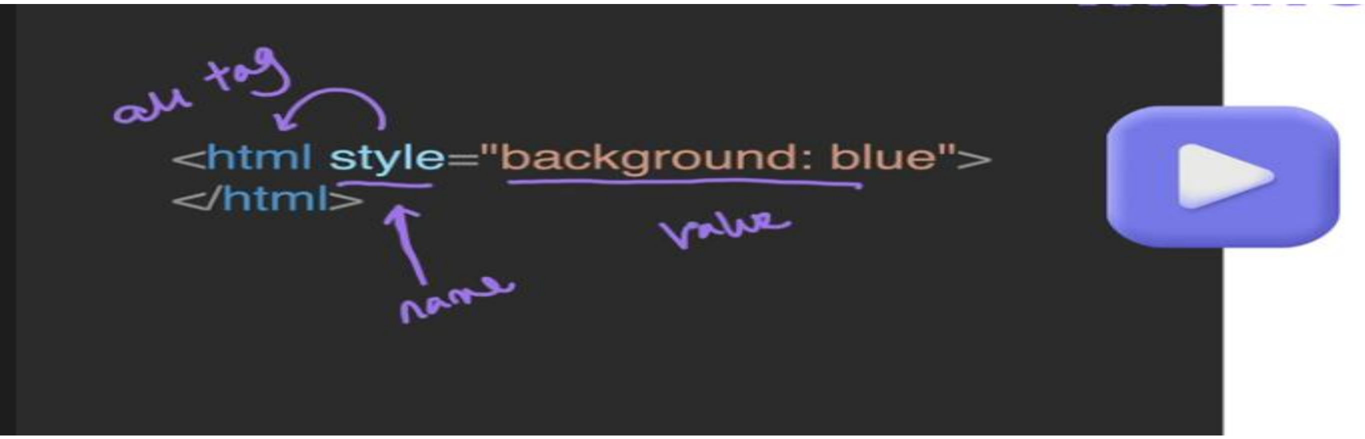
```
<html style="background: blue">
</html>
```

Now, in this case, the inline CSS goes into the opening tag of the HTML.

The

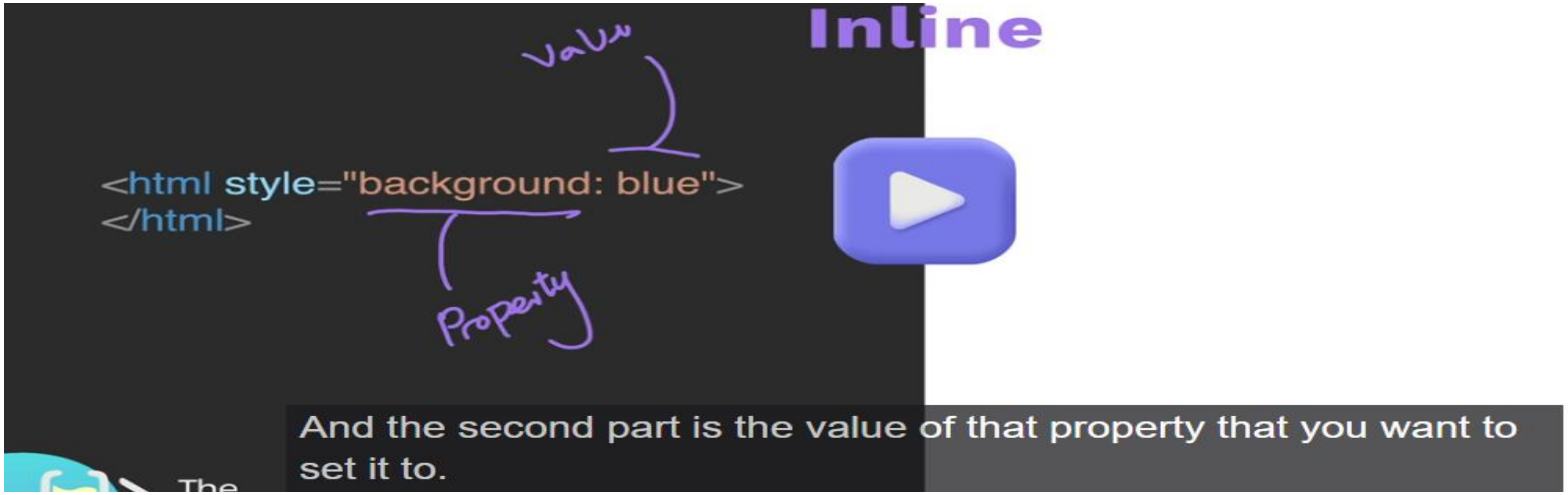


```
<html style="background: blue">
</html>
```



```
<html style="background: blue">
</html>
```

Our CSS code is broken down like this.



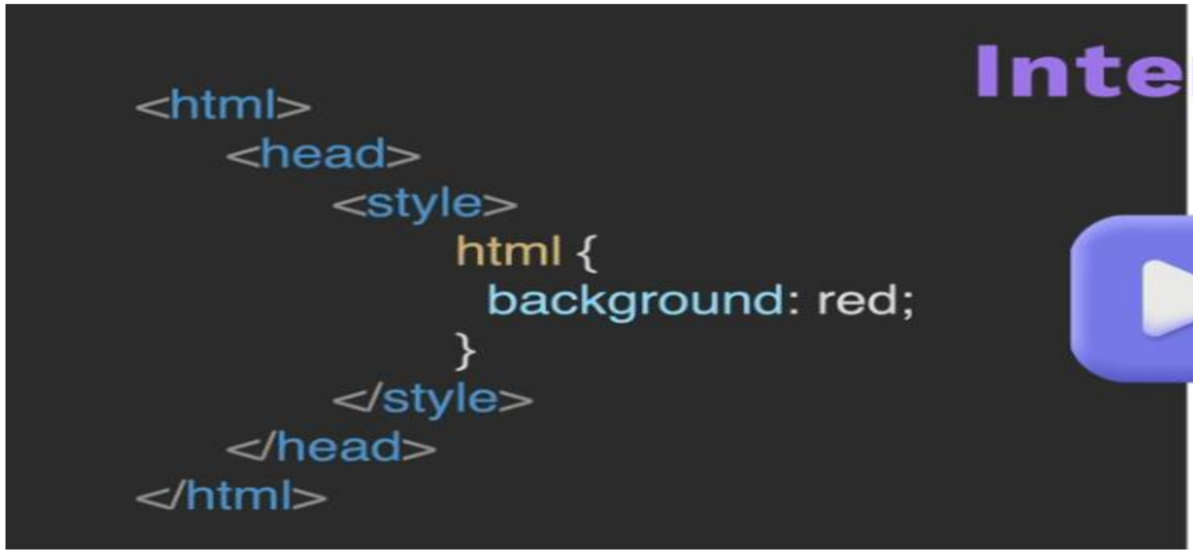
```
<html style="background: blue">
</html>
```

And the second part is the value of that property that you want to set it to.

Inline elements are really useful for adding CSS style to just a single element on your HTML page.

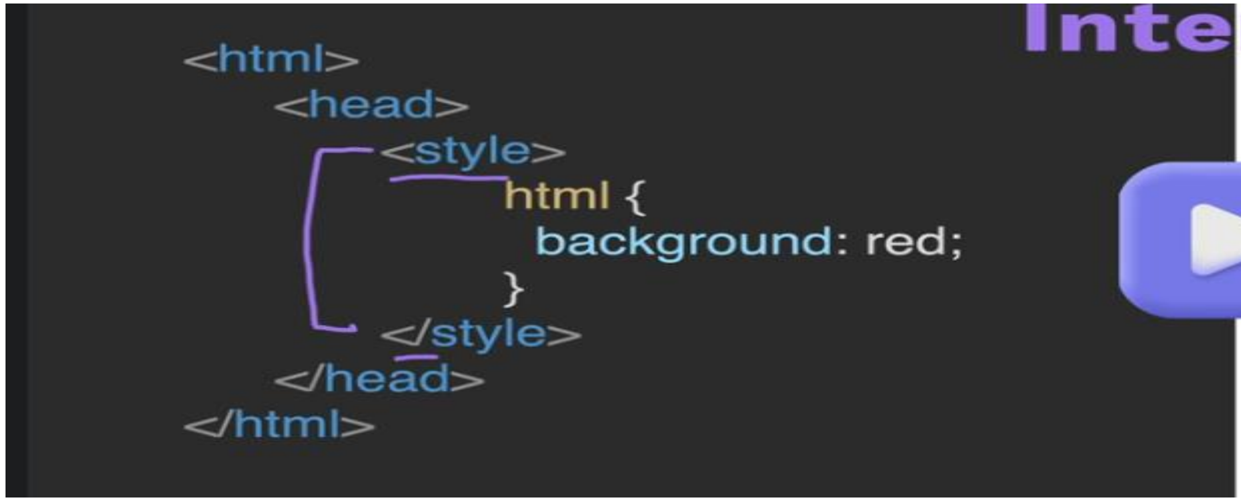
It's not normally recommended to use inline styles in your entire document.

Internal



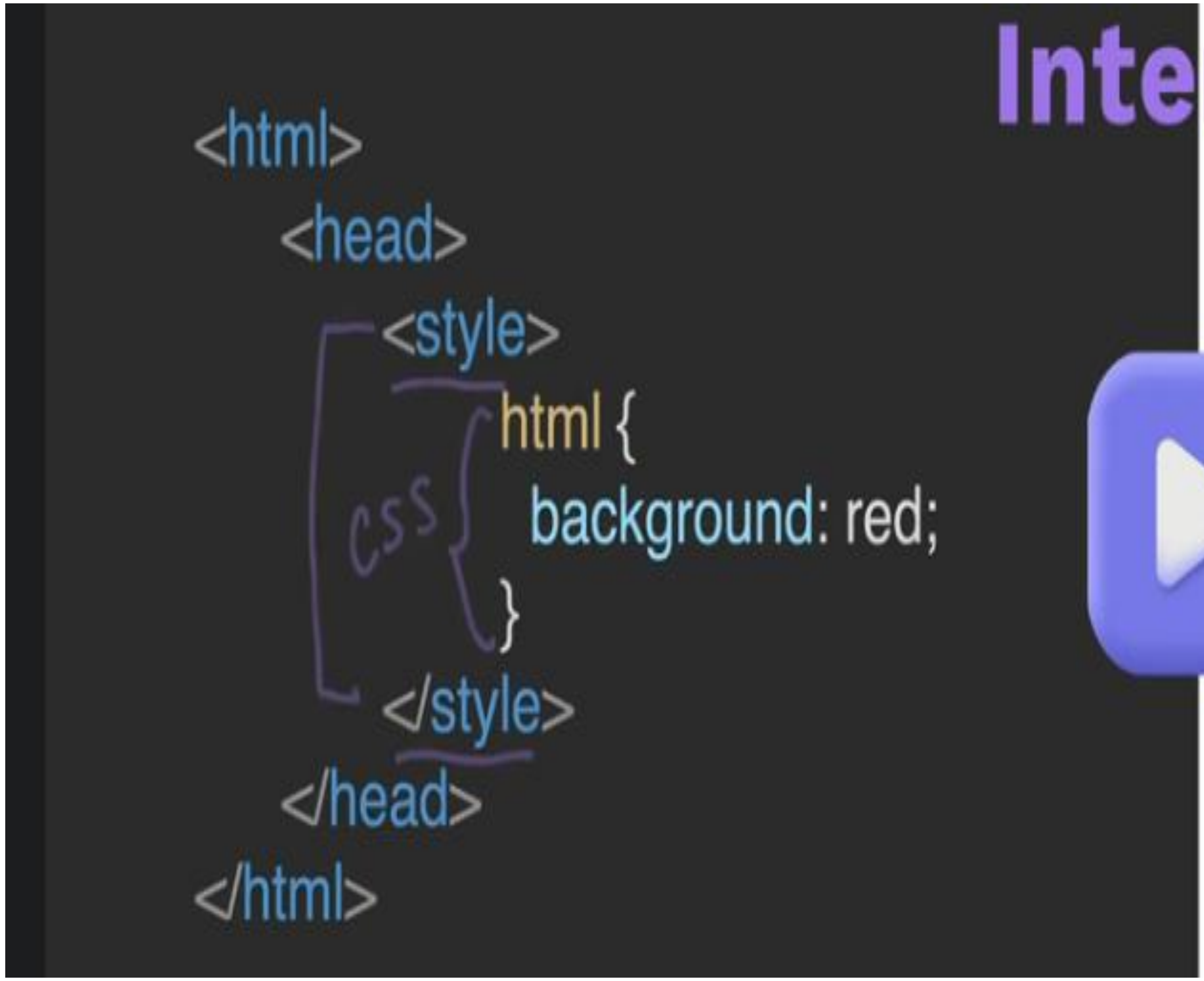
```
<html>
    <head>
        <style>
            html {
              background: red;
            }
        </style>
    </head>
</html>
```

Internal



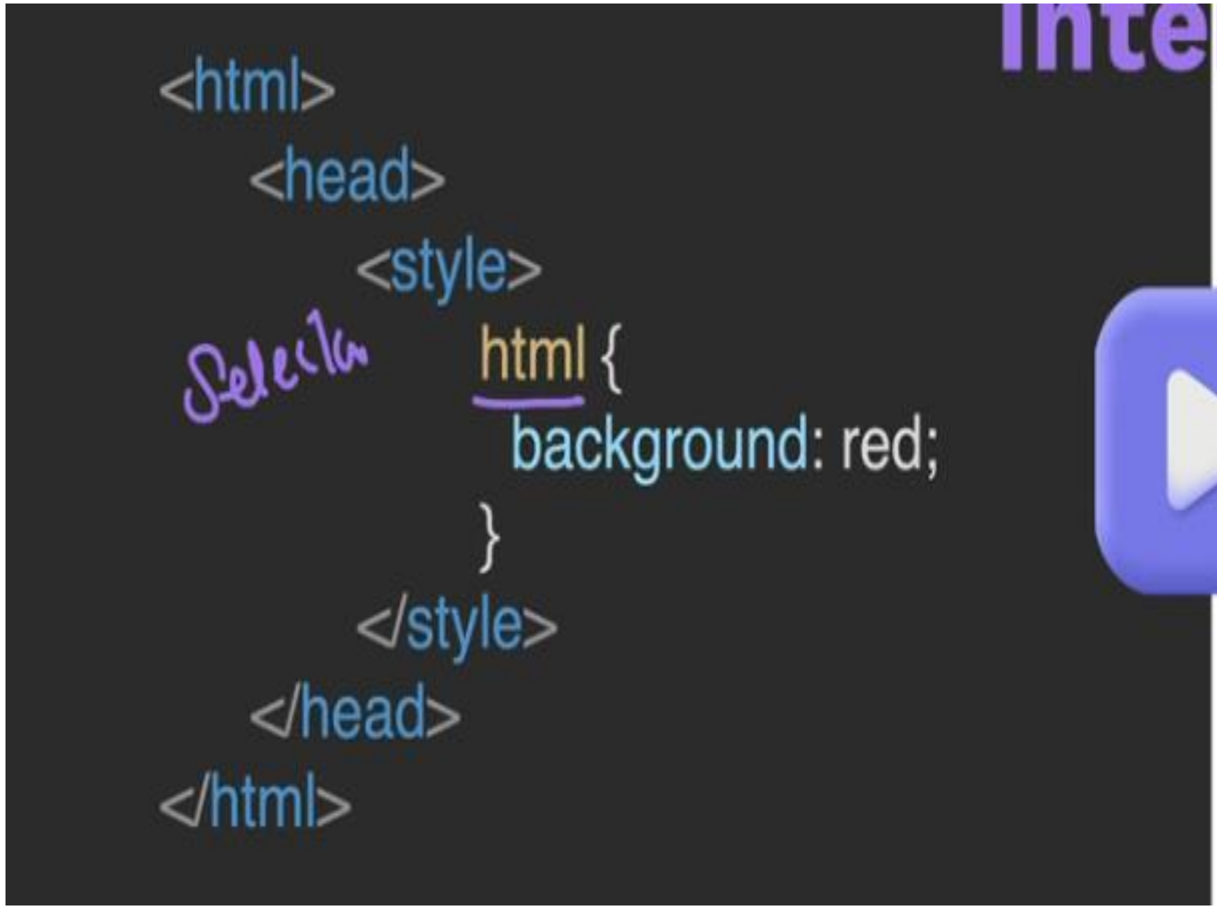
```
<html>
    <head>
        <style>
            html {
              background: red;
            }
        </style>
    </head>
</html>
```

Internal

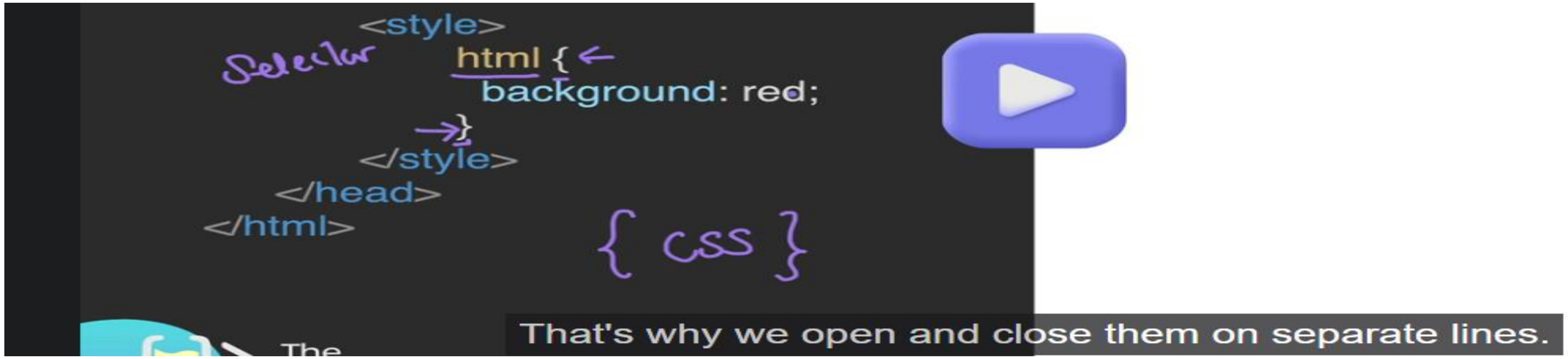


```
<html>
    <head>
        <style>
            html {
              background: red;
            }
        </style>
    </head>
</html>
```

Internal

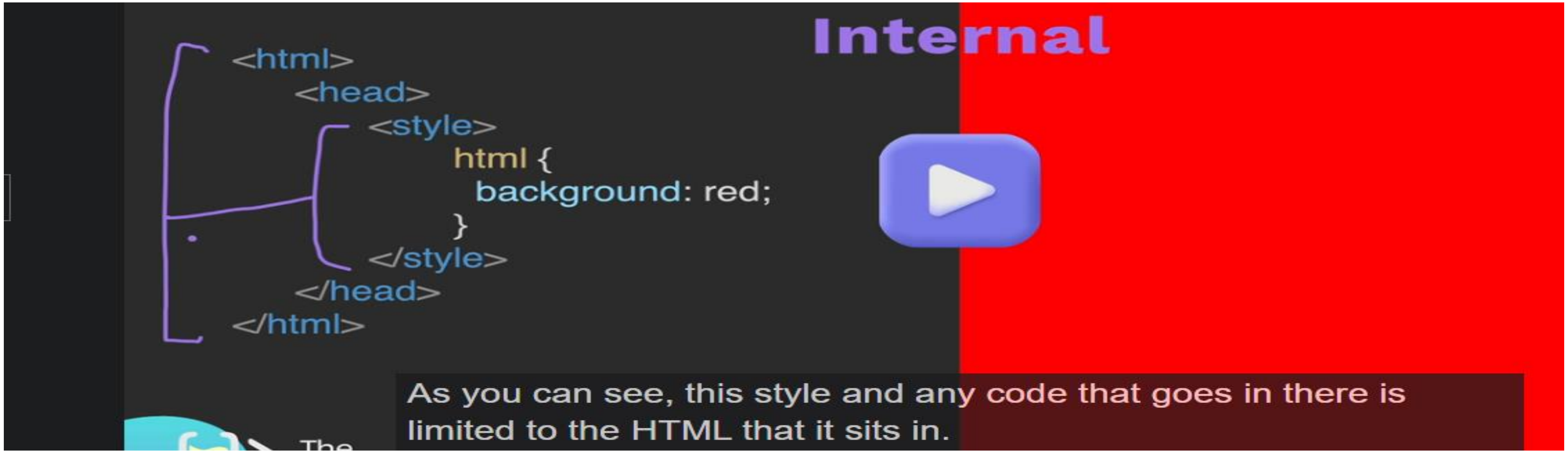


```
<html>
    <head>
        <style>
            html {
              background: red;
            }
        </style>
    </head>
</html>
```



```
    <style>
      html {
        background: red;
      }
    </style>
  </head>
</html>
```

That's why we open and close them on separate lines.

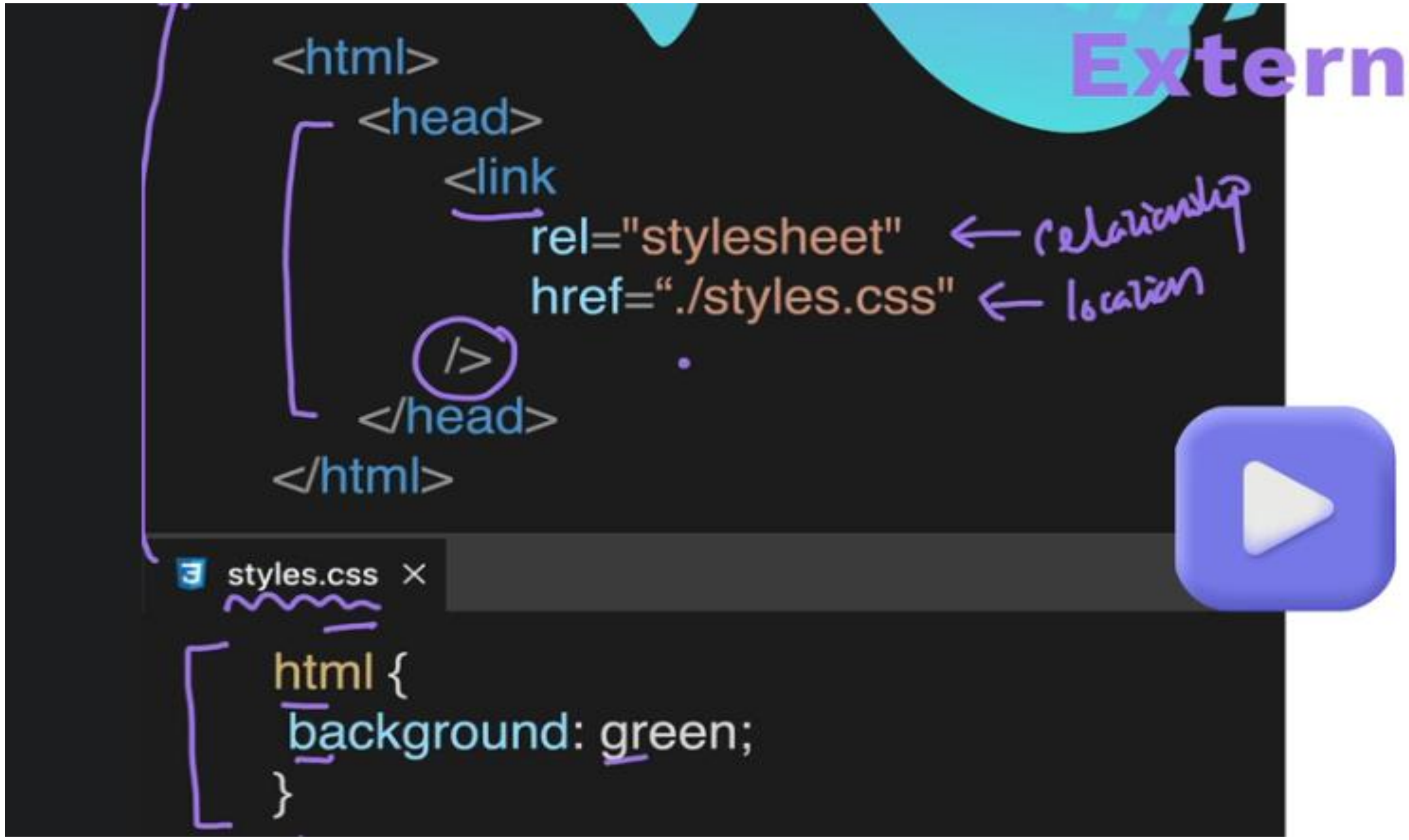


Internal

```
<html>
  <head>
    <style>
      html {
        background: red;
      }
    </style>
  </head>
</html>
```

As you can see, this style and any code that goes in there is limited to the HTML that it sits in.

So it means if you have a multi-page website, then you probably shouldn't be using the internal style.

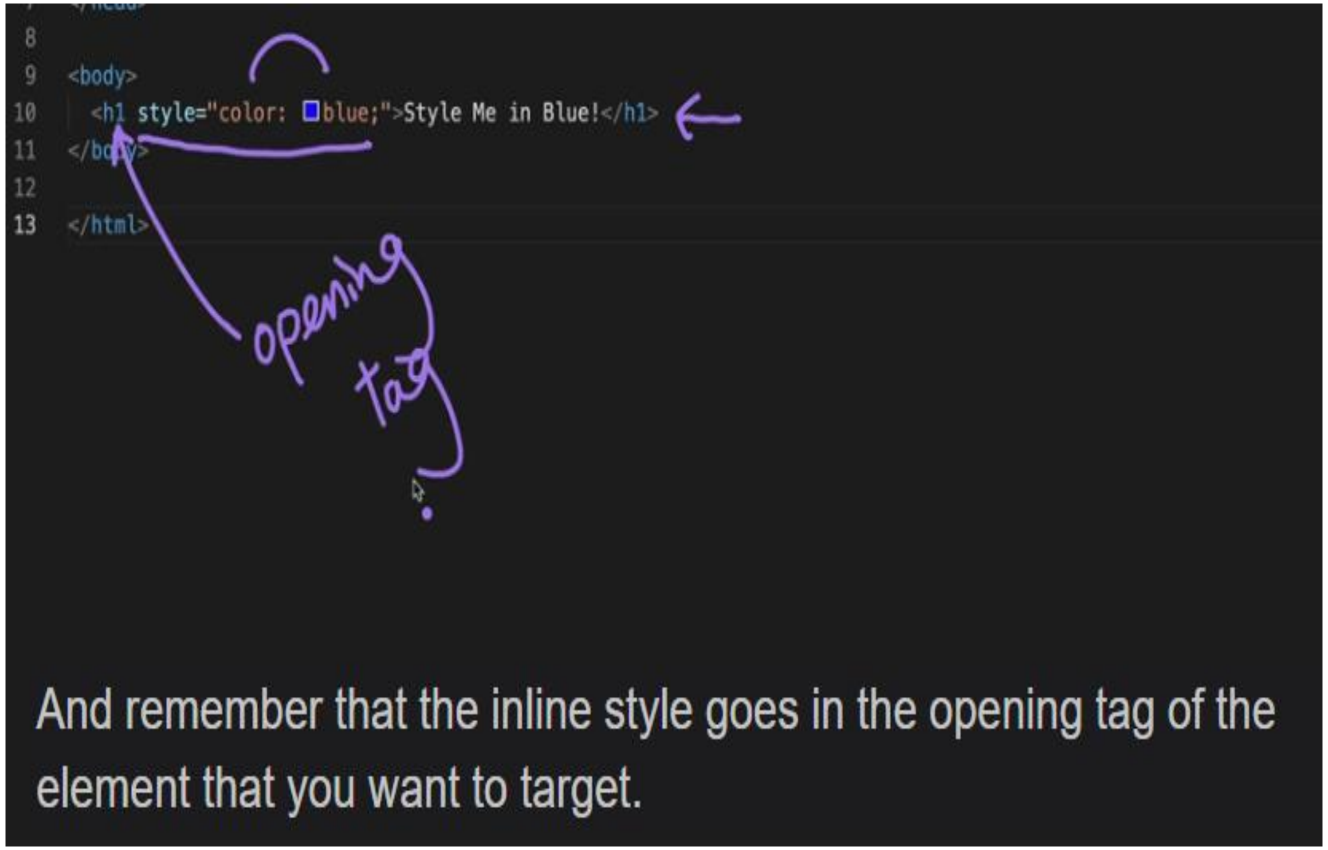


```
<html>
  <head>
    <link
      rel="stylesheet"
      href="./styles.css"
    />
  </head>
</html>
```

styles.css

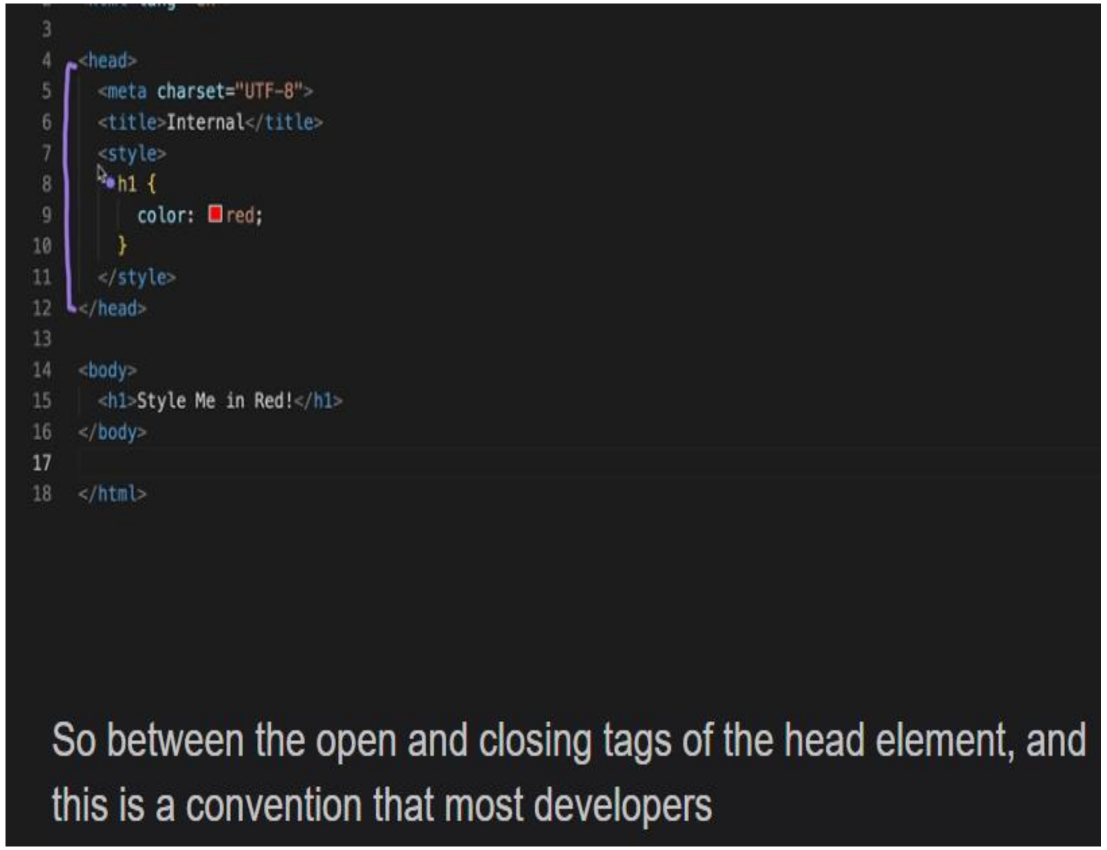
```
html {
 background: green;
}
```

And this style of external CSS is what's used most commonly in web development.



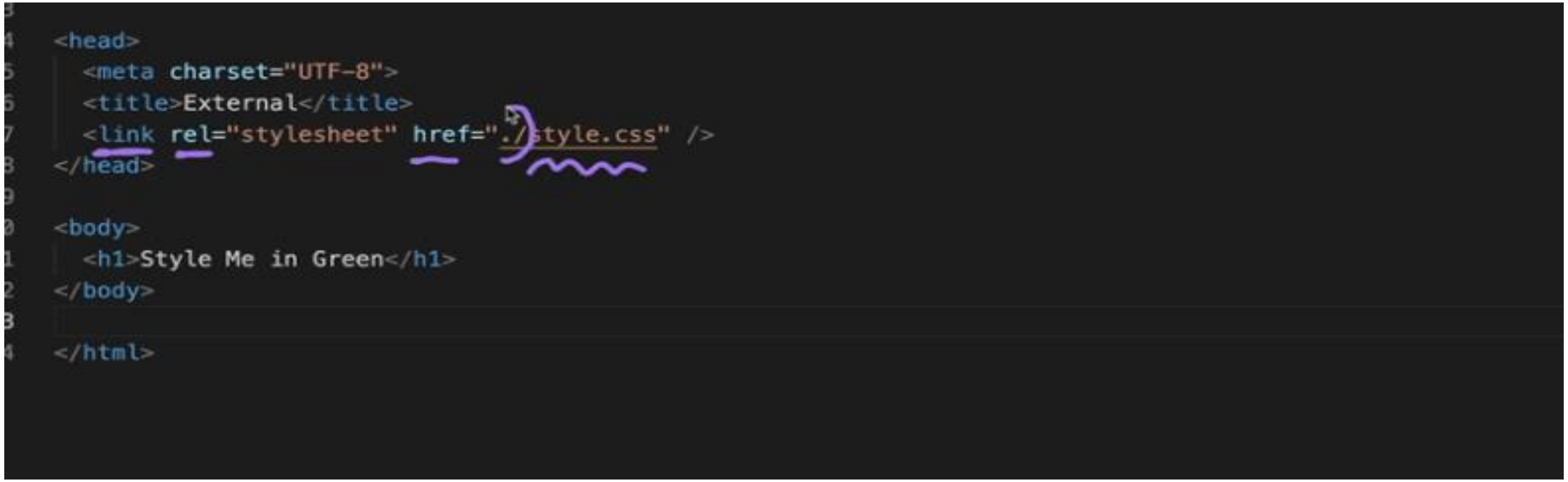
```html
<body>
  <h1 style="color: blue;">Style Me in Blue!</h1>
</body>

</html>
```

And remember that the inline style goes in the opening tag of the element that you want to target.



```html
<head>
  <meta charset="UTF-8">
  <title>Internal</title>
  <style>
    h1 {
      color: red;
    }
  </style>
</head>

<body>
  <h1>Style Me in Red!</h1>
</body>

</html>
```

So between the open and closing tags of the head element, and this is a convention that most developers



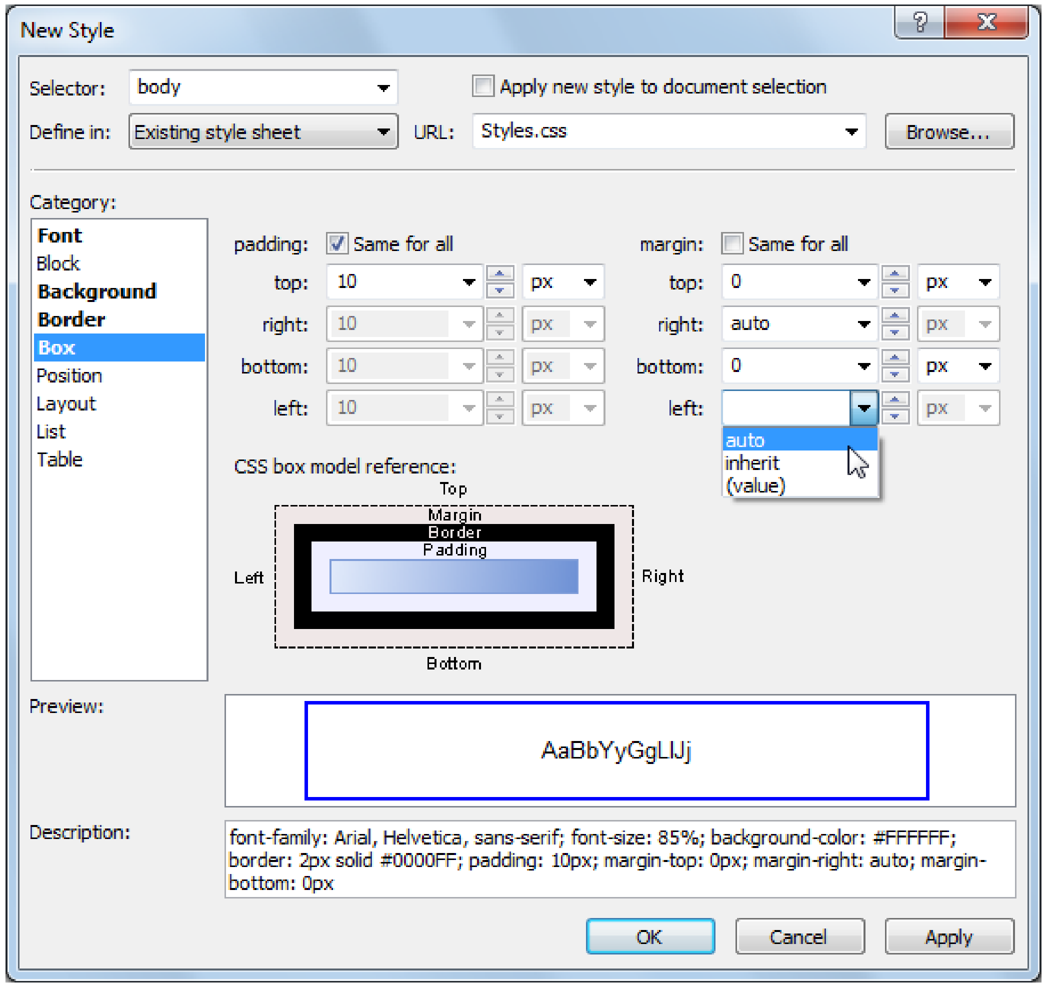
```
<head>
  <meta charset="UTF-8">
  <title>External</title>
  <link rel="stylesheet" href="./style.css" />
</head>

<body>
  <h1>Style Me in Green</h1>
</body>

</html>
```

How to create a new style

- From Design view, open the New Style dialog box by choosing the FORMAT→New Style command or by selecting Apply New Style from the Target Rule drop-down list in the Formatting toolbar.
- In the New Style dialog box, enter or select the Selector for the style, select Existing Style Sheet from the Define In list, and use the Browse button for the URL entry to find the style sheet you want the new style to be placed in.
- To specify the rules for the style, select a Category and set the values for the properties in that category.
- Continue with any of the other categories.

© 2013, Mike Murach & Associates, Inc.

The New Style dialog box


New Style

Selector: body

Define in: Existing style sheet

URL: Styles.css

[ ] Apply new style to document selection

Category: Font, Block, Background, Border, Box, Position, Layout, List, Table

padding: [✓] Same for all			margin: [ ] Same for all		
top:	10	px	top:	0	px
right:	10	px	right:	auto	px
bottom:	10	px	bottom:	0	px
left:	10	px	left:		px

CSS box model reference:

Preview: AaBbYyGgLlJj

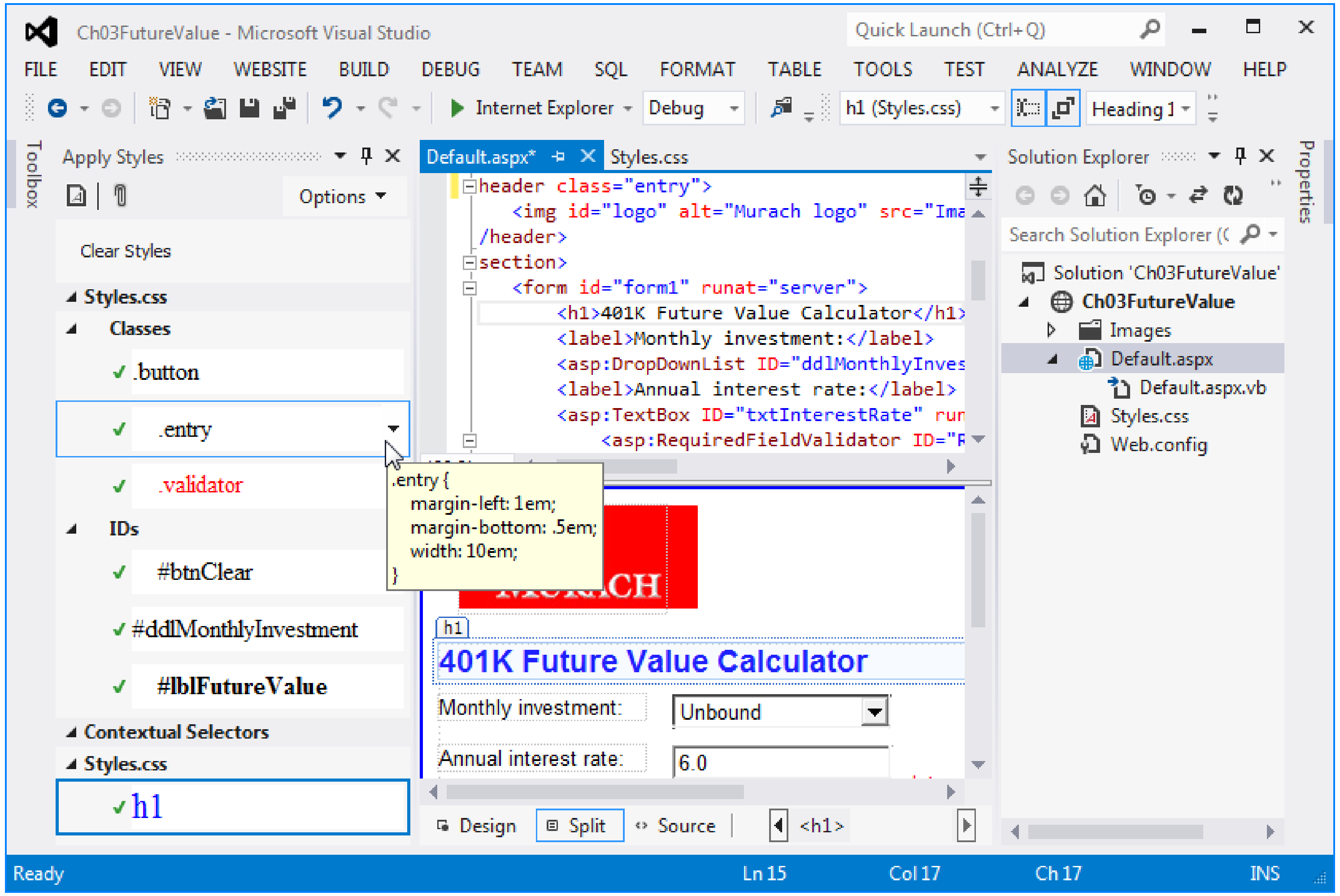
Description: font-family: Arial, Helvetica, sans-serif; font-size: 85%; background-color: #FFFFFF; border: 2px solid #0000FF; padding: 10px; margin-top: 0px; margin-right: auto; margin-bottom: 0px

© 2013, Mike Murach & Associates, Inc.

How to modify a style

- In the Editor for a style sheet, right-click in a style and select Build Style or click on the Build Style button in the Style Sheet toolbar.
- In the Modify Style dialog box, select a category and set or reset the values for the properties in that category.
- Continue with any of the other categories.

The Apply Styles window



```
header class="entry">
    <img id="logo" alt="Murach logo" src="Ima
/header>
section>
    <form id="form1" runat="server">
        <h1>401K Future Value Calculator</h1>
        <label>Monthly investment:</label>
        <asp:DropDownList ID="ddlMonthlyInves
        <label>Annual interest rate:</label>
        <asp:TextBox ID="txtInterestRate" run
            <asp:RequiredFieldValidator ID="R
```

```
.entry {
   margin-left: 1em;
   margin-bottom: .5em;
   width: 10em;
}
```

401K Future Value Calculator

Monthly investment: Unbound

Annual interest rate: 6.0

© 2013, Mike Murach & Associates, Inc.

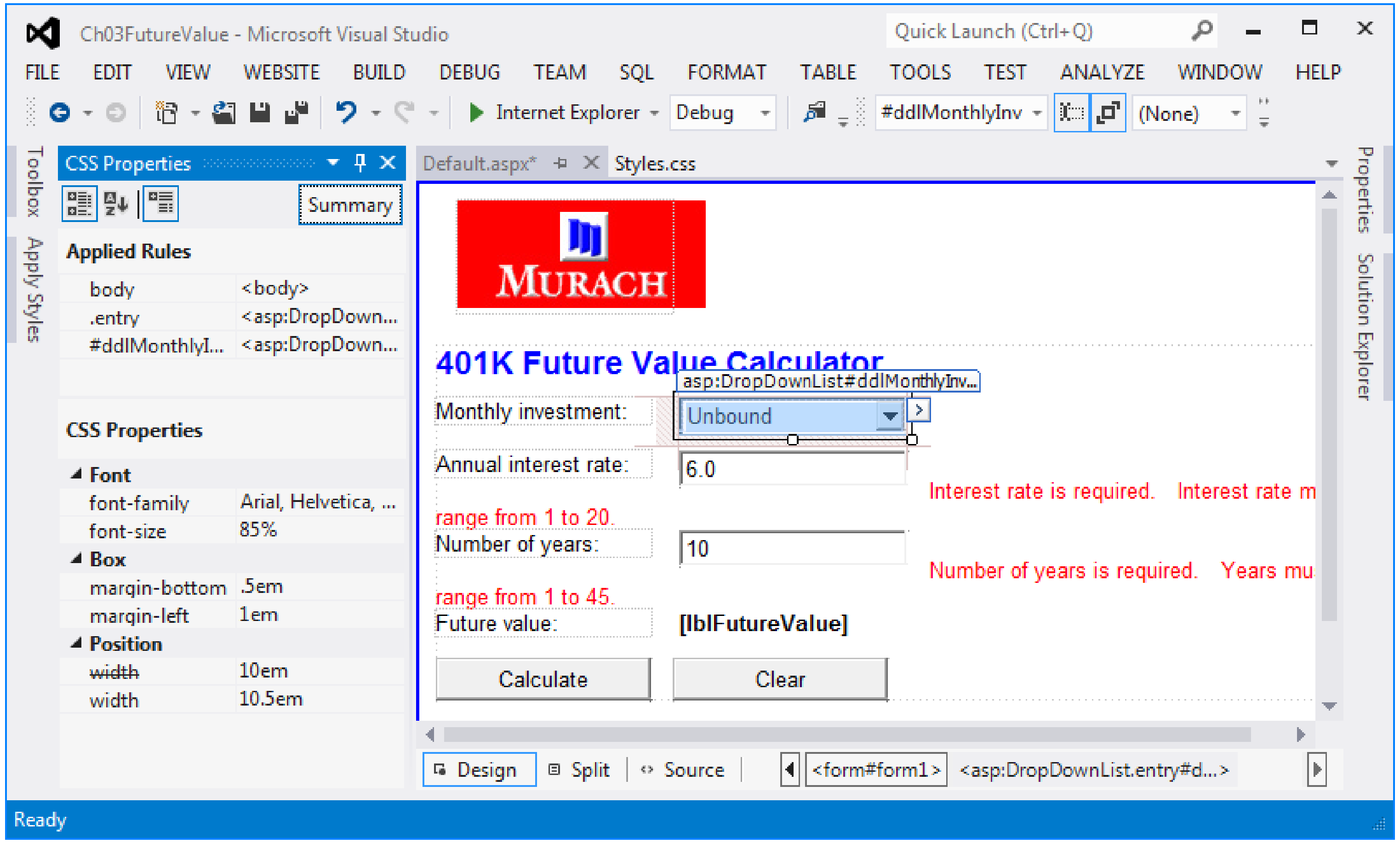
How to display the Apply Styles window

- In any of the Designer views, use the VIEW→Apply Styles command.

How to use the Apply Styles window (cont.)

- To delete a style, select it and choose Delete Style from its drop-down list.
- To remove all class and inline styles for selected elements, click Clear Styles. This removes the class and style attributes from the elements.

The CSS Properties window



Applied Rules

body	<body>
.entry	<asp:DropDown...
#ddlMonthlyI...	<asp:DropDown...

CSS Properties

Font	
font-family	Arial, Helvetica, ...
font-size	85%
Box	
margin-bottom	.5em
margin-left	1em
Position	
~~width~~	10em
width	10.5em

MURACH

401K Future Value Calculator

asp:DropDownList#ddlMonthlyInv...

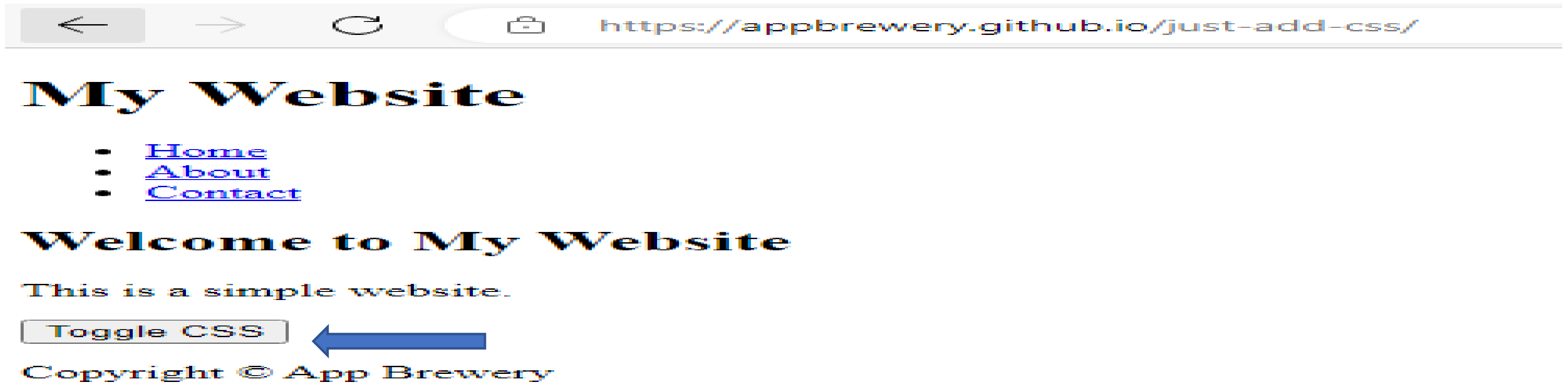
Monthly investment:	Unbound	
Annual interest rate:	6.0	Interest rate is required. Interest rate m
range from 1 to 20.		
Number of years:	10	Number of years is required. Years mu
range from 1 to 45.		
Future value:	[lblFutureValue]	

Calculate   Clear

© 2013, Mike Murach & Associates, Inc.

If you go to this Web page and you Toggle CSS you will find the effects of CSS on the web page transformation.

https://appbrewery.github.io/just-add-css/


https://appbrewery.github.io/just-add-css/

My Website

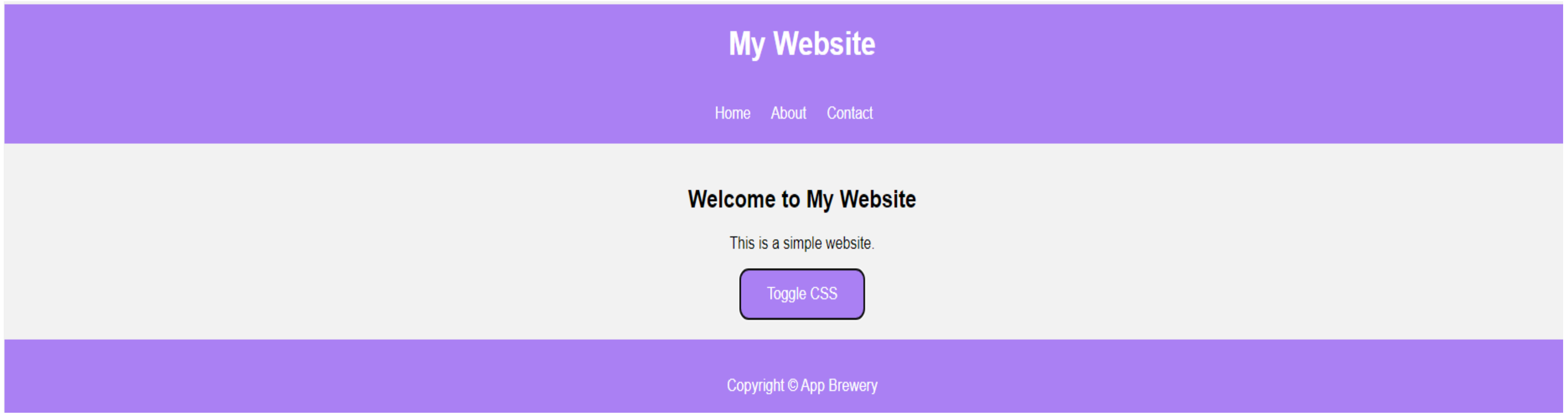
- Home
- About
- Contact

Welcome to My Website

This is a simple website.

Toggle CSS

Copyright © App Brewery


My Website

Home About Contact

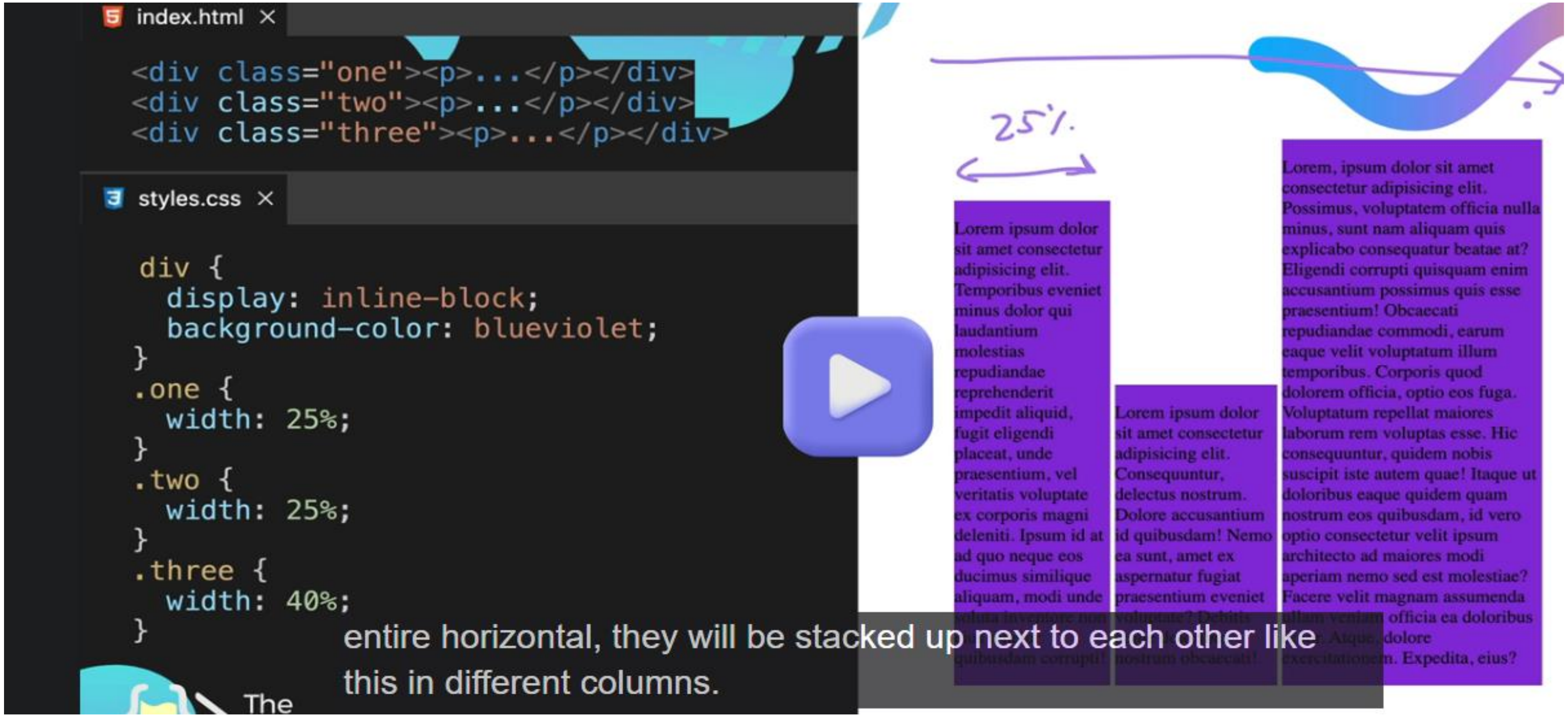
Welcome to My Website

This is a simple website.

Toggle CSS

Copyright © App Brewery

As we mentioned, a lot of the web layout was originally inspired by newspapers and magazine articles,



index.html

```html
<div class="one"><p>...</p></div>
<div class="two"><p>...</p></div>
<div class="three"><p>...</p></div>
```

styles.css

```css
div {
  display: inline-block;
  background-color: blueviolet;
}
.one {
  width: 25%;
}
.two {
  width: 25%;
}
.three {
  width: 40%;
}
```

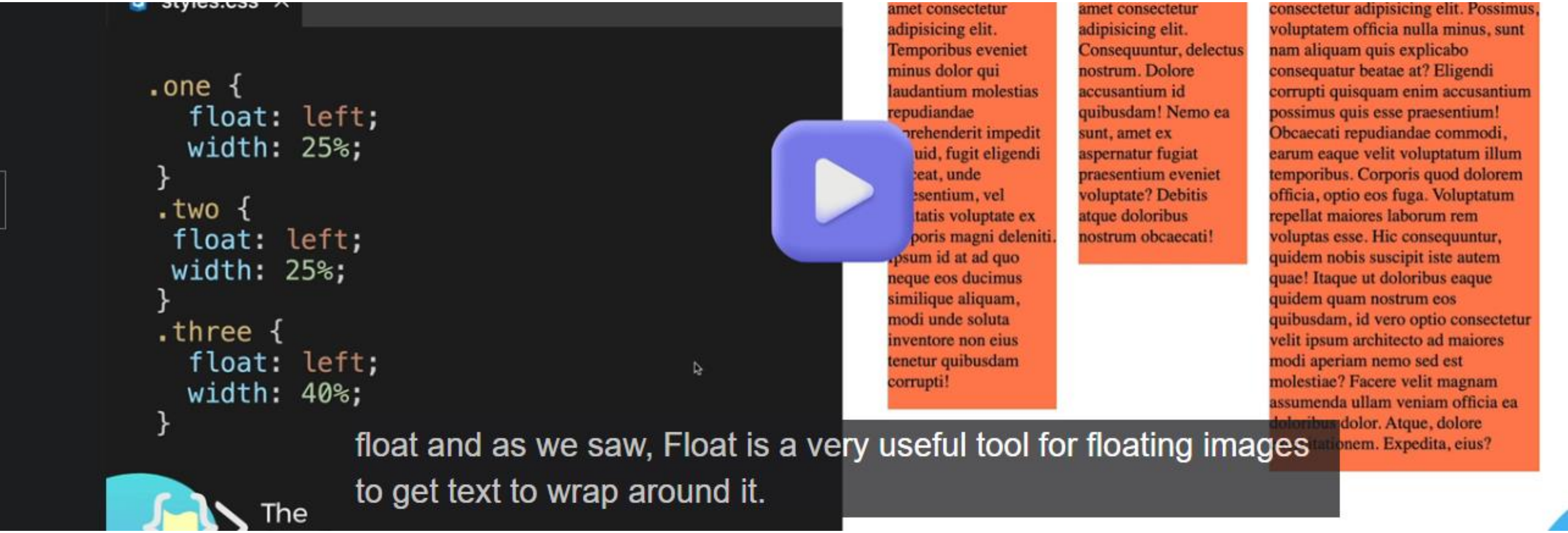
Lorem ipsum dolor sit amet consectetur adipisicing elit. Temporibus eveniet minus dolor qui laudantium molestias repudiandae reprehenderit impedit aliquid, fugit eligendi placeat, unde praesentium, vel veritatis voluptate ex corporis magni deleniti. Ipsum id at ad quo neque eos ducimus similique aliquam, modi unde soluta inventore non quibusdam corrupti!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Consequuntur, delectus nostrum. Dolore accusantium id quibusdam! Nemo ea sunt, amet ex aspernatur fugiat praesentium eveniet voluptate? Debitis nostrum obcaecati!

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Possimus, voluptatem officia nulla minus, sunt nam aliquam quis explicabo consequatur beatae at? Eligendi corrupti quisquam enim accusantium possimus quis esse praesentium! Obcaecati repudiandae commodi, earum eaque velit voluptatum illum temporibus. Corporis quod dolorem officia, optio eos fuga. Voluptatum repellat maiores laborum rem voluptas esse. Hic consequuntur, quidem nobis suscipit iste autem quae! Itaque ut doloribus eaque quidem quam nostrum eos quibusdam, id vero optio consectetur velit ipsum architecto ad maiores modi aperiam nemo sed est molestiae? Facere velit magnam assumenda ullam veniam officia ea doloribus [illegible]. Atque, dolore exercitationem. Expedita, eius?

entire horizontal, they will be stacked up next to each other like this in different columns.

The



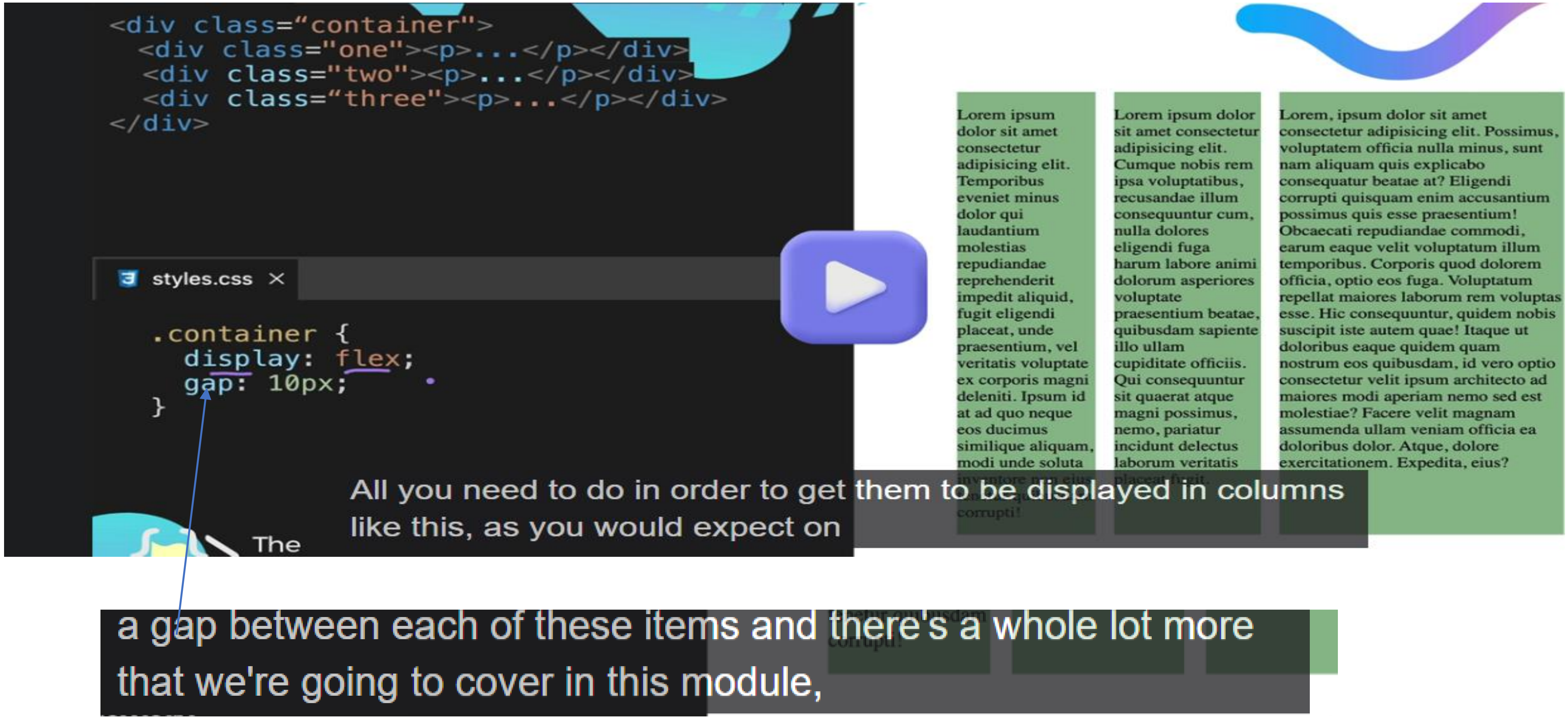
```css
.one {
  float: left;
  width: 25%;
}
.two {
 float: left;
 width: 25%;
}
.three {
  float: left;
  width: 40%;
}
```

amet consectetur
adipisicing elit.
Temporibus eveniet
minus dolor qui
laudantium molestias
repudiandae
prehenderit impedit
uid, fugit eligendi
eat, unde
sentium, vel
tatis voluptate ex
poris magni deleniti.
psum id at ad quo
neque eos ducimus
similique aliquam,
modi unde soluta
inventore non eius
tenetur quibusdam
corrupti!

amet consectetur
adipisicing elit.
Consequuntur, delectus
nostrum. Dolore
accusantium id
quibusdam! Nemo ea
sunt, amet ex
aspernatur fugiat
praesentium eveniet
voluptate? Debitis
atque doloribus
nostrum obcaecati!

consectetur adipisicing elit. Possimus,
voluptatem officia nulla minus, sunt
nam aliquam quis explicabo
consequatur beatae at? Eligendi
corrupti quisquam enim accusantium
possimus quis esse praesentium!
Obcaecati repudiandae commodi,
earum eaque velit voluptatum illum
temporibus. Corporis quod dolorem
officia, optio eos fuga. Voluptatum
repellat maiores laborum rem
voluptas esse. Hic consequuntur,
quidem nobis suscipit iste autem
quae! Itaque ut doloribus eaque
quidem quam nostrum eos
quibusdam, id vero optio consectetur
velit ipsum architecto ad maiores
modi aperiam nemo sed est
molestiae? Facere velit magnam
assumenda ullam veniam officia ea
doloribus dolor. Atque, dolore
tationem. Expedita, eius?

float and as we saw, Float is a very useful tool for floating images to get text to wrap around it.



```
<div class="container">
  <div class="one"><p>...</p></div>
   <div class="two"><p>...</p></div>
   <div class="three"><p>...</p></div>
</div>
```

```
.container {
  display: flex;
  gap: 10px;
}
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Temporibus eveniet minus dolor qui laudantium molestias repudiandae reprehenderit impedit aliquid, fugit eligendi placeat, unde praesentium, vel veritatis voluptate ex corporis magni deleniti. Ipsum id at ad quo neque eos ducimus similique aliquam, modi unde soluta inventore non eius [illegible] corrupti!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cumque nobis rem ipsa voluptatibus, recusandae illum consequuntur cum, nulla dolores eligendi fuga harum labore animi dolorum asperiores voluptate praesentium beatae, quibusdam sapiente illo ullam cupiditate officiis. Qui consequuntur sit quaerat atque magni possimus, nemo, pariatur incidunt delectus laborum veritatis placeat fugit.

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Possimus, voluptatem officia nulla minus, sunt nam aliquam quis explicabo consequatur beatae at? Eligendi corrupti quisquam enim accusantium possimus quis esse praesentium! Obcaecati repudiandae commodi, earum eaque velit voluptatum illum temporibus. Corporis quod dolorem officia, optio eos fuga. Voluptatum repellat maiores laborum rem voluptas esse. Hic consequuntur, quidem nobis suscipit iste autem quae! Itaque ut doloribus eaque quidem quam nostrum eos quibusdam, id vero optio consectetur velit ipsum architecto ad maiores modi aperiam nemo sed est molestiae? Facere velit magnam assumenda ullam veniam officia ea doloribus dolor. Atque, dolore exercitationem. Expedita, eius?

All you need to do in order to get them to be displayed in columns like this, as you would expect on

The

a gap between each of these items and there's a whole lot more that we're going to cover in this module,

Flexbox container and items

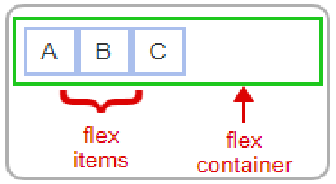
The ***Flexible Box*** or ***flexbox*** is a CSS layout mode that provides an efficient way to lay out elements in a container so the elements behave predictably when the container is resized or viewed on different screen sizes.

A ***flex container*** is an element that has the CSS property `display` set to `flex` to create a block-level flex container or `inline-flex` to create an inline flex container. Ex: `<div style="display: flex">`. Flex containers hold flex items. A ***flex item*** is a child element of a flex container that is positioned and sized according to various CSS flexbox properties.

PARTICIPATION ACTIVITY | 5.1.1: Flexbox example renders three div elements on the same row.

Start 2x speed

```html
<div id="container">
   <div>A</div>
   <div>B</div>
   <div>C</div>
</div>
```



```css
/* flex container */
#container {
   display: flex;
   border: 1px green solid;
   padding: 5px;
}

/* flex items */
#container > div {
   padding: 10px;
   border: 1px blue solid;
}
```

Captions

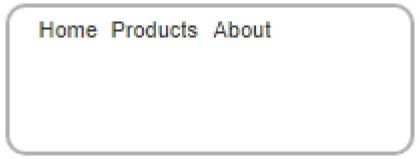
1. Without any CSS, the A, B, and C div elements display vertically, each filling the browser width.
2. Setting the CSS display property to "flex" makes the outer div the flex container. The flex items now display on the same row.
3. The flex items have padding and blue borders.

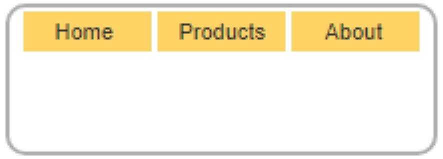
```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="products.html">Products</a></li>
    <li><a href="about.html">About</a></li>
  </ul>
</nav>
```

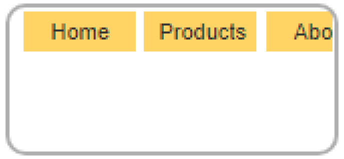
```
nav ul {
  display: flex;
  list-style-type: none;
  padding: 0;
}
```

```
nav li {
  flex-grow: 1;
  background-color: gold;
  text-align: center;
}
```

```
nav li {
  flex-basis: 100px;
  flex-shrink: 0;
  background-color: gold;
  text-align: center;
}
```







aptions

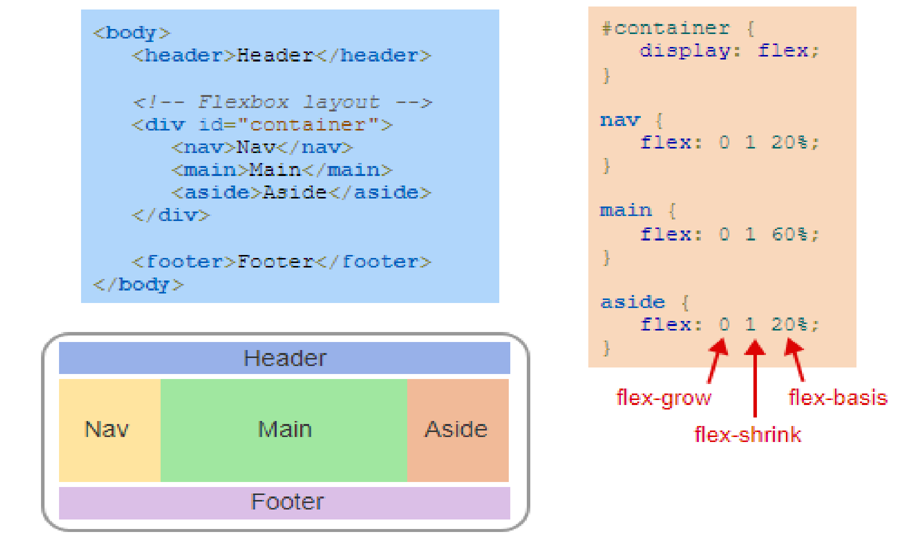
1. A website's navigation links are displayed in an unordered list.
2. Making the ul element a flex container places the nav links on the same row.
3. By default, the li elements have flex-basis:auto and flex-grow:0, so li elements are only as wide as the item's content.
4. Changing flex-grow from the default 0 to 1 gives all li elements the same proportion. The elements fill the flex container.
5. Replacing "flex-grow:1" with "flex-basis:100px" makes each li element 100px wide.
6. Resizing the browser changes the container size. When the container shrinks, the li elements shrink to fill the available space.
7. Changing flex-shrink from the default 1 to 0 prevents the li elements from shrinking when the browser is resized.

The flex property

The shorthand property ***flex*** specifies `flex-grow`, `flex-shrink`, and `flex-basis` together. Ex: `flex: 0 1 auto;` is the same as `flex-grow: 0; flex-shrink: 1; flex-basis: auto;`.

PARTICIPATION ACTIVITY 5.1.7: Flexbox layout using the flex property.

Start 2x speed



```
<body>
   <header>Header</header>

   <!-- Flexbox layout -->
   <div id="container">
      <nav>Nav</nav>
      <main>Main</main>
      <aside>Aside</aside>
   </div>

   <footer>Footer</footer>
</body>
```

```
#container {
   display: flex;
}

nav {
   flex: 0 1 20%;
}

main {
   flex: 0 1 60%;
}

aside {
   flex: 0 1 20%;
}
```

Captions

1. <header> and <footer> span the entire width of <body>, but the <div> is a flex container that displays the flex items on the same row.
2. <nav>, <main>, and <aside> all have flex-grow = 0, so all three flex items' width should be based on each item's content.
3. <nav>, <main>, and <aside> all have flex-shrink = 1, so all three flex items shrink at the same rate when the browser is resized.
4. <nav> occupies 20% of the row, <main> occupies 60%, and <aside> occupies 20%. 20% + 60% + 20% = 100% of the row.