

topicNine

Cascading Style Sheet

Learning Objectives

- At the end of this chapter the students should be able to:
 - To use *class selector* to apply styles into html attributes.
 - To use style sheet to give all web pages of the websites the same format and layout.
 - To specify standard font, font color, font size, and others aspect of text
 - To create a style sheet that governs the total appearance of the whole website.
 - To specify the form styles of the web pages.

Introduction

- **What is CSS?**

- ☐ **CSS** stands for **Cascading Style Sheets**
- ☐ Styles define **how to display** HTML elements
- ☐ Styles are normally stored in **Style Sheets**
- ☐ Styles were added to HTML 4.0 **to solve a problem**
- ☐ **External Style Sheets** can save you a lot of work
- ☐ External Style Sheets are stored in **CSS files**
- ☐ Multiple style definitions will **cascade** into one

Style Sheets Can Save a Lot of Work

- Styles sheets define HOW HTML elements are to be displayed, just like the font tag and the color attribute in HTML 3.2.
- Styles are normally saved in external .css files.
- External style sheets enable you to change the appearance and layout of all the pages in your Web, just by editing one single CSS document.
- CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once.
- As a Web developer you can define a style for each HTML element and apply it to as many Web pages as you want.
- To make a global change, simply change the style, and all elements in the Web are updated automatically.

Multiple Styles Will Cascade Into One

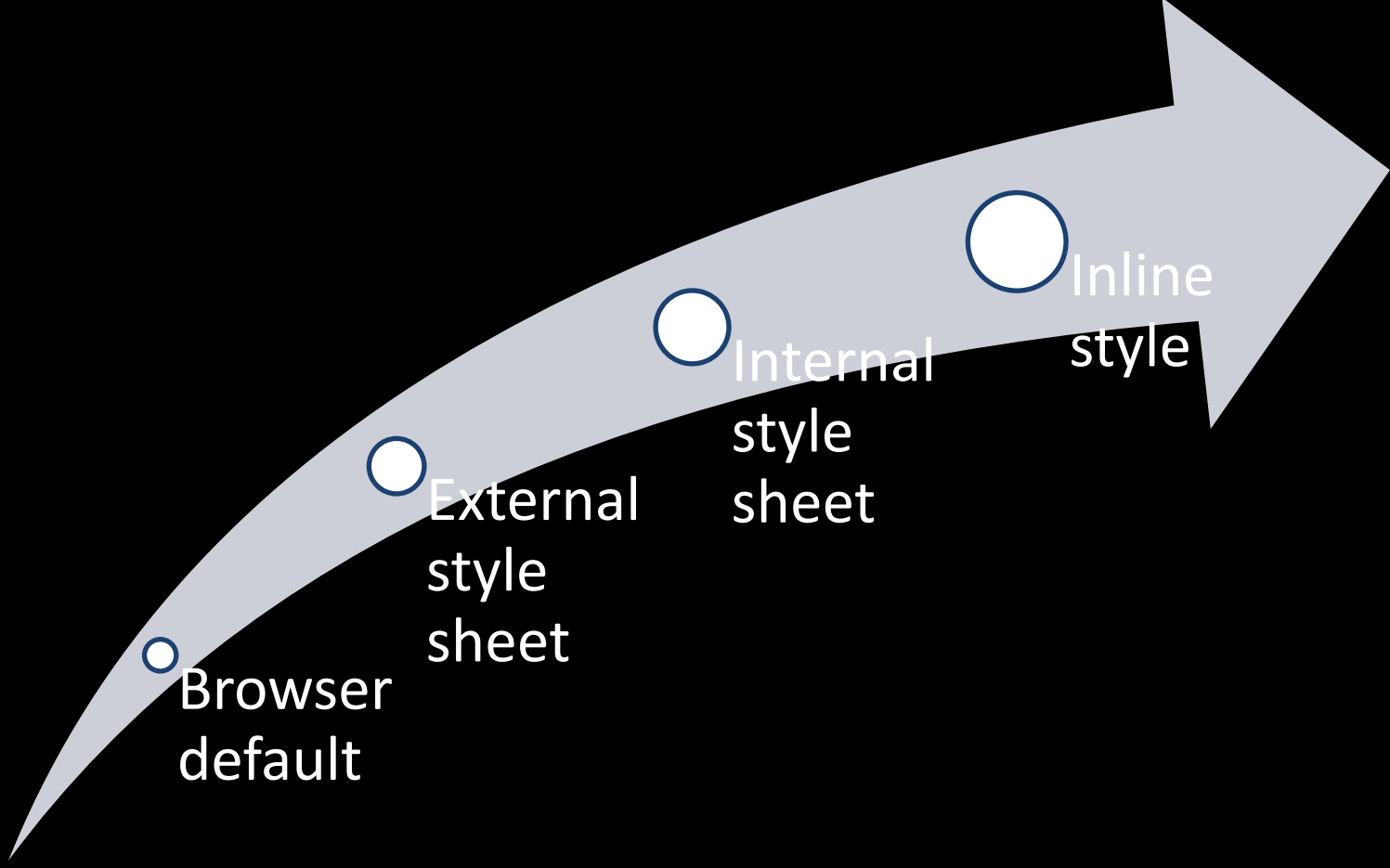
- Style sheets allow style information to be specified in many ways.
- Styles can be specified inside a single HTML element, inside the <head> element of an HTML page, or in an external CSS file.
- Even multiple external style sheets can be referenced inside a single HTML document.

Cascading Order

- **What style will be used when there is more than one style specified for an HTML element?**
- Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:
 - Browser default
 - External style sheet
 - Internal style sheet (inside the <head> tag)
 - Inline style (inside an HTML element)

Cascading Order

- So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style declared inside the <head> tag, in an external style sheet, or in a browser (a default value).



The advantages of using Cascading Style Sheet



Separating style from contents



Improved usability



Simplify presentation



Protecting website styles



Standardize the web pages appearance

CSS : Syntax

- The CSS syntax is made up of three parts: a selector, a property and a value

selector {property: value}

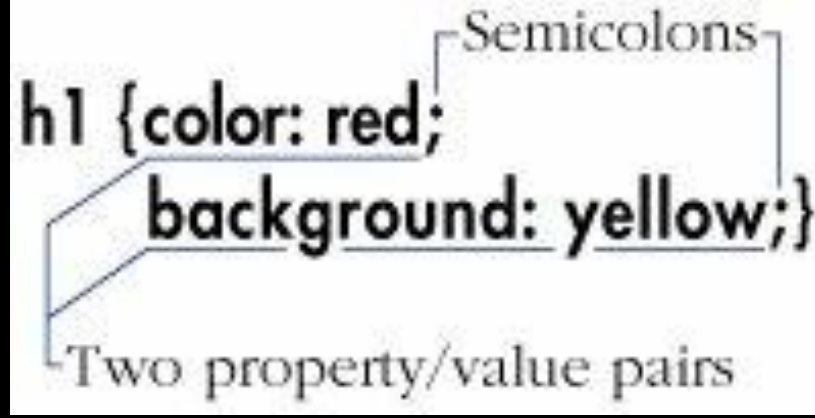
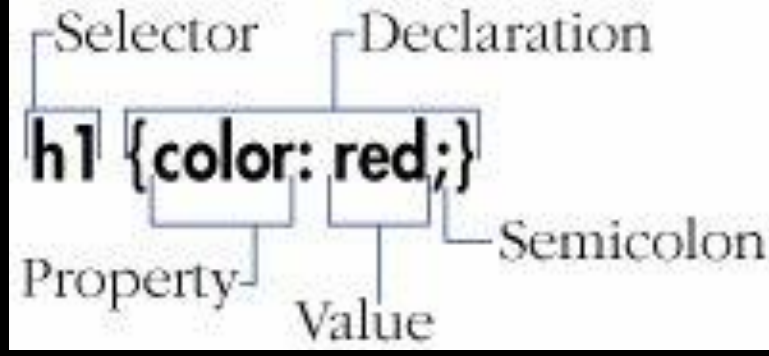
- The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value.
- The property and value are separated by a colon, and surrounded by curly braces

CSS : Syntax

- The CSS syntax is made up of three parts: a selector, a property and a value

selector {property: value}

- The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value.
- The property and value are separated by a colon, and surrounded by curly braces



CSS : Syntax

```
body {color: black}
```

Note: If the value is multiple words, put quotes around the value:

```
p {font-family: "sans serif"}
```

Note: If you wish to specify more than one property, you must separate each property with a semicolon. The example below shows how to define a center aligned paragraph, with a red text color:

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

To make the style definitions more readable, you can describe one property on each line, like this:

```
p
{
text-align: center;
color: black;
font-family: arial
}
```

Grouping

You can group selectors. Separate each selector with a comma. In the example below we have grouped all the header elements. All header elements will be displayed in green text color:

```
h1,h2,h3,h4,h5,h6
{
color: green
}
```

A Property's Value

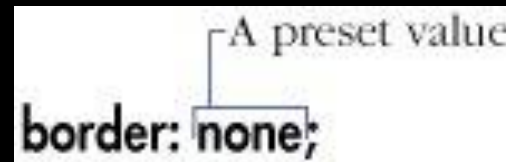
- Each CSS property has different rules about what values it can accept.
- Some properties only accept one of a list of predefined values.
- Others accept numbers, integers, relative values, percentages, URLs, or colors.
- Some can accept more than one type of value.

A Property's Value: *Inherit*

- You can use the inherit value for any property when you want to explicitly specify that the value for that property be the same as that of the element's parent.

A Property's Value: *Predefined Values*

- Most CSS properties have a few predefined values that can be used.
- For example, the display property can be set to **block**, **inline**, **list-item**, or **none**.
- In contrast with HTML, you don't need to and indeed *must not* enclose predefined values in quotation marks.
- Many CSS properties will only accept values from a predefined list.
- Type them exactly and do not enclose them in quotation marks.



A preset value

```
border: none;
```

A Property's Value: *Lengths and Percentages*

- Many CSS properties take a *length* for their value.
- All length values must contain a quantity and a unit, with no spaces between them, for example, **3em** or **10px**.
- The only exception is **0**, which may be used with or without units.
- Lengths must always explicitly state the unit.
- There should be no space between the unit and the measurement

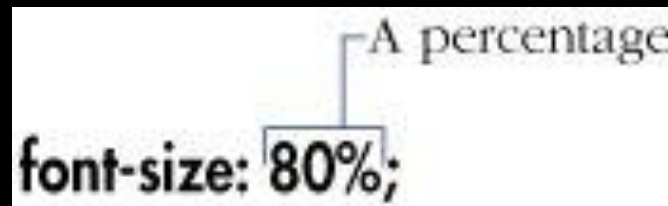


A Property's Value: *Lengths and Percentages*

- There are length types that are *relative* to other values.
- An *em* is usually equal to the element's font-size, so **2em** would mean "twice the font-size".
- The *ex* should be equal to the font's x-height, that is, the height of a letter x in the font, but it's not well supported.
- Pixels (px) are relative to the resolution of the monitor though not to other style rules

A Property's Value: *Lengths and Percentages*

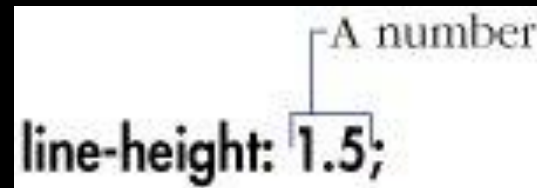
- There are also the largely self-explanatory *absolute* units inches (in), centimeters (cm), millimeters (mm), points (pt), and picas (pc).
- In general, you should only use absolute lengths when the size of the output is known.
- Percentages are generally relative to the parent element.



A diagram illustrating a percentage value in a CSS property. It shows the text "font-size: 80%;" with a bracket pointing to "80%" and the label "A percentage".

A Property's Value: *Bare Numbers*

- A very few CSS properties accept a value in the form of a number, without a unit, like 3.
- The most common are line-height and z-index.
- Don't confuse numbers and integers with length.
- A number or integer has no unit (like px).
- In this case, the value shown here is a factor that will be multiplied by the font-size to get the line-height.

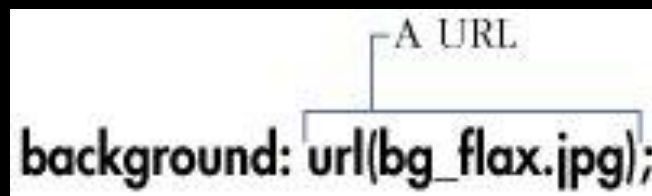


A number

line-height: 1.5;

A Property's Value: *URLs*

- Some CSS properties allow you to specify the URL of another file.
- In that case, use `url(file.ext)`, where *file.ext* is the path and file name of the desired document.
- Note that the specifications state that relative URLs should be relative to the style sheet and not the HTML document.
- URLs in CSS properties do not need to be enclosed in quotation marks.



A diagram illustrating a CSS property value. The text `background: url(bg_flax.jpg);` is shown. A bracket is drawn above the `url(bg_flax.jpg);` portion, with a line pointing to the label "A URL".

A Property's Value: *URLs*

- While you may use quotations around the file name, they're not required.
- On the other hand, there should be no space between the word `url` and the opening parentheses.
- White space between the parentheses and the address is allowed but not required.

A Property's Value: *CSS Colors*

- There are several ways to specify colors for CSS properties.
- First, and easiest, the value can be one of 16 predefined color names.
- Of course, 16 colors get pretty boring pretty quickly.

Sixteen Predefined Colors

Note that the color names and hex values are not case-sensitive, not even in XHTML.

| | | | |
|---|--|---|--|
|  Aqua #00FFFF |  Black #000000 |  Blue #0000FF |  Fuchsia #FF00FF |
|  Gray #808080 |  Green #008000 |  Lime #00FF00 |  Maroon #800000 |
|  Navy #000080 |  Olive #808000 |  Purple #800080 |  Red #FF0000 |
|  Silver #C0C0C0 |  Teal #008080 |  White #FFFFFF |  Yellow #FFFF00 |

A Property's Value: *CSS Colors*

- Instead of limiting yourself to those colors, you can construct your own by specifying the amount of red, green, and blue in the desired color.
- You can give the values of each of these contributing colors as a percentage, a number from 0255, or a hexadecimal representation of the number

The amount of
red in the color

color: rgb(%35, 0%, 50%)

The amount of
green in the color

The amount of
blue in the color

The amount of
red in the color

color: #59007F

A hash symbol

The amount of
green in the color

The amount of
blue in the color

The class Selector

- With the class selector you can define different styles for the same type of HTML element.
- Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph.
- Here is how you can do it with styles:

```
p.right {text-align: right}
p.center {text-align: center}
```

You have to use the class attribute in your HTML document:

```
<p class="right">
This paragraph will be right-aligned.
</p>

<p class="center">
This paragraph will be center-aligned.
</p>
```

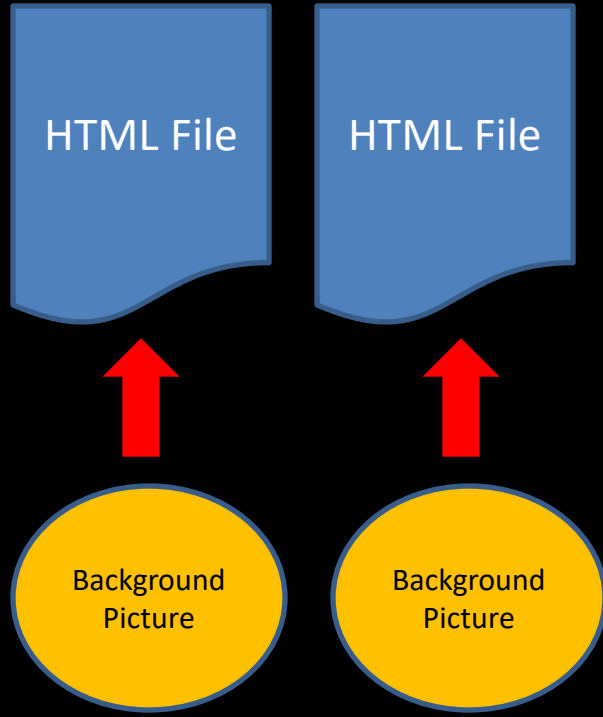
Add Styles to Elements with Particular Attributes

- You can also apply styles to HTML elements with particular attributes.
- The style rule below will match all input elements that have a type attribute with a value of "text":

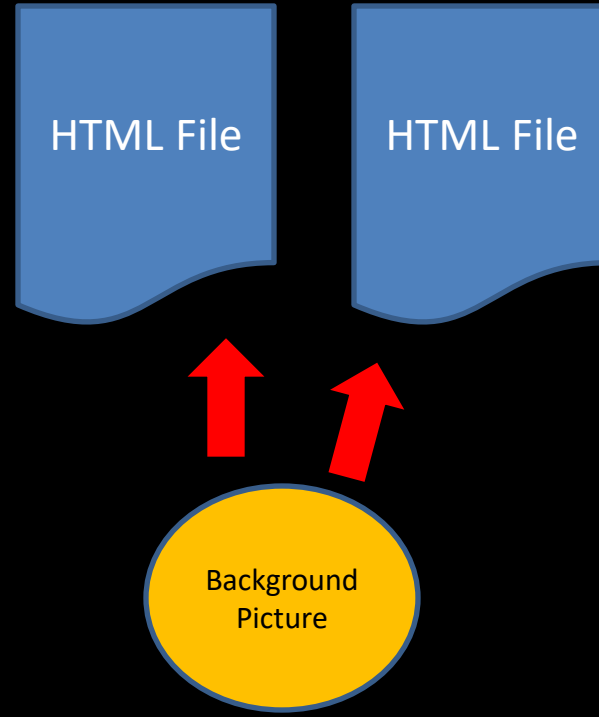
```
input[type="text"] {background-color: blue}
```

How to Insert a Style Sheet

- When a browser reads a style sheet, it will format the document according to it. There are three ways of inserting a style sheet:
 - External
 - Internal
 - Inline



Internal/inline



External

Internal Style Sheet

- An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section by using the <style> tag, like this:

```
<head>
<style type="text/css">
hr {color: sienna}
p {margin-left: 20px}
body {background-image: url("images/back40.gif")}
</style>
</head>
```


Inline Styles

- An inline style loses many of the advantages of style sheets by mixing content with presentation.
- Use this method sparingly, such as when a style is to be applied to a single occurrence of an element.
- To use inline styles you use the style attribute in the relevant tag.
- The style attribute can contain any CSS property.
- The example shows how to change the color and the left margin of a paragraph:

```
<p style="color: sienna; margin-left: 20px">  
This is a paragraph  
</p>
```

External Style Sheet

- An external style sheet is ideal when the style is applied to many pages.
- With an external style sheet, you can change the look of an entire Web site by changing one file.
- Each page must link to the style sheet using the <link> tag.
- The <link> tag goes inside the head section:

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

The browser will read the style definitions from the file `mystyle.css`, and format the document according to it.

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a `.css` extension. An example of a style sheet file is shown below:

```
hr {color: sienna}
p {margin-left: 20px}
body {background-image: url("images/back40.gif")}
```

Offering Alternate Style Sheets

- You can link to more than one style sheet and let visitors choose the styles they like best.
- **To offer alternate style sheets:**
 - To designate the style sheet that should be used as a base, regardless of the visitor's preferences, use the simple syntax with no title.
 - To designate the style sheet that should be offered as a first choice, but that can be deactivated by another choice, add **title="label"** to the link element, where *label* identifies the preferred style sheet.
 - To designate a style sheet that should be offered as an alternate choice, use **rel="alternate stylesheet" title="label"** in the link element, where *label* identifies the alternate style sheet

```
<head><meta http-equiv="content-type"
content="text/html; charset=utf-8" />
<title>Palau de la Música</title>
  <link rel="stylesheet" type="text/css"
href="base.css" />
  <link rel="stylesheet" type="text/css"
href="preferred.css" title="Dashed" />
  <link rel="alternate stylesheet" type="text/css"
href="alternate.css" title="Dotted" />
```



I love the Palau de la Música. It is ornate and gaudy and everything that was wonderful about modernism. It's also

Selecting Elements by ID or Class

- If you've labeled elements with an id or class, you can use that criteria in a selector to apply formatting to only those elements that are so labelled. The division with an id of gaudi encloses almost the entire page (everything but the initial h1).

```
<h1>Antoni Gaudí</h1>  
<div id="gaudi">  
  <p>Many tourists are drawn to Barcelona to see  
  Antoni Gaudí's incredible architecture. </p>  
  <p>Barcelona <a href="http://  
  www.gaudi2002.bcn.es/english/">celebrates  
  </a> the 150th anniversary of Gaudí's birth in  
  2002.</p>  
  <div class="works"><h2>La Casa Milà</h2>  
  <p>Gaudí's work was essentially useful. La Casa  
  Milà is an apartment building and <em>real  
  people</em> live there.</p>  
  </div>  
  <div class="works"><h2>La Sagrada  
  Família</h2>
```

To select elements to format based on their id:

1. Type # (a hash or pound sign).
2. With no intervening space, immediately type **id**, where *id* uniquely identifies the element to which you'd like to apply the styles.

Example: This selector will choose the div element with an id equal to "gaudi".

```
div#gaudi {color:red;}
```

You can use class and id selectors alone or together with other selector criteria. For example, **.news {color: red;}** would affect all elements with the news class, while **h1.news {color: red;}** would affect only the h1 elements with the news class.

CSS Example

- All Examples
 - http://www.w3schools.com/css/css_examples.asp
- CSS Background
 - http://www.w3schools.com/css/css_background.asp
- CSS Text
 - http://www.w3schools.com/css/css_text.asp
- CSS Font
 - http://www.w3schools.com/css/css_font.asp
- CSS Border
 - http://www.w3schools.com/css/css_border.asp

CSS Example

- CSS Outline
 - http://www.w3schools.com/css/css_outline.asp
- CSS Margin
 - http://www.w3schools.com/css/css_margin.asp
- CSS Padding
 - http://www.w3schools.com/css/css_padding.asp
- CSS List
 - http://www.w3schools.com/css/css_list.asp
- CSS Table
 - http://www.w3schools.com/css/css_table.asp

Example 1

```
<html>
<head>
<style type="text/css">
span.highlight
{
background-color:yellow
}
</style>
</head>

<body>
<p>
<span class="highlight">This is a text.</span> This is a text. This is a
text. This is a text. This is a text. This is a text. This is a text.
This is a text. This is a text. <span class="highlight">This is a
text.</span>
</p>
</body>
</html>
```

The HTML file below links to an external style sheet with the <link> tag:

```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="ex2.css" />
</head>

<body>

<h1>This is a header 1</h1>
<hr />

<p>You can see that the style
sheet formats the text</p>

<p><a href="http://www.w3schools.com"
target="_blank">This is a link</a></p>

</body>
... ..
```

This is the style sheet file (ex2.css):

```
body {background-color: tan}
h1 {color:maroon; font-size:20pt}
hr {color:navy}
p {font-size:11pt; margin-left: 15px}
a:link      {color:green}
a:visited   {color:yellow}
a:hover     {color:black}
a:active    {color:blue}
```

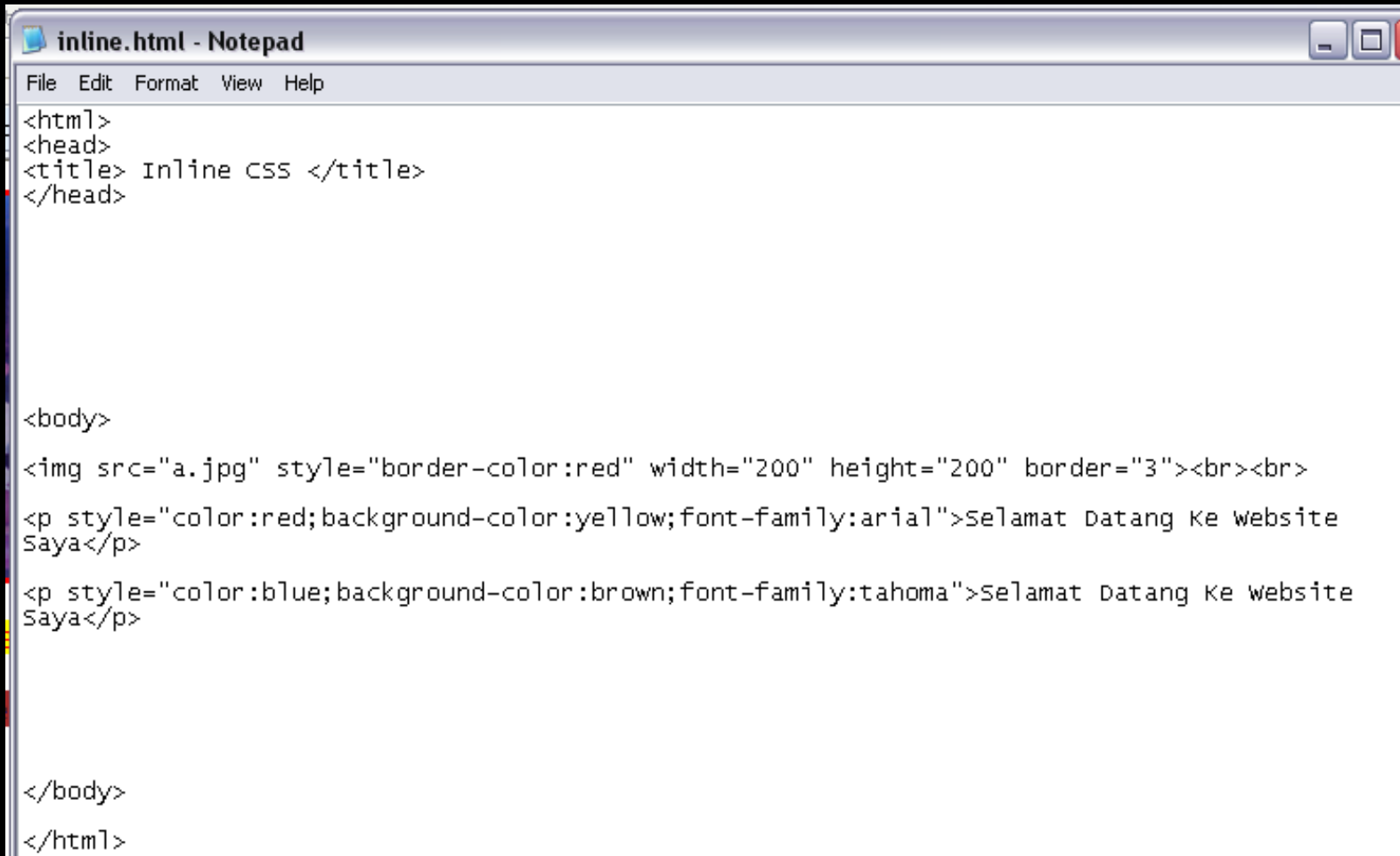
The result is in the frame below:

This is a header 1

You can see that the style sheet formats the text

[This is a link](http://www.w3schools.com)

Exercise 1



```
inline.html - Notepad
File Edit Format View Help

<html>
<head>
<title> inline CSS </title>
</head>

<body>

<br><br>

<p style="color:red;background-color:yellow;font-family:arial">Selamat Datang Ke website
Saya</p>

<p style="color:blue;background-color:brown;font-family:tahoma">Selamat Datang Ke website
Saya</p>

</body>

</html>
```

Exercise 2

```
a - Notepad
File Edit Format View Help
<html>
<head><title>Manipulating CSS</title></head>

<body style="background-color:black">




<h1 style="color:red;text-align:right;font-family:arial;letter-spacing:20">This is Header
1</h1>
<h1 style="color:blue;text-align:center;font-family:tahoma;font-style:italic">This is
Header 2</h1>
<h1 style="color:blue;text-align:right;font-family:magneto;font-style:oblique">This is
Header 3</h1>

<p style="text-decoration:overline;color:gold">This is first paragraph</p>
<p style="text-decoration:underline;color:gold">This is second paragraph</p>
<p style="text-decoration:line-through;color:red">This is third paragraph</p>

<a href="index.html" style="text-decoration: underline overline;color: green">Click Me For
Navigation</a>

</body>
</html>
```