Cascading Style Sheets (CSS):

- is a simple mechanism for *adding style* (e.g. fonts, colors, layouts) to Web documents.
- Styles provide *powerful control* over the **presentation** of web pages.

The Purpose of CSS

- If HTML is the content and meaning
  - ➞CSS helps to convey that meaning
- Allows developers to separate the content from layout and design
  - Content and design inherently different in nature
  - ➞ Change in content does not require change in design

CSS Benefits

- Separates structure from presentation
- Provides advanced control of presentation
- Easy maintenance of multiple pages
- Faster page loading
- Better accessibility for disabled users
- Easy to learn

HTML & CSS

- HTML and CSS work together to produce beautiful and functional Web sites
- HTML = structure
- CSS = *style*

Three Different Scopes of CSS

- Local
  - confined to a **single element (tag)**
- Internal
  - affect elements in an **entire page**
- External
  - can affect **multiple pages**
- **Precedence**
  - **Local > Internal > External**

Attaching a Style Sheet

Attach a style sheet to a page by adding the code to the <head> section of the HTML page. There are ***3 ways*** to attach CSS to a page:

1. External Style Sheet: Best used to control styling on multiple pages.

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

2. Internal Style Sheet: Best used to control styling on one page.

```
<style type="text/css">
h1 {color: red)
</style>
```

3. Inline Style Sheet*: CSS is not attached in the <header> but is used directly within HTML tags.

```
<p style="color: red;">Some Text</p>
```

Local/Inline Style Sheet

- **Example**

```
<h1 style="color:white; background-color:orange;">
Local Style Sheet Applied to Header 1
</h1>
```

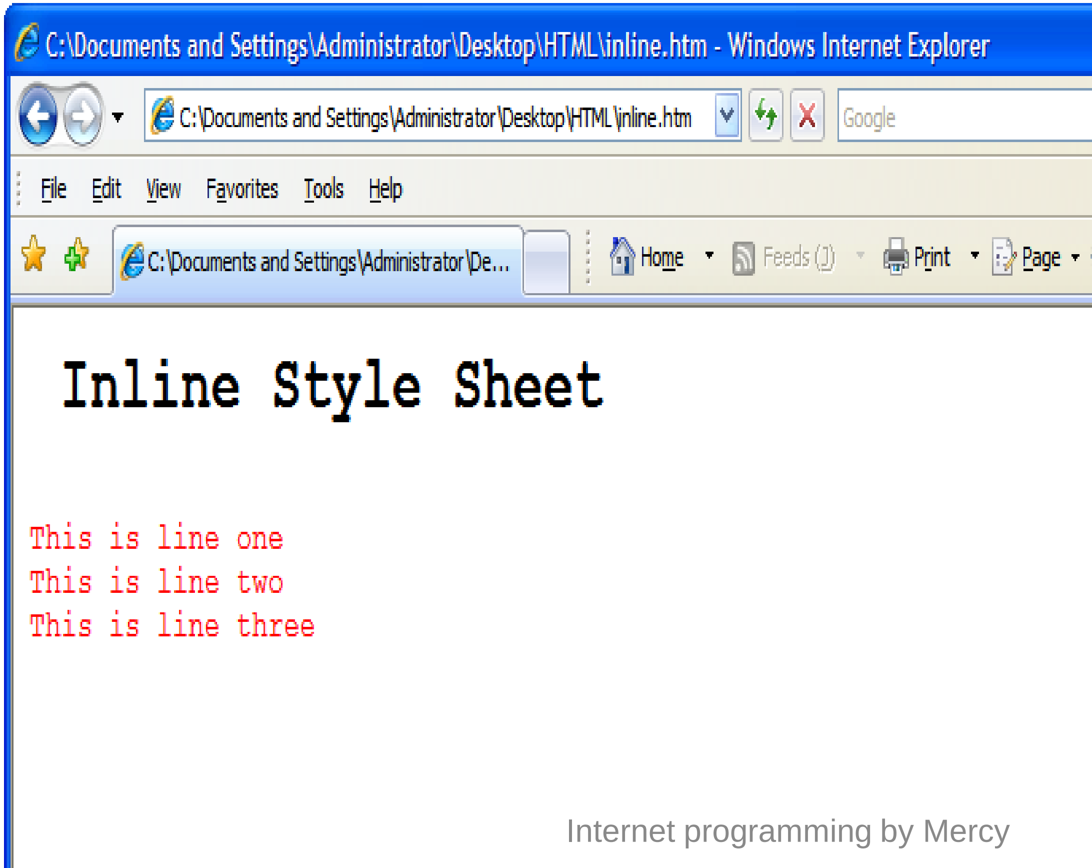
- **Practice**

1. add “text-align” property to make it centered
2. add “border” property to let it has black, 1px thick, solid border at left, right, top, and bottom

- **Tip**: use “border: <top> <right> <bottom> <left>;” format for 4 sides; use “border-<side>: xx yy zz;” for a particular side, where <side> can be left, right, top or bottom. Can apply to other similar properties.

Inline Style Sheet(contd.)

```
<html>
<body>
<pre>
<h1> Inline Style Sheet</h1>
<p style="font-size: 12px; color: red">
This is line one
This is line two
This is line three
</p>
</body>
</html>
```


C:\Documents and Settings\Administrator\Desktop\HTML\inline.htm - Windows Internet Explorer
C:\Documents and Settings\Administrator\Desktop\HTML\inline.htm
Google
File
Edit
View
Favorites
Tools
Help
C:\Documents and Settings\Administrator\De...
Home
Feeds (J)
Print
Page

Inline Style Sheet

```
This is line one
This is line two
This is line three
```

Example:

```
<!doctype html>
<head></head>
<body>

<h1 style=“color:white; background-color:orange; border:black 3px solid;”>
   Local Style Sheet Applied to Header 1
</h1>
<p style=“font-size:24px; font-family:times; height:200px; width:40%;
           background-color:aqua;”>
 Local style applied in the paragraph
</p>

</body>
</html>
```

EXERCISE NO5: CREATING A WEB PAGE USING A TABLE

```
<html>
<head>
<title>Untitled Document</title>
</head>
<body>
<table border="1" cellspacing="0" cellpadding="0">
 <tr>
   <td width="160" valign="top"><p><strong>Nom</strong></p></td>
   <td width="147" valign="top"><p><strong>Date</strong></p></td>
   <td width="200" valign="top"><p><strong>Microprocesseur</strong></p></td>
   <td width="190" valign="top"><p><strong>RAM</strong></p></td>
</tr>
 <tr>
   <td width="160" valign="top"><p> Pentium   I</p></td>
   <td width="147" valign="top"><p>1993-1996</p></td>
   <td width="200" valign="top"><p>60 or 90  upto 120    mégahertz</p></td>
   <td width="190" valign="top"><p>8 MB, 16 MB or 32MB</p></td> </tr> <tr>
   <td width="160" valign="top"><p>Pentium II</p></td>
   <td width="147" valign="top"><p>1997-1999</p></td>
   <td width="200" valign="top"><p>233 or 333 mégahertz</p></td>
   <td width="190" valign="top"><p>16 MB, 32 MB or 64MB</p></td></tr> <tr>
   <td width="160" valign="top"><p>Pentium III</p></td>
   <td width="147" valign="top"><p>1999-2002</p></td>
   <td width="200" valign="top"><p>450 up to 900 mégahertz</p></td>
   <td width="190" valign="top"><p>32 MB, 64 MB or 128MB</p></td></tr><tr>
   <td width="160" valign="top"><p>Pentium    IV</p></td>
   <td width="147" valign="top"><p>1999-2006</p></td>
   <td width="200" valign="top"><p>1,5 Gigahertz</p></td>
   <td width="190" valign="top"><p>64 MB, 128 MB or 256MB</p></td></tr><tr>
    </table>
</body>
</html>
```

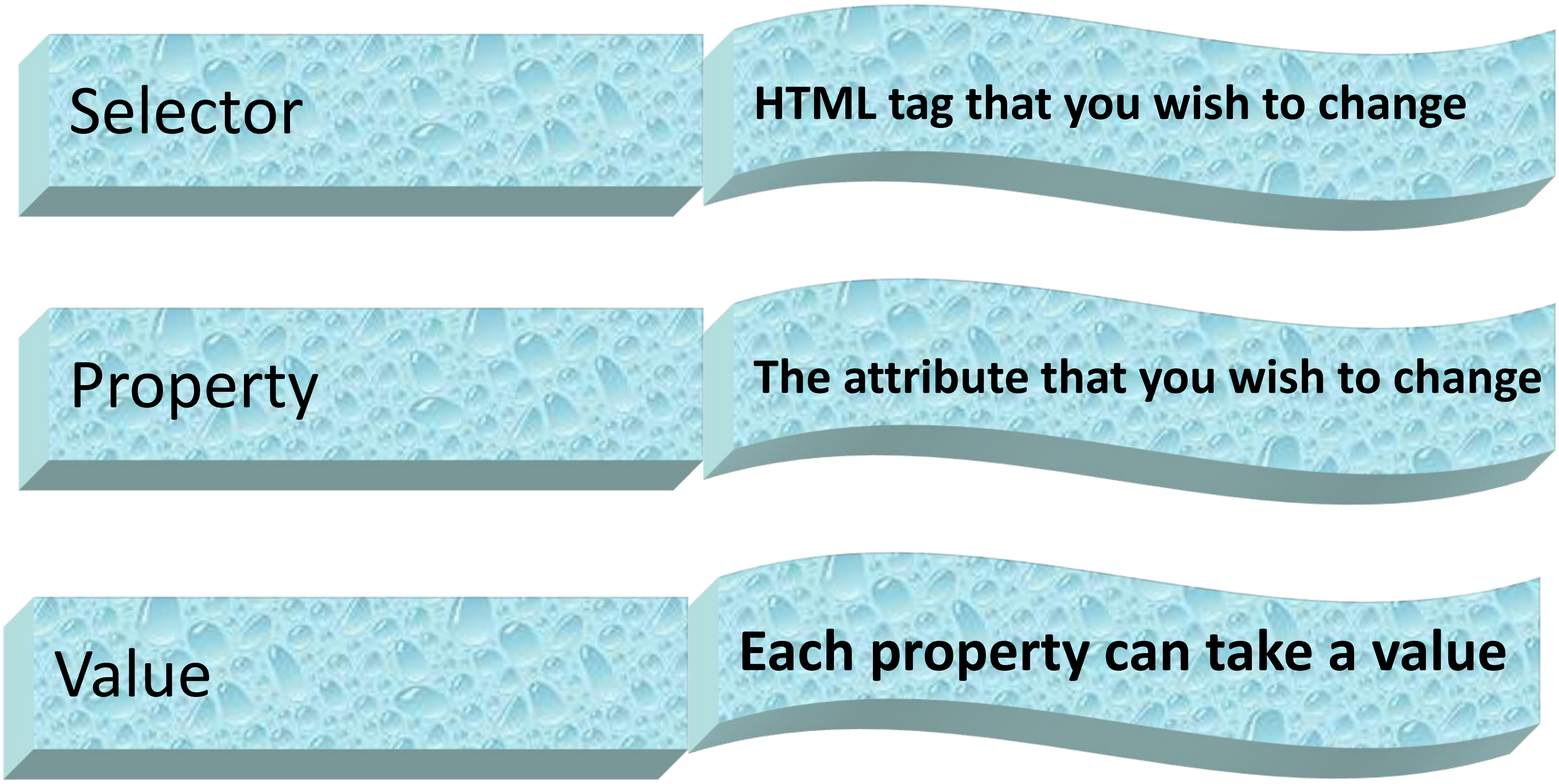
Style Sheet(Internal/External)

- A style sheet consists of a set of ***rules***.
- Each ***rule*** consists of one or more ***selectors*** and a ***declaration block***.
- A ***declaration block*** consists of a list of ***declarations*** in curly braces ({}).
- Each ***declaration*** consists of a **property**, a **colon** (**:**), a **value**, then a **semi-colon** (**;**).

selector {property: value;}

Parts of a Cascading Style Sheet

- CSS syntax is made up of three parts:



Selector

HTML tag that you wish to change

Property

The attribute that you wish to change

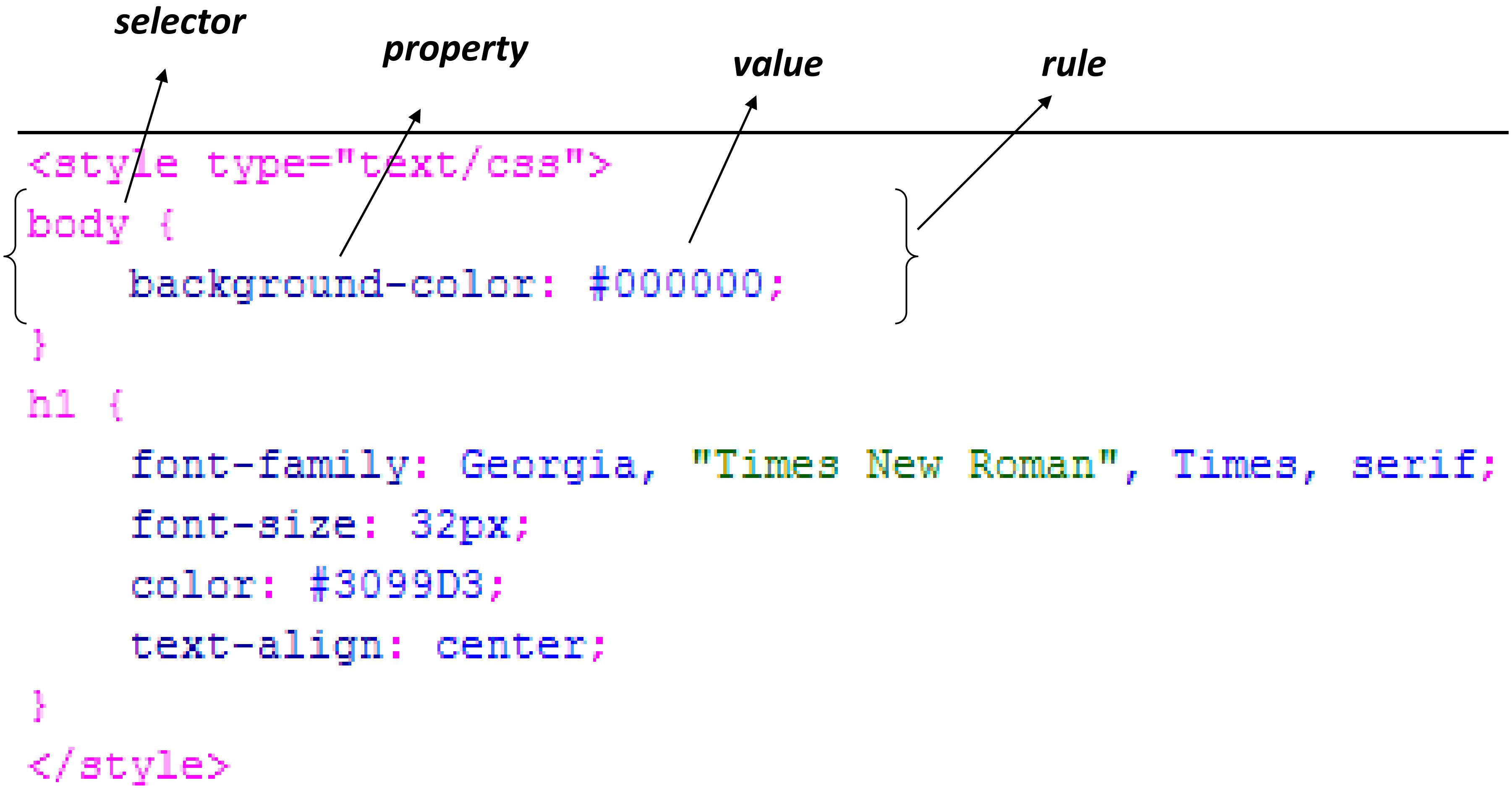
Value

Each property can take a value

Internal Style Sheet

- How to create?
  - Put **<style> </style>** tag between <head> and </head> tags of your HTML page
  - Use type attribute to indicate the style sheet type, usually ***type="text/css***"
  - Put your set of style sheet ***rules*** in between ***<style> and </style>*** tags

Style Sheet Syntax Explained



```
<style type="text/css">
body {
    background-color: #000000;
}
h1 {
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: 32px;
    color: #3099D3;
    text-align: center;
}
</style>
```

Syntax of Style sheet

selector {property: value}

Example

p {color: green}

If the value consist of more than one word the value has to be embedded within quotes.

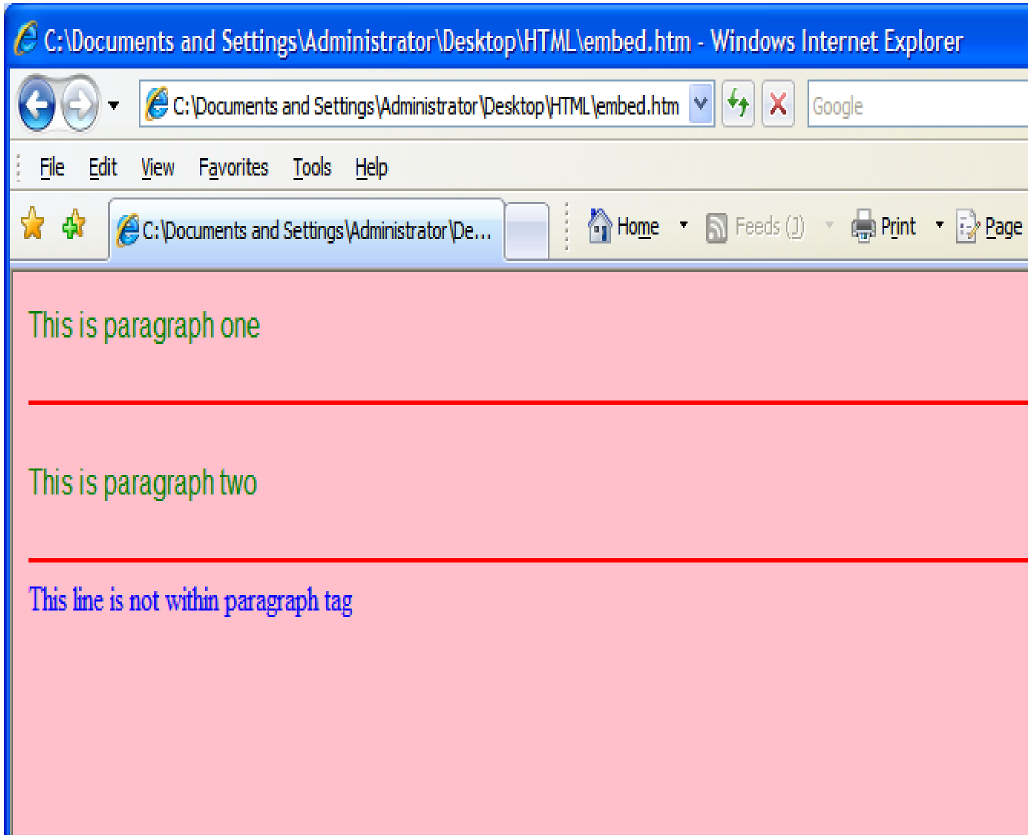
p {font-family: “sans-serif”}

If more than one property needs to be defined then the properties need to be separated by semi colon.

p {font-family: “sans-serif” ; color: green}

Example of an internal Style sheet cont'd

```
<html>
<head>
<style type="text/css">
p {font-family:"sans-serif"; color:
    green}
hr {color: red}
body {color:blue; background-
    color:pink}
</style>
</head>
<body>
<p>This is paragraph one</p>
<hr>
<p> This is paragraph two</p>
<hr>
This line is not within paragraph tag
</body>
</html>
```



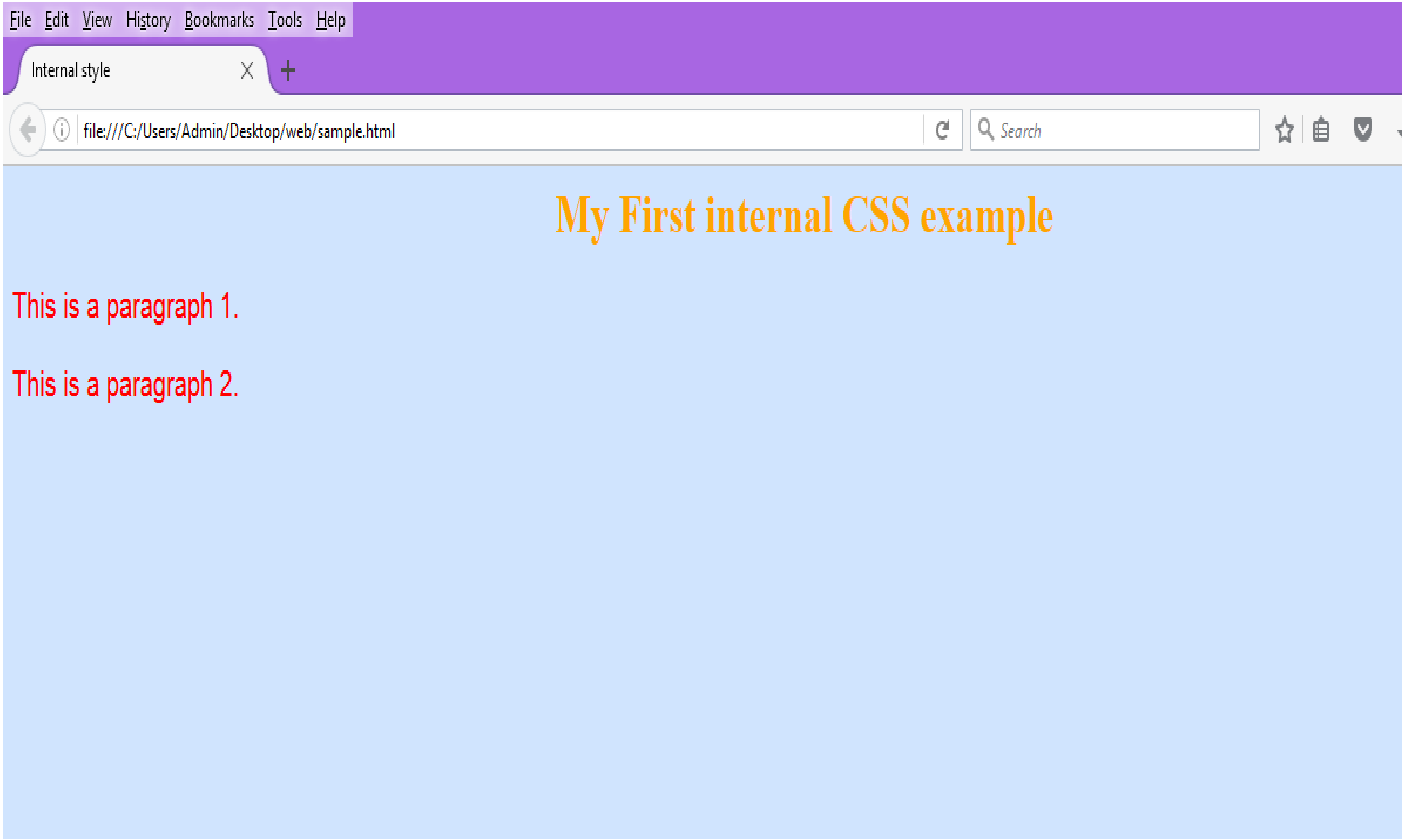
C:\Documents and Settings\Administrator\Desktop\HTML\embed.htm - Windows Internet Explorer
C:\Documents and Settings\Administrator\Desktop\HTML\embed.htm
Google
File
Edit
View
Favorites
Tools
Help
C:\Documents and Settings\Administrator\De...
Home
Feeds (J)
Print
Page

This is paragraph one

This is paragraph two

This line is not within paragraph tag

```
<html>
<head>
<title>Internal style</title>
<style>
body { background-color: #d0e4fe; }

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

p {  color:red;
     font-family: arial;
     font-size: 20px;
   }
</style>
</head>
<body>

<h1>My First  internal CSS  example</h1>
<p>This is a paragraph 1.</p>
<p>This is a paragraph 2.</p
</body>
</html>
```



My First internal CSS example

This is a paragraph 1.

This is a paragraph 2.

Grouping Selectors

Group **different selectors** with the same declaration on one line.

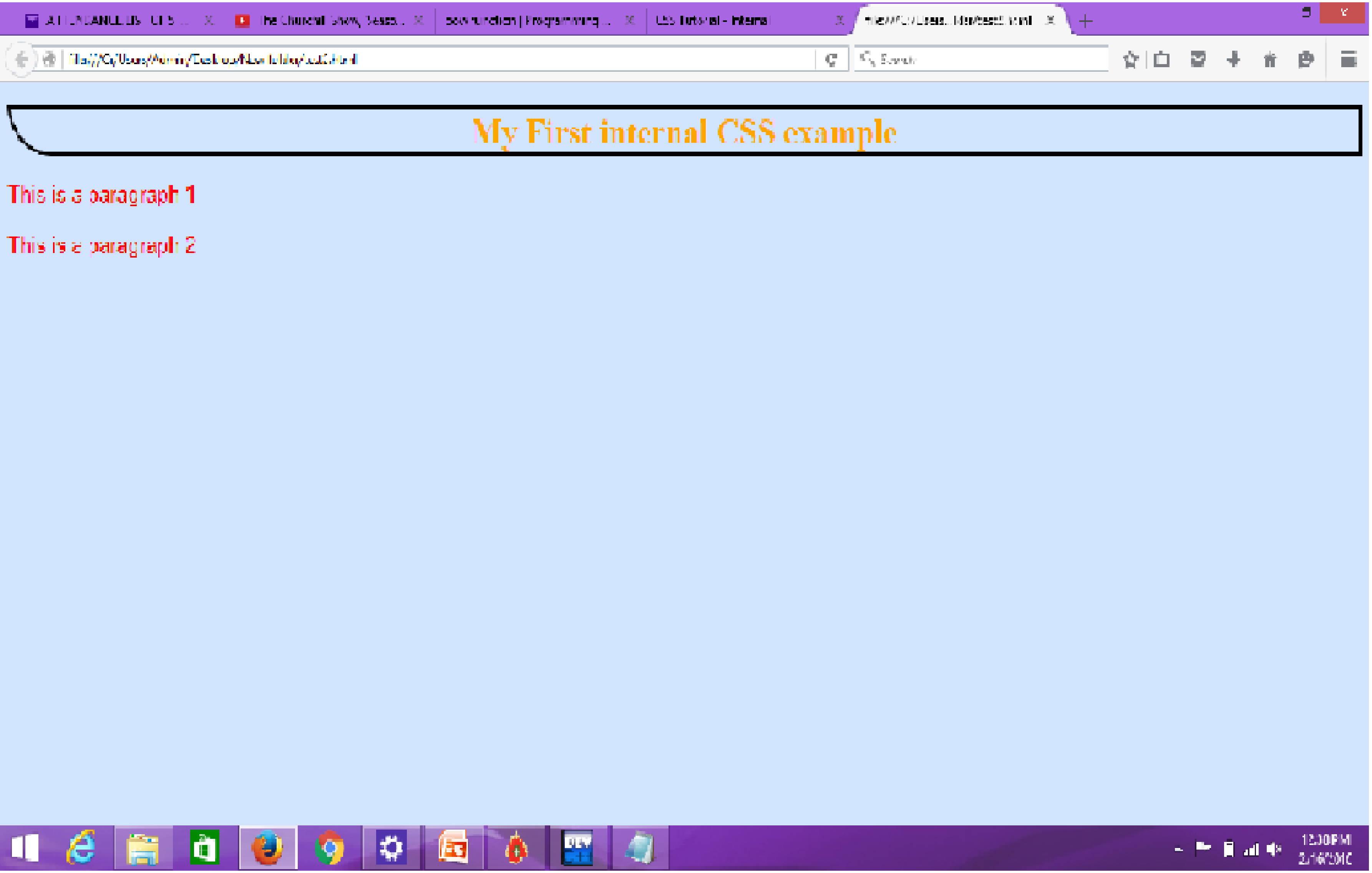
```
h1 {color: yellow;}
h2 {color: yellow;}
h3 {color: yellow;}
```

Example of grouping selectors (both are correct):

```
h1, h2, h3 {color: yellow;}
```

Adding border

```
<html>
<head>
<style>
body { background-color: #d0e4fe; }

h1 { color: orange;
       text-align: center;
       border:3px solid;
        border-bottom-left-radius:2em;
     }

p {  color:red;
      font-family: arial;
      font-size: 20px;
   }
</style>
</head>
<body>

<h1>My First  internal CSS  example</h1>
<p>This is a paragraph 1.</p>
<p>This is a paragraph 2.</p
</body>
</html>
```



My First internal CSS example

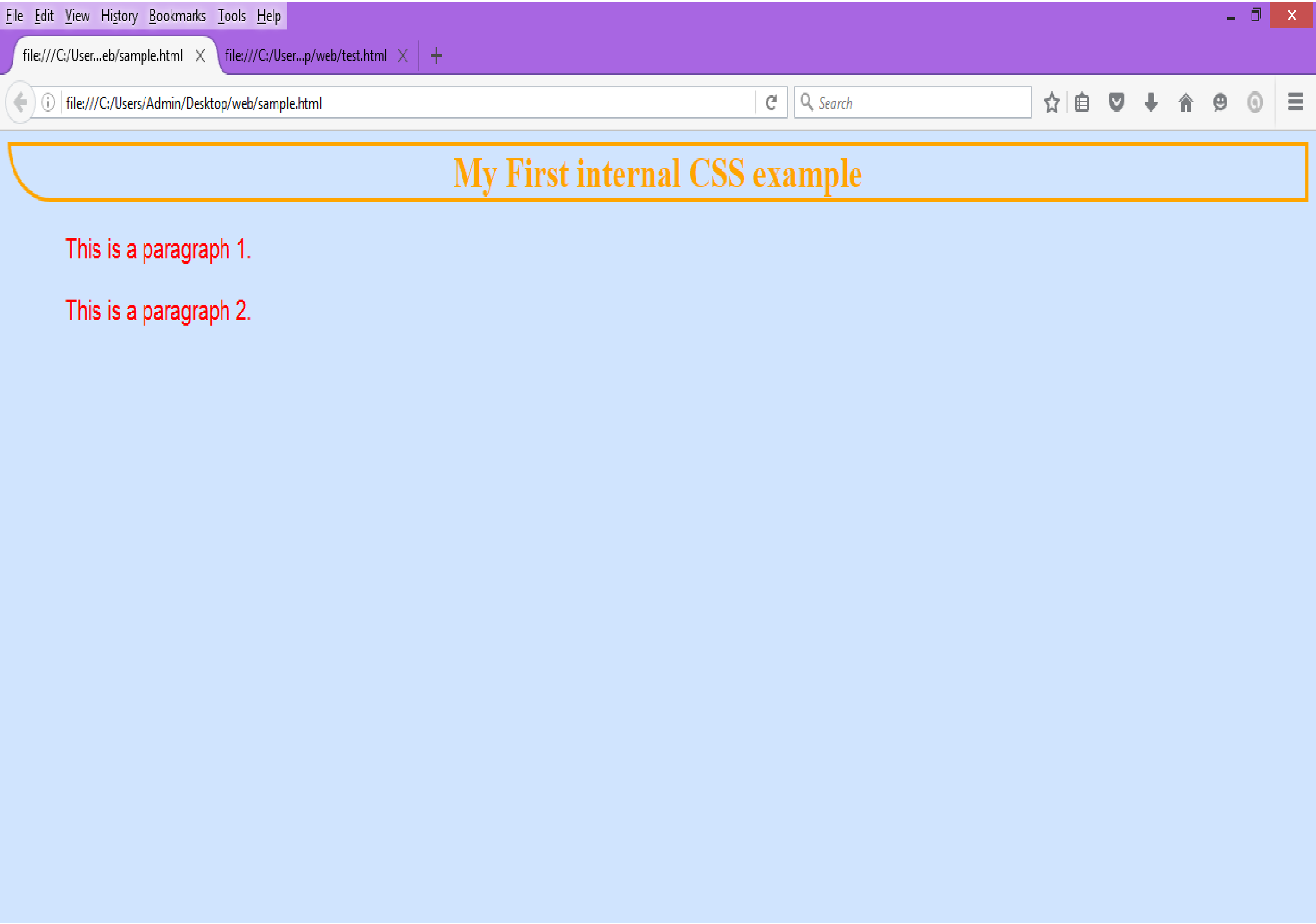
This is a paragraph 1

This is a paragraph 2

- The text-indent property is used to specify the indentation of the first line of a text:
- To indent every paragraph automatically Here's the CSS code:
- **p { text-indent: 3em; }**

Adding indent to paragraphs

```
<html>
<head>
<style>
body { background-color: #d0e4fe; }

h1 { color: orange;
     text-align: center;
     border:3px solid;
      border-bottom-left-radius:2em;
    }

p {  color:red;
     font-family: arial;
     font-size: 20px;
     text-indent:3em;
   }
</style>
</head>
<body>
<h1>My First  internal CSS  example</h1>
<p>This is a paragraph 1.</p>
<p>This is a paragraph 2.</p
</body></html>
```



File Edit View History Bookmarks Tools Help
file:///C:/User...eb/sample.html
file:///C:/User...p/web/test.html
file:///C:/Users/Admin/Desktop/web/sample.html
Search

My First internal CSS example

This is a paragraph 1.

This is a paragraph 2.

CSS fonts

Text color

- Color:#036;
- Color:Red;

Font face

- font-family: "Times New Roman";
- font-family: serif;

- font-style: italic;
- font-style: normal;

- font-size: 14px;
- font-size: 100%;

- font-weight: bold;
- font-weight: normal;

Text-Decoration

- text-decoration:underline
- text-decoration:none;
- text-decoration:overline;
- text-decoration:line-through;

Text-Transformation

- text-transformation:uppercase;
- text-transformation:lowercase;
- text-transformation:capitalize;

CSS Text

Text Alignment

- The text-align property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centered, or justified.

```
text-align: center;
text-align: left;
text-align: right;
text-align: justify;
```

Line Height

- The line-height property is used to specify the space between lines:

```
line-height: 1.8;
```

CSS Background, Image and Color Styles

- You are probably familiar with the <body> tag. A typical <body> tag looks something like this:
- <body background="graphic.jpg" text="#FFFFFF" bgcolor="#000000">
- To convert that into CSS, it looks like this:

```
body { background-image: url(graphic.jpg);
    color: #FFFFFF; background-color: #000000; }
```

Border (top, right, bottom, left)

You can define the entire border or only the top, bottom, left, or right. You can also define the border using one declaration. The code could be any of the following:

```
border: red dotted 1px;
```

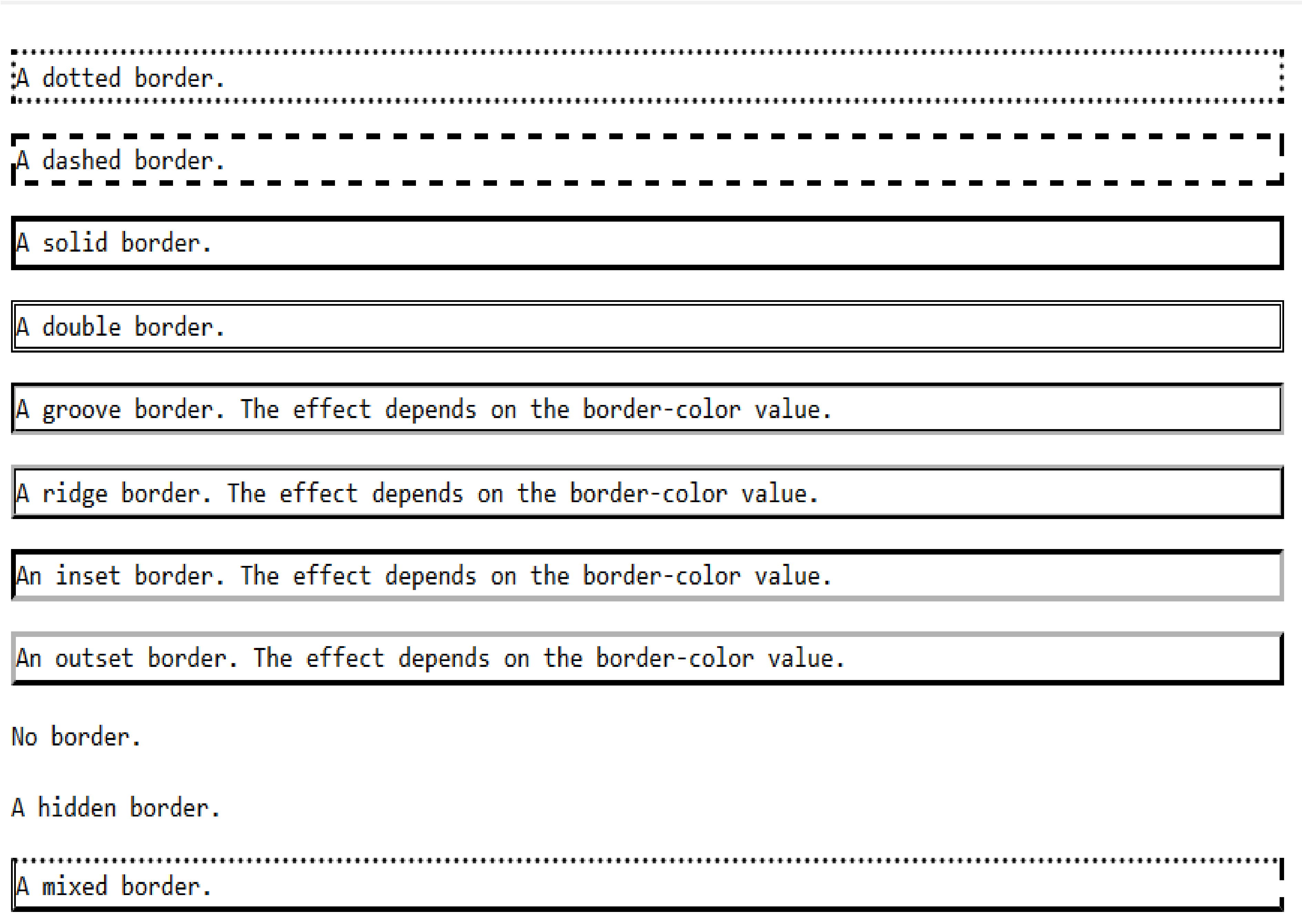
```
border-color: red;
border-style: dotted;
border-width: 2px;
```

```
Border-radius: 2em;
```

```
border-top: red dotted 1px;
border-bottom: red dotted 1px;
border-left: red dotted 1px;
border-right: red dotted 1px;
```

Border Style

- The border-style property specifies what kind of border to display.
- The following values are allowed:
  - ✓ ***dotted*** - Defines a dotted border
  - ✓ ***dashed*** - Defines a dashed border
  - ✓ ***solid*** - Defines a solid border
  - ✓ ***double*** - Defines a double border
  - ✓ ***groove*** - Defines a 3D grooved border. The effect depends on the border-color value
  - ✓ ***ridge*** - Defines a 3D ridged border. The effect depends on the border-color value
  - ✓ ***inset*** - Defines a 3D inset border. The effect depends on the border-color value
  - ✓ ***outset*** - Defines a 3D outset border. The effect depends on the border-color value
  - ✓ ***none*** - Defines no border
  - ✓ ***hidden*** - Defines a hidden border
- The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).



A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

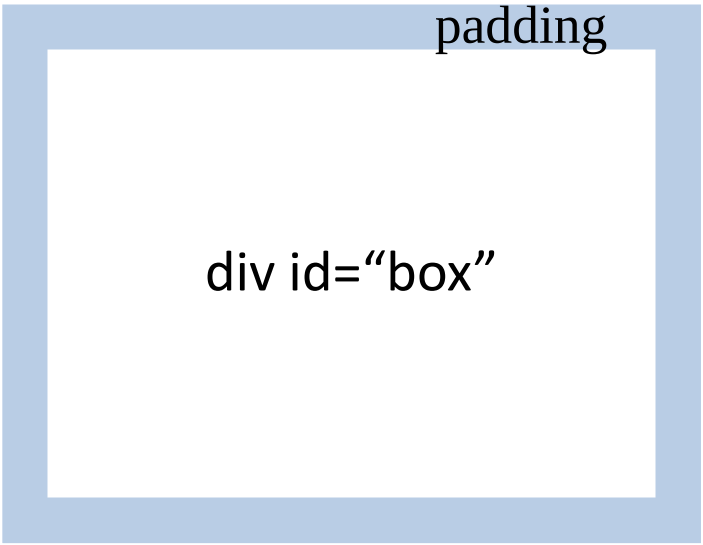
CSS Padding Properties

- The CSS padding properties are used to generate space around content.
- The padding properties set the size of the white space between the element content and the element border.
- ***The padding clears an area around the content (inside the border) of an element***.

Padding (top, right, bottom, left)

You can use padding for all around the element or specify each side of the rectangle separately.

The code could be any of the following:



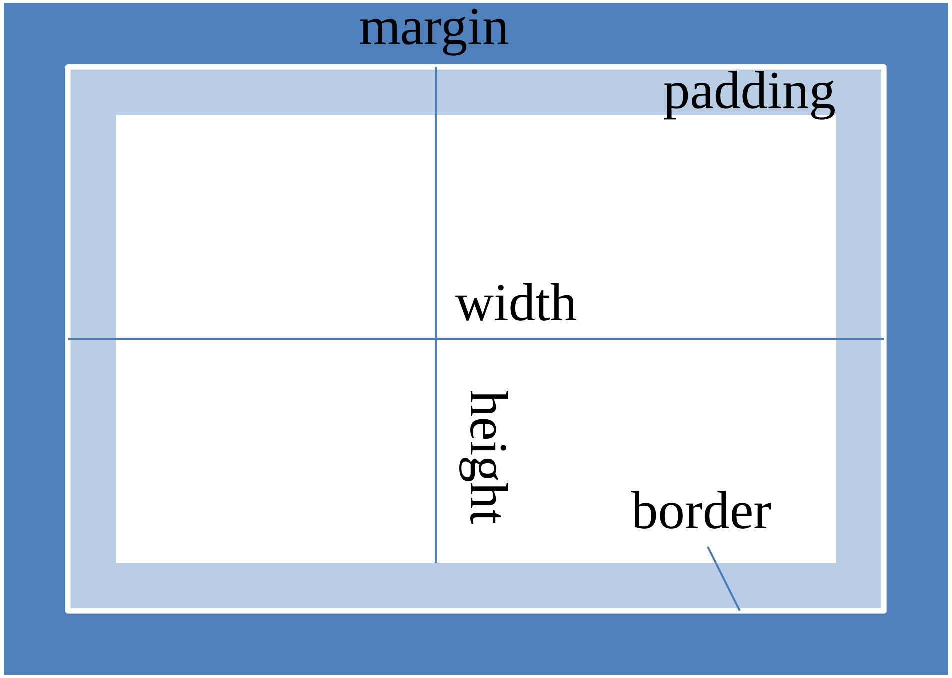
padding

div id="box"

```
padding: 10px;


Padding: 10px 10px;


padding: 10px 10px 10px 10px;


padding-left: 10px;
padding-right: 10px;
padding-bottom: 10px;
padding-top: 10px;
```

CSS Margin Properties

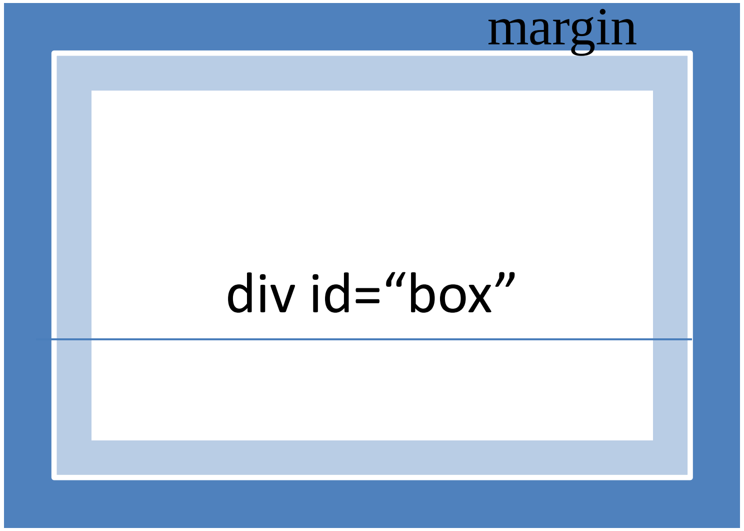
- The CSS margin properties are used to generate space around elements.
- The margin properties set the size of the white space OUTSIDE the border.



Margin (top, right, bottom, left)

You can use margin for all around the element or specify each side of the rectangle separately.

The code could be any of the following:



```
margin: 10px;
```

or

```
margin: 10px 10px;
```

or

```
margin: 10px 10px 10px 10px;
```

or

```
margin-left: 10px;
margin-right: 10px;
margin-bottom: 10px;
margin-top: 10px;
```

The display Property

- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Block-level Elements

- A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

Examples of block-level elements:

  - <div>
  - <h1> - <h6>
  - <p>
  - <form>

Inline Elements

- An inline element does not start on a new line and only takes up as much width as necessary.

Examples of inline elements:

  - <span>
  - <a>
  - <img>

The position Property

- The position property specifies the type of positioning method used for an element (static, relative, fixed or absolute).

There are four different position values:

  - static
  - relative
  - fixed
  - absolute

- Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

position: static;

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

```
div.static {
        position: static;
        border: 3px solid #73AD21;}
```

position: relative;

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

```
div.relative {
    position: relative;
    left: 30px;
    border: 3px solid #73AD21;
```

position: fixed;

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.

```
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
 }
```

position: absolute;

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
- **Note:** A "positioned" element is one whose position is anything except static.

Here is a simple example:

```
div.relative {
        position: relative;
        width: 400px;
        height: 200px;
        border: 3px solid #73AD21;
    }
div.absolute {
        position: absolute;
        top: 80px;
        right: 0;
        width: 200px;
        height: 100px;
        border: 3px solid #73AD21;
    }
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
```

id and class

- **ID selectors are used for special formatting of only a few elements**. ID selectors use a # .Imagine 3 paragraphs, we'll call them par1, par2, par3. We'll make one red, one orange, and one blue.

id and class

- We could use these styles:

```
#par1 { color: red; }
#par2 { color: orange; }
#par3 { color: blue; }

<p id="par1">I'm in red</p>
<p id="par2">I'm in orange</p>
<p id="par3">I'm in blue</p>
```

id and class

- **class selectors are used for special formatting of only a few elements**. class selectors use a (.) period.Imagine 3 paragraphs, we'll call them par1, par2, par3. We'll make one red, one orange, and one blue.

id and class

- We could use these styles:

```
.par1 { color: red; }
.par2 { color: orange; }
.par3 { color: blue; }

<p class="par1">I'm in red</p>
<p class="par2">I'm in orange</p>
<p class="par3">I'm in blue</p>
```

IDs and Classes

- **IDs (#)** are unique and can only be used once on the page
- **Classes (.)** can be used as many times as needed

HTML Code:

```
<h1 id=“mainHeading”>Names</h1>
<p class=“name”>Joe</p>
```

CSS Code:

```
#mainHeading {color: green}
.name {color: red}
```

CSS Background, Image and Color Styles

- You are probably familiar with the <body> tag. A typical <body> tag looks something like this:
- <body background="graphic.jpg" text="#FFFFFF" bgcolor="#000000">
- To convert that into CSS, it looks like this:

```
body { background-image: url(graphic.jpg);
    color: #FFFFFF; background-color: #000000; }
```

HTML Links

- HTML links are hyperlinks.
- A hyperlink is a text or an image you can click on, and jump to another document.
- HTML use anchor tags to create links. Text, Images, and Forms may be used to create these links.

Syntax

- Links are defined with the **<a>** tag:

  <a href="*url*">*link text***</a>**

(href):

- The **href** attribute specifies the destination address. Basically this is where the user will be taken if they wish to click this link.

Text Links:

- The **link text** is the visible part.

- Hypertext references can be Internal, Local, or Global.
  - Internal - Links to anchors on the current page
  - Local - Links to other pages within your domain
  - Global - Links to other domains outside of your site

Code:

Internal - href="#anchorname“

Local - href="../pics/picturefile.jpg"

Global - href="http://www.tizag.com/"

The target Attribute of link element

- The target attribute defines whether to open the page in a separate window, or to open the link in the current browser window.
- The target attribute is placed inside the <a> tag

HTML Code:

- target= "_blank“ Opens new page in a new
  browser window
  “ _self“ Loads the new page in current window
  “ _parent“ Loads new page into a frame that
  is superior to where the link lies
  _top“ Loads new page into the current browser
  window, cancelling all frames

Anchors

- To link to sections of your existing page a name must be given to the anchor.
- In the example below, we've created a mini Table of Contents for this page. By placing blank anchors just after each heading, and naming them, we can then create reference links to those sections on this page as shown below.
- First, the headings of this page contain blank, named anchors. They look like this.

Internal links

Example:

```
<h2>HTML Links and Anchors<a name="top"></a></h2>
<h2>HTML Text Links<a name="text"></a></h2>
<h2>HTML Email<a name="email"></a></h2>
```

Now create the reference links, placing the pound symbol followed by the name of the anchor in the href of the new link.

Anchor Code:

```
<a href="#top">Go to the Top</a>
<a href="#text">Learn about Text Links</a>
<a href="#email">Learn about Email Links</a>
```

Local Links:

Go to the Top

Learn about Text Links

Learn about Email Links

Local link

Example: link1.html

```
<html>
 <head>
   <title>HTML local links </title>
 </head>
 <body>
<h1 >Local Link</h1>
<p>Links to other pages within your
domain</p>
<p>Click
<a href=“link2.html” > here </a>
   to go next page
</p>
 </body>
</html>
```

Example: link2.html

```
<html>
 <head>
   <title>HTML local links
</title>
 </head>
 <body>
<h1 >Welcome</h1>
<p>page was linked from
      link 1
 </p>
 </body>
</html>
```

Default attributes

- The browser has default settings for text color, link color, active link color and visited link color in addition to the background color. The defaults are...
- Text is usually black
  Links are usually blue
  Active link is usually red
  Visited link is usually purple
- You can change these if you NEED to.

Styling Links

The links property defines how inactive, hovered, active, and visited link states appear to the user.

```
a:link {color: red; text-decoration:
none;border-bottom: 1px dashed red;
background: white;}

a:visited {color: yellow;}

a:active {color: green;}

a:hover {color: orange;}
```

```
<html>
<head>
<style>
a:link {color: red; text-decoration: none;
  background: white;}
a:visited {color: yellow;}
a:active {color: green;}
a:hover {color: orange;}
<style>
</head>
<body>
<a href=“”>HOME</a>
<a href=“”>CONTACT</a>
</body>
```

HTML Lists

An unordered list and an ordered list in HTML

Unordered HTML Lists

- 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):

```
<!DOCTYPE html>
<html>
<body>
<h2>Unordered List with Default
Bullets</h2>
<ul>
  <li>Orange</li>
  <li>Banana</li>
  <li>Mango</li>
</ul>
</body>
</html>
```

Unordered List with Default Bullets

•Orange

•Banana

•Mango

CSS list-style-type Property

- A list-style-type: property can be added to an **unordered list**, to define the style of the marker:

Style	**Description**
list-style-type:disc	The list items will be marked with bullets (default)
list-style-type:circle	The list items will be marked with circles
list-style-type:square	The list items will be marked with squares
list-style-type:none	The list items will not be marked

Ordered HTML Lists

- An ordered list starts with the **<ol>** tag. Each list item starts with the **<li>** tag.
- The list items will be marked with numbers

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

<h2>Ordered List</h2>

<ol>
  <li>Banana</li>
  <li>Mango</li>
  <li>Orange</li>
</ol>

</body>
</html>
```

Ordered List

1. Banana
2. Mango
3. Orange

CSS Syntax

list-style-type: *value;*

- ➢ list-style-type: lower-alpha; (a, b, c, d, e, etc.)
- ➢ list-style-type: lower-latin;
- ➢ list-style-type: lower-roman; (i, ii, iii, iv, v, etc.)
- ➢ list-style-type: upper-alpha;
- ➢ list-style-type: upper-roman;
- ➢ list-style-type: decimal-leading-zero; (01, 02, 03, etc.)
- ➢ list-style-type: hebrew;
- ➢ list-style-type: lower-greek;
- ➢ list-style-type: none;

HTML – MARQUEES

- An HTML marquee is a scrolling piece of text displayed either *horizontally* across or *vertically* down your webpage depending on the settings. This is created by using HTML <*marquee*> tag.
- **Note:** The HTML <marquee> tag may not be supported by various browsers so it is not recommended to rely on this tag, instead you can use JavaScript and CSS to create such effects.

- A simple syntax to use HTML <marquee> tag is as follows:

```
<marquee attribute_name="attribute_value"....more attributes>
One or more lines or text message or image
</marquee>
```

Following is the list of important attributes which can be used with <marquee> tag.

Attribute	Description
width	This specifies the width of the marquee. This can be a value like 10 or 20% etc.
height	This specifies the height of the marquee. This can be a value like 10 etc.
direction	This specifies the direction in which marquee should scroll. This can be a value like up, down, left or right.
behavior	This specifies the type of scrolling of the marquee. This can have a value like scroll, slide and alternate.

scrollamount	**This specifies the speed of marquee text. This can have a value like 10 etc.**
loop	**This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.**
bgcolor	**This specifies background color in terms of color name or color hex value.**
hspace	**This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.**
vspace	**This specifies vertical space around the marquee. This can be a value like 10 or 20% etc.**

```
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee>This is basic example of marquee</marquee>
</body>
</html>
```

```html
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee width="50%">This example will take only 50%
width</marquee>
</body>
</html>
```

```html
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee direction="right">This text will scroll from left to right</marquee>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body >
<marquee direction="up">This text will scroll from bottom
to up</marquee>
</body>
</html>
```