CUI Abbottabad

Department of Computer Science

Web Technologies

Lecture-6-CLO[2]
CSS-Part3
Teacher: Bushra Mushtaq

COMSATS University Islamabad, Abbottabad Campus

Agenda

- Introduction to Positioning
- Positioning Elements: Absolute Positioning, z-index
- Positioning Elements: Relative Positioning, span
- Backgrounds
- Element dimensions
- Box Model:
  - Margin
  - Border
  - Padding
  - Contents
- Floating elements and Box-Model
- Media Types and Media Queries

POSITIONING

- Before CSS, controlling element positioning in HTML documents was difficult, the browser determined positioning.
- CSS introduces the **position** property to control the position of html elements in a web document.
- The position property specifies the type of positioning method used for an element :
  1. Absolute
  2. Relative
  3. Static
  4. Fixed
  5. Sticky
- **We will study first 4 types as they are supported by google-chrome,Edge and IE but the “Sticky” is supported by safari with –webkit-prefix support.**

POSITIONING

- After setting the positioning method the elements are positioned using:
  - top
  - bottom
  - left
  - right

  properties

1. ABSOLUTE POSITIONING AND Z-INDEX

- Normally, elements are positioned on the page in the order in which they appear in the HTML5 document.

- But specifying an element's position as absolute removes the element from the normal flow of elements on the page, instead positioning it according to the distance from the top, left, right or bottom margins of its **containing block-level element**.

- This means that it displayed on its own line and has a **virtual box** around it.

1. ABSOLUTE POSITIONING AND Z-INDEX

- Absolute positioning needs:
  a) length measurement in absolute/relative units for the top, bottom, left and right length.
  b) Z-index layer number

1. ABSOLUTE POSITIONING AND Z-INDEX

a) **<u>Absolute-length measurements</u>**

Absolute units do *not* vary in size based on the system. These are fixed in all systems.

- These units are
  - **in** (inches),
  - **cm** (centimeters),
  - **mm** (millimeters),
  - **pt** (points; 1 pt = 1/72 in)
  - **pc** (picas; 1 pc = 12 pt)

RELATIVE LENGTH MEASUREMENTS

Relative-length Measurement

- Relative lengths include
  - px: **stands for pixel**, it varies in size, based on screen resolution
  - em: it is the measurement of uppercase *M* height—the most frequently used font measurement),
  - ex: it is the measurement of lowercase *x*-height—usually set to a lowercase *x*'s height).
  - %: (e.g., font-size: 50%). To set an element to display text at 150 percent of its default text size, you could use :

font-size: 1.5em **or**

font-size: 150%

1. ABSOLUTE POSITIONING AND Z-INDEX

b) <u>Z-index</u>

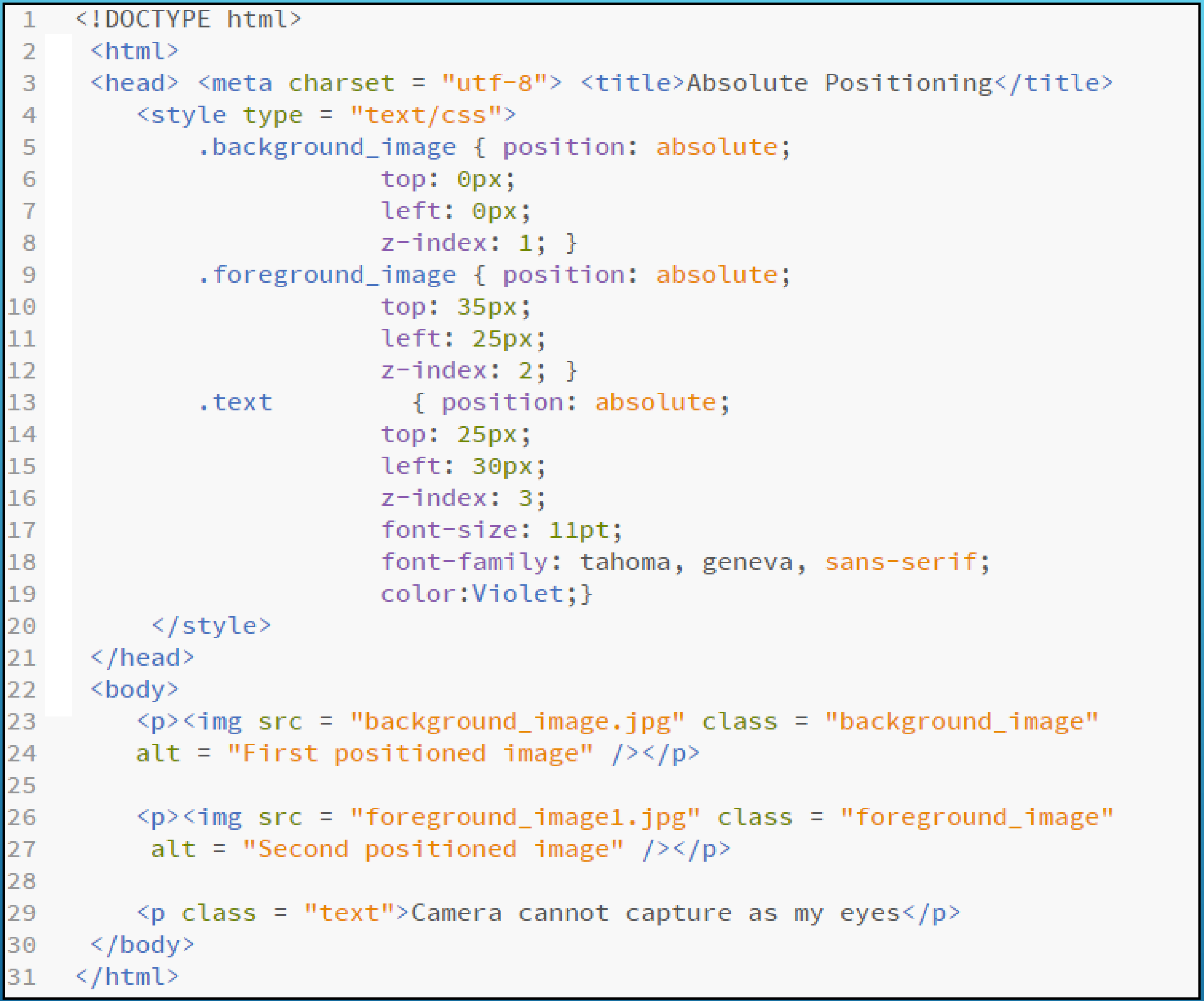
- This property allows you to **layer overlapping elements**
  - Elements that have higher z-index values are displayed in front of elements with lower z-index values.
- If elements have the same z-index value,
  - the elements are placed from background to foreground in the order in which they are encountered in the document.
- The **default z-index** value is 0.

1. ABSOLUTE POSITIONING AND Z-INDEX

- In Program—Listing6.1 we are going to display the following contents using CSS embedded stylesheet:

  1. background_image.jpg on top left corner of the browser at layer-1 →background image
  2. foreground_image.jpg on 35px below from top and 25px away from left corner of the browser at layer-2 →above layer 1
  3. User defined text of size 11pt with any font-family among(**tahoma, geneva, sans-serif**)on 25px below from top and 30 px away from left corner of the browser at 3rd layer of→above layer 2

1. ABSOLUTE POSITIONING AND Z-INDEX



```
<!DOCTYPE html>
 <html>
 <head> <meta charset = "utf-8"> <title>Absolute Positioning</title>
    <style type = "text/css">
        .background_image { position: absolute;
                    top: 0px;
                    left: 0px;
                    z-index: 1; }
        .foreground_image { position: absolute;
                    top: 35px;
                    left: 25px;
                    z-index: 2; }
        .text             { position: absolute;
                    top: 25px;
                    left: 30px;
                    z-index: 3;
                    font-size: 11pt;
                    font-family: tahoma, geneva, sans-serif;
                    color:Violet;}
     </style>
 </head>
 <body>
    <p><img src = "background_image.jpg" class = "background_image"
    alt = "First positioned image" /></p>

    <p><img src = "foreground_image1.jpg" class = "foreground_image"
     alt = "Second positioned image" /></p>

    <p class = "text">Camera cannot capture as my eyes</p>
 </body>
</html>
```

Listing-6.1-absolute_positioning.html

1. ABSOLUTE POSITIONING AND Z-INDEX

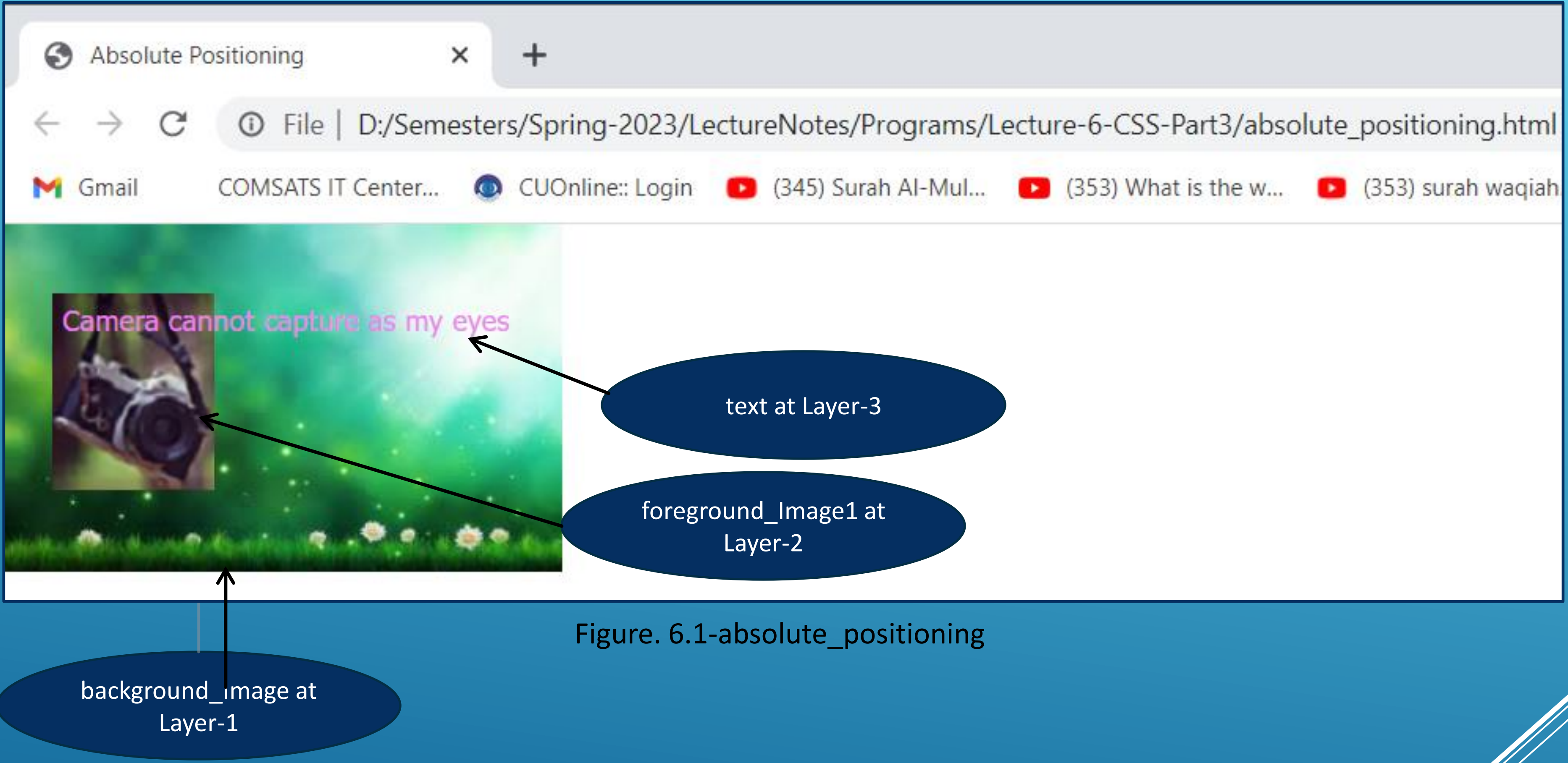

Absolute Positioning
File | D:/Semesters/Spring-2023/LectureNotes/Programs/Lecture-6-CSS-Part3/absolute_positioning.html
Gmail
COMSATS IT Center...
CUOnline:: Login
(345) Surah Al-Mul...
(353) What is the w...
(353) surah waqiah
Camera cannot capture as my eyes
text at Layer-3
foreground_Image1 at Layer-2
background_image at Layer-1

Figure. 6.1-absolute_positioning

BEFORE EXPLANATION OF LISTING 6.1

- **Block level-elements** are those elements which have a virtual box around them, so that we can set its top, left, right, bottom margins, padding etc.
- Some examples of block-level elements include
  - section,
  - div,
  - p
  - heading elements (h1 through h6).
- Note that block-level elements have starting and ending tag<></>
- In listing 6.1 we use <p></p>around <img> tag and text ,hence created a virtual box around them for setting margins for top and left

EXPLANATION LISTING-6.1

- The style section line#(4-20)
  - Line#(5-8) style for class ”background_image”
    - Specifying position as **absolute**
    - Margin from Top is 0px
    - margin from left is 0px
    - Z-index is 1
  - Line#(9-12) style for class ”foreground_image”
    - Specifying position as **absolute**
    - Margin below from Top is 35px
    - margin from left is 25px
    - Z-index is 2
  - Line#(13-19) style for class ”text”
    - Specifying position as **absolute**
    - Margin below from Top is 25px
    - margin from left is 30px
    - Z-index is 3
    - Font-size-11pt
    - Font-family any among tahoma, geneva or sans serif
    - Color is Violet

EXPLANATION LISTING-6.1

- The document body secion line#(22-30)
  - Line#(23-25):
    - the <img> tag has class name ”background-image” hence it adapted the style given at Line#(5-8)
  - Line#(26-27):
    - the <img> tag has class name ”foreground-image” hence it adapted the style given at Line#(9-12)
  - Line#(29):
    - the <p> tag has class name ”text” hence it adapted the style given at Line#(13-19)

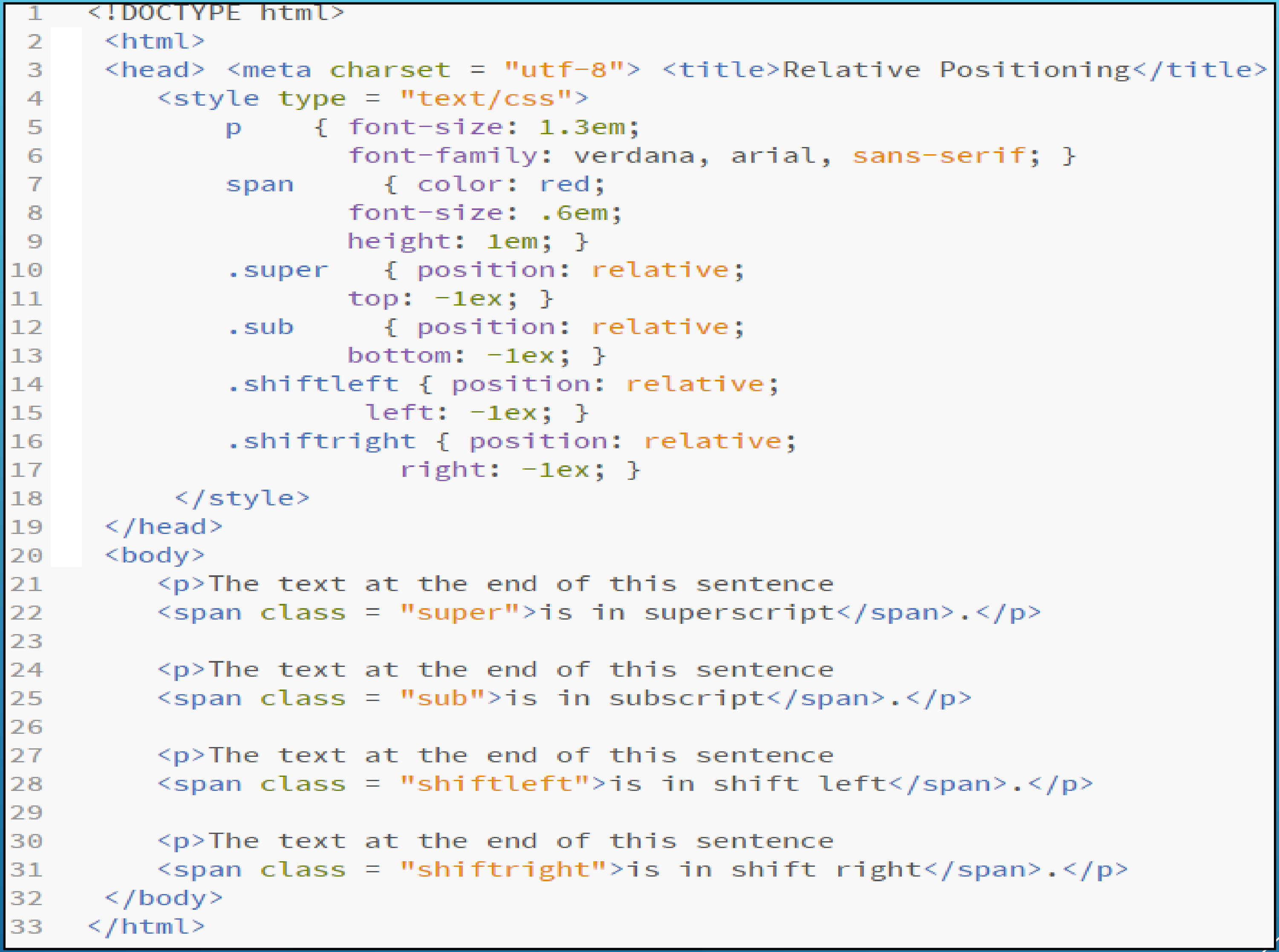
2. RELATIVE POSITIONING

- Setting the **position property to** relative
  - Unlike absolute positioning, relative positioning keeps elements in the general flow of elements on the page, so **positioning is relative to other elements in the flow.**
  - For relative positioning we usually use **relative units of measurement** to set left, right, top and bottom margins, padding, borders etc. of block level elements

2. RELATIVE POSITIONING AND NESTED SPAN

- In Program—Listing 6.2 we are going to display the following result using CSS embedded stylesheet:
  1. class “super” used in <span>to display a text is in superscript relative to given normal text in <p>
  2. class “sub” used in <span>to display a text is in subscript relative to given normal text in <p>
  3. class “shiftleft” used in <span>to displaying a text to little shift left relative to given normal text in <p>
  4. class “shiftright” used in <span>to displaying a text to little shift right relative to given normal text in <p>
- **Note:** usually no need of z-index layer number in relative positioning, in absence of z-index all element will be displayed on layer 0

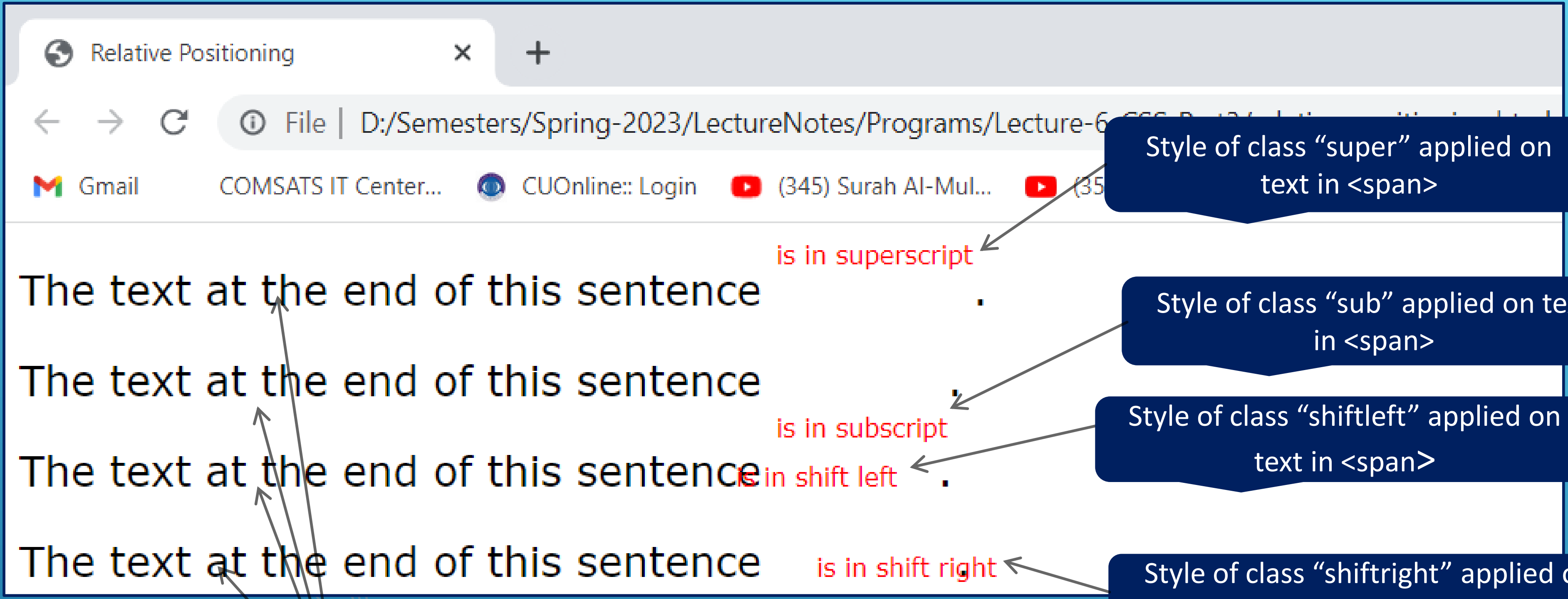
2. RELATIVE POSITIONING



```
<!DOCTYPE html>
 <html>
 <head> <meta charset = "utf-8"> <title>Relative Positioning</title>
    <style type = "text/css">
        p    { font-size: 1.3em;
               font-family: verdana, arial, sans-serif; }
        span     { color: red;
               font-size: .6em;
               height: 1em; }
        .super   { position: relative;
               top: -1ex; }
        .sub     { position: relative;
               bottom: -1ex; }
        .shiftleft { position: relative;
                left: -1ex; }
        .shiftright { position: relative;
                  right: -1ex; }
     </style>
 </head>
 <body>
    <p>The text at the end of this sentence
    <span class = "super">is in superscript</span>.</p>

    <p>The text at the end of this sentence
    <span class = "sub">is in subscript</span>.</p>

    <p>The text at the end of this sentence
    <span class = "shiftleft">is in shift left</span>.</p>

    <p>The text at the end of this sentence
    <span class = "shiftright">is in shift right</span>.</p>
 </body>
</html>
```

Listing- 6.2-relative_positioning.html

2. RELATIVE POSITIONING



```
The text at the end of this sentence is in superscript .
The text at the end of this sentence is in subscript .
The text at the end of this sentence is in shift left .
The text at the end of this sentence is in shift right .
```

Style of class “super” applied on text in <span>

Style of class “sub” applied on text in <span>

Style of class “shiftleft” applied on text in <span>

Style of class “shiftright” applied on text in <span>

Figure -6.2-relative_positioning

Style at line#(5-6) applied on element <p> given four times in body

Style at line#(7-9) applied on element <span> given four times in body

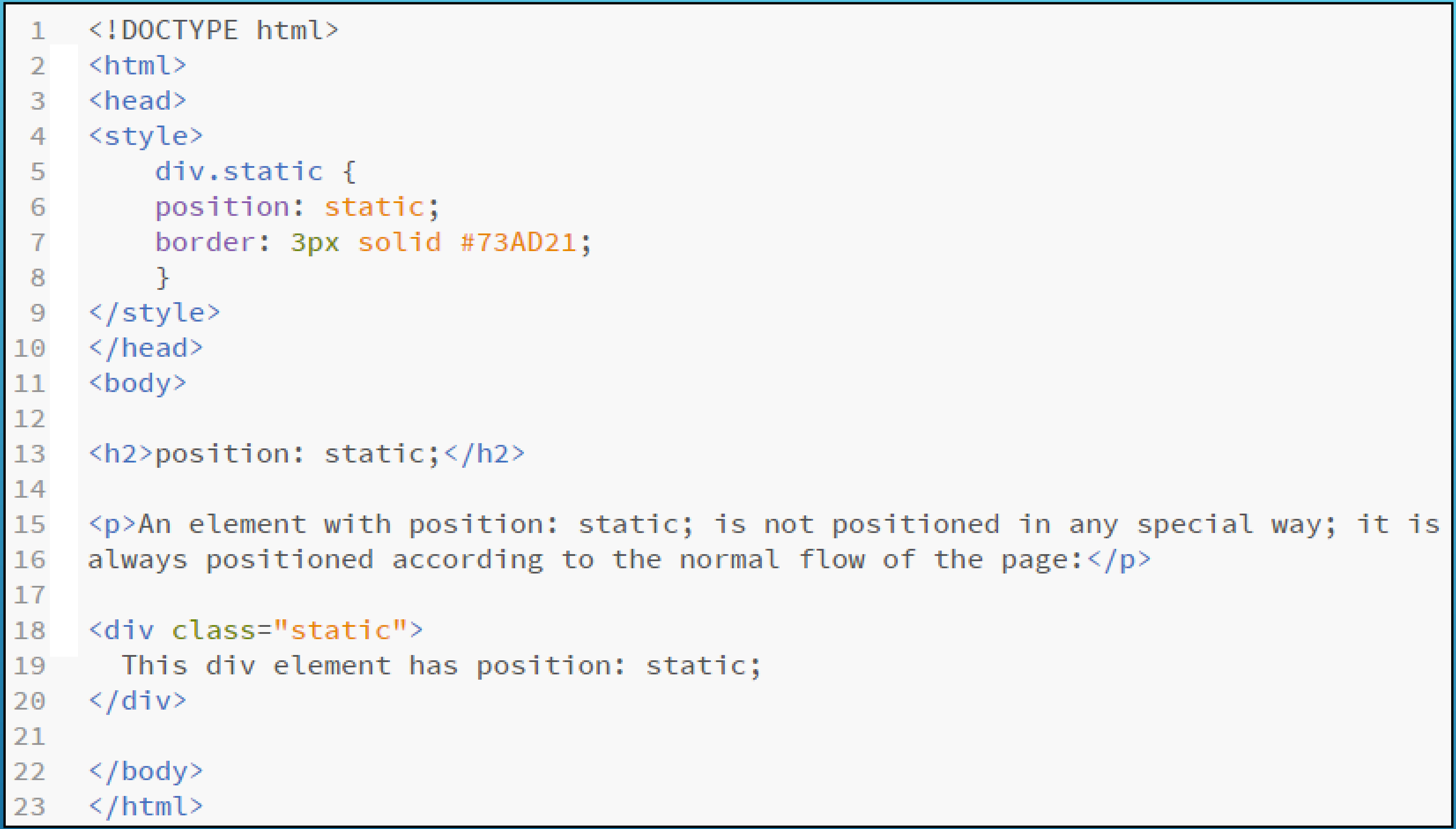
2. RELATIVE POSITIONING

- The style section line#(4-18)
  - Line#(5-6) style for element <p>
    - Self explanatory
  - Line#(7-9) style for element <span>
    - Self explanatory
  - Line#(10-11) style for class ”super”
    - Specifying position as **relative**
    - Margin Top is -1ex:
      - Move this text towards top(-ve sign) according to 1ex relative to element just before it
  - Line#(12-13) style for class ”sub”
    - Specifying position as **relative**
    - Margin bottom is -1ex:
      - Move this text towards bottom(-ve sign) according to 1ex relative to element just before it
  - Line#(14-15) style for class ”shiftleft”
    - Specifying position as **relative**
    - Margin left is -1ex:
      - Move this text towards left(-ve sign) according to 1ex relative to element just before it
  - Line#(16-17) style for class ”shiftright”
    - Specifying position as **relative**
    - Margin right is -1ex:
      - Move this text towards right(-ve sign) according to 1ex relative to element just before it

3. STATIC POSITIONING

- 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:
- Listing 6.3 describing the static position for html element <div>

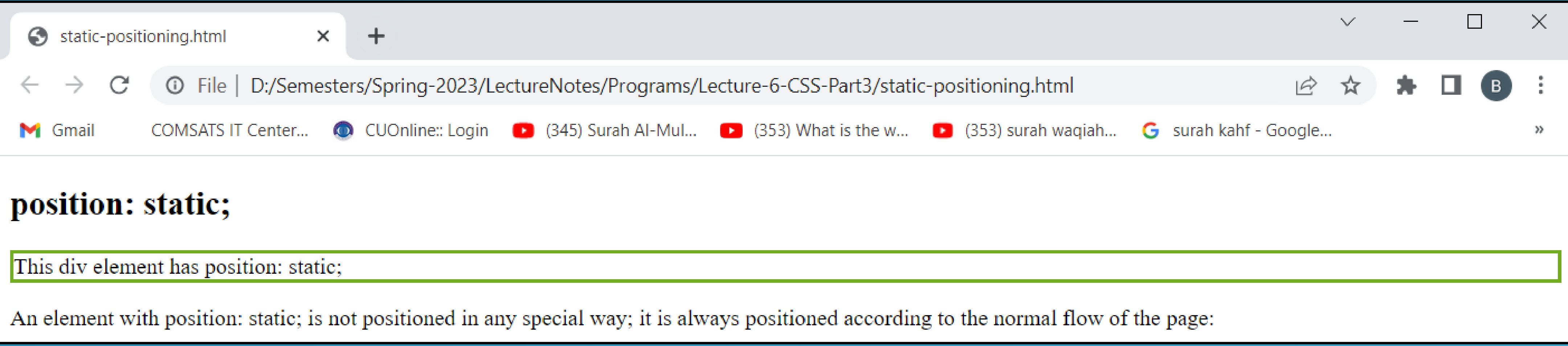
3. STATIC POSITIONING



```html
<!DOCTYPE html>
<html>
<head>
<style>
    div.static {
    position: static;
    border: 3px solid #73AD21;
    }
</style>
</head>
<body>

<h2>position: static;</h2>

<p>An element with position: static; is not positioned in any special way; it is
always positioned according to the normal flow of the page:</p>

<div class="static">
  This div element has position: static;
</div>

</body>
</html>
```

Listing-6.3-static-positioning.html

3. STATIC POSITIONING



position: static;

This div element has position: static;

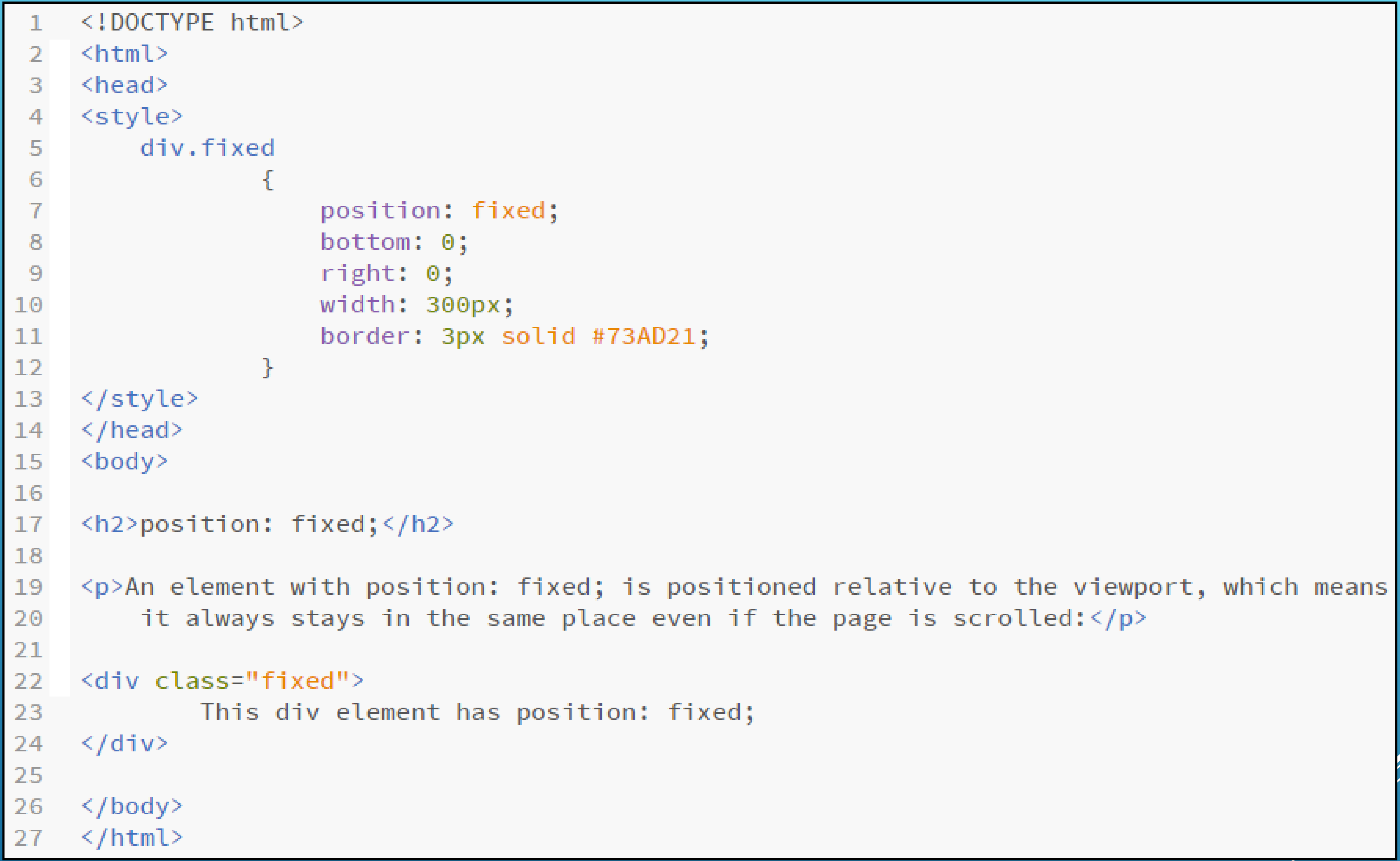
An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

Figure-6.3-static-positioning

4. FIXED POSITIONING

- 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.
- Notice the fixed element in the lower-right corner of the page. Here is the listing-6.4 in which CSS is used for fixed positioning of div element:

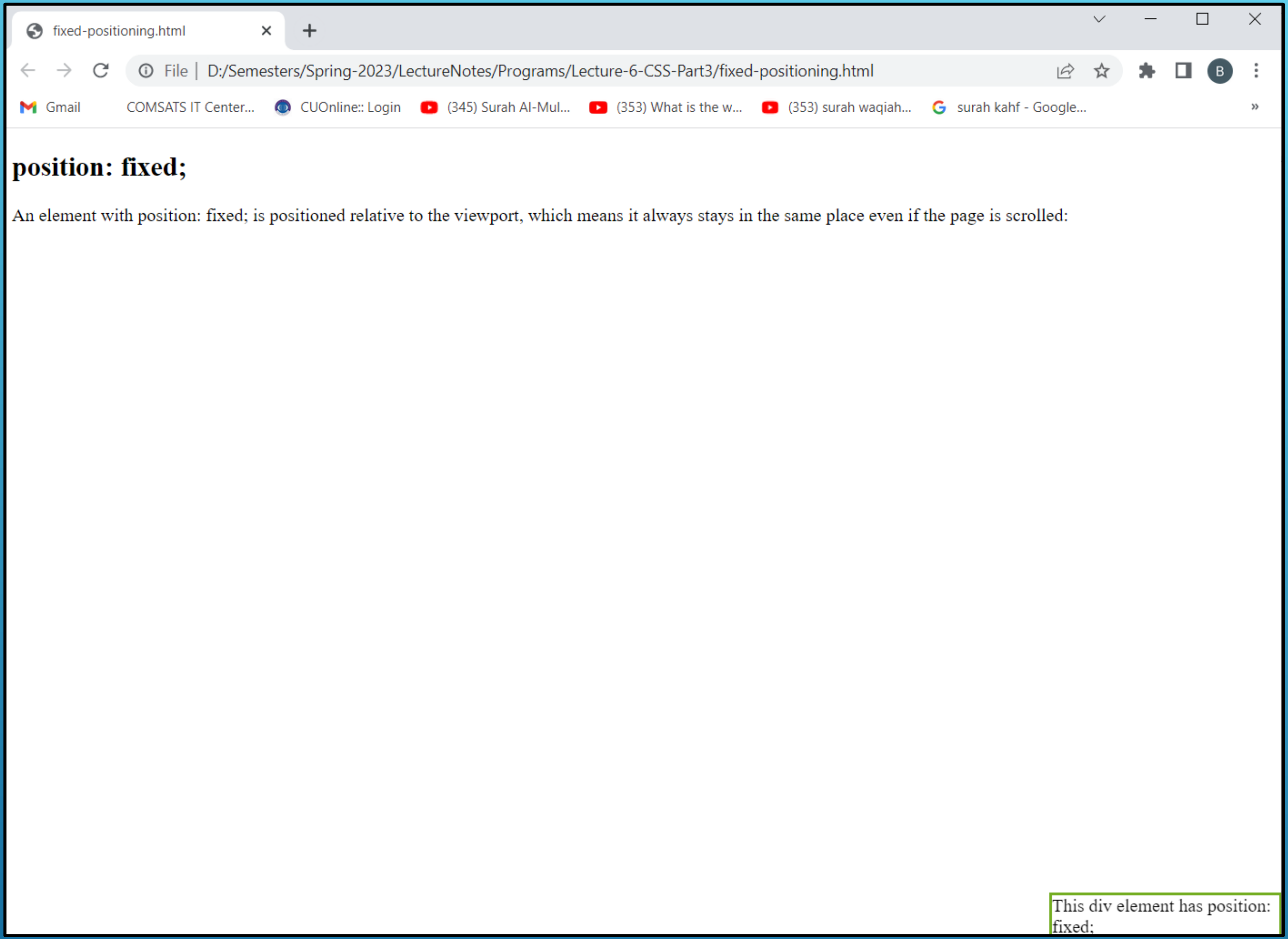
4. FIXED POSITIONING



```
<!DOCTYPE html>
<html>
<head>
<style>
    div.fixed
            {
                position: fixed;
                bottom: 0;
                right: 0;
                width: 300px;
                border: 3px solid #73AD21;
            }
</style>
</head>
<body>

<h2>position: fixed;</h2>

<p>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:</p>

<div class="fixed">
        This div element has position: fixed;
</div>

</body>
</html>
```

Figure-6.4-fixed-positioning.html

4. FIXED POSITIONING


fixed-positioning.html

File | D:/Semesters/Spring-2023/LectureNotes/Programs/Lecture-6-CSS-Part3/fixed-positioning.html

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:

This div element has position: fixed;

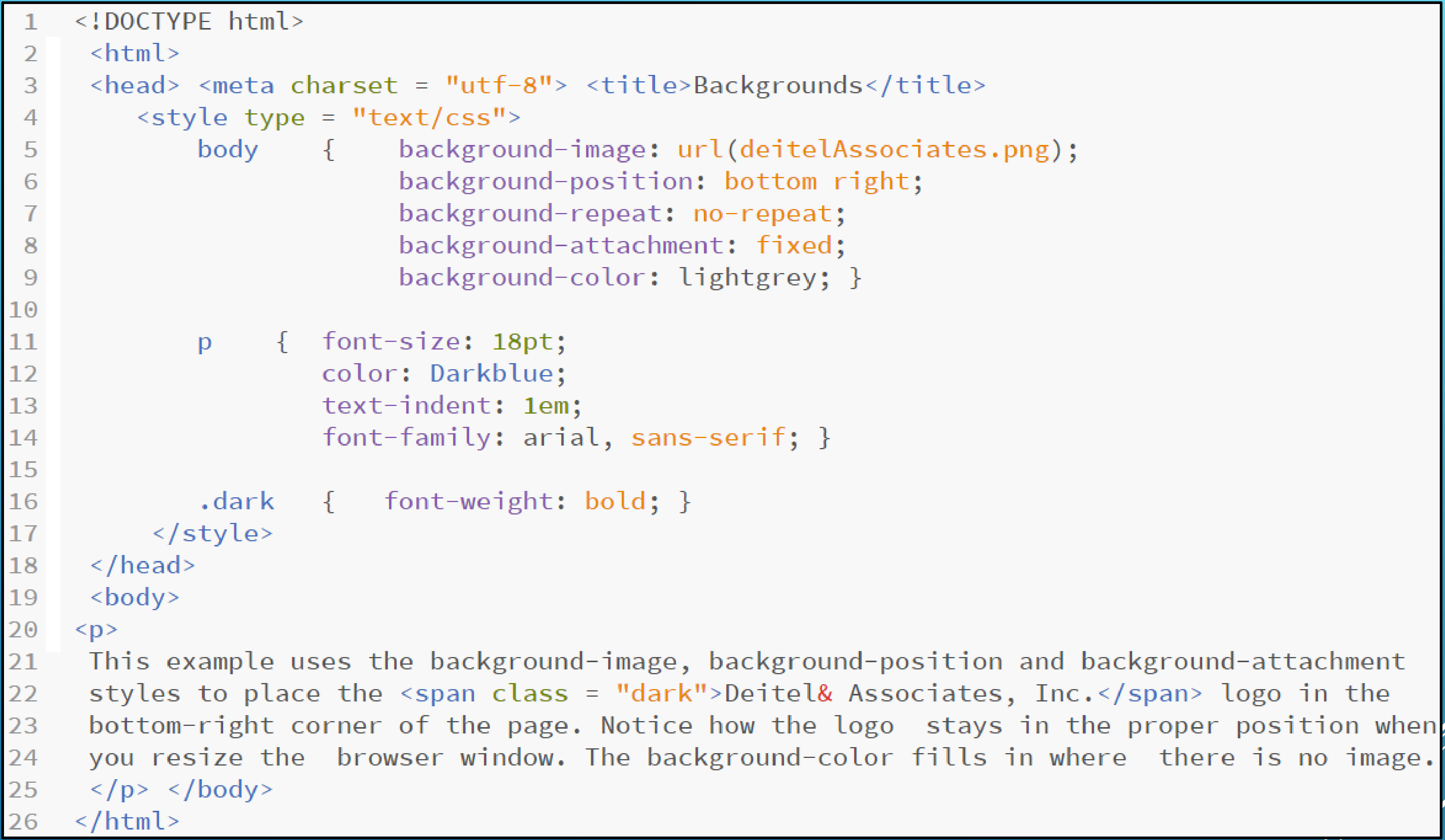
Figure-6.4-fixed-positioning

BACKGROUNDS

The CSS background properties are used to define the background effects for elements.

- you will learn about the following CSS background properties:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position
- In listing 6.5 all above properties are used, their explanation is given after listing and its output

BACKGROUNDS



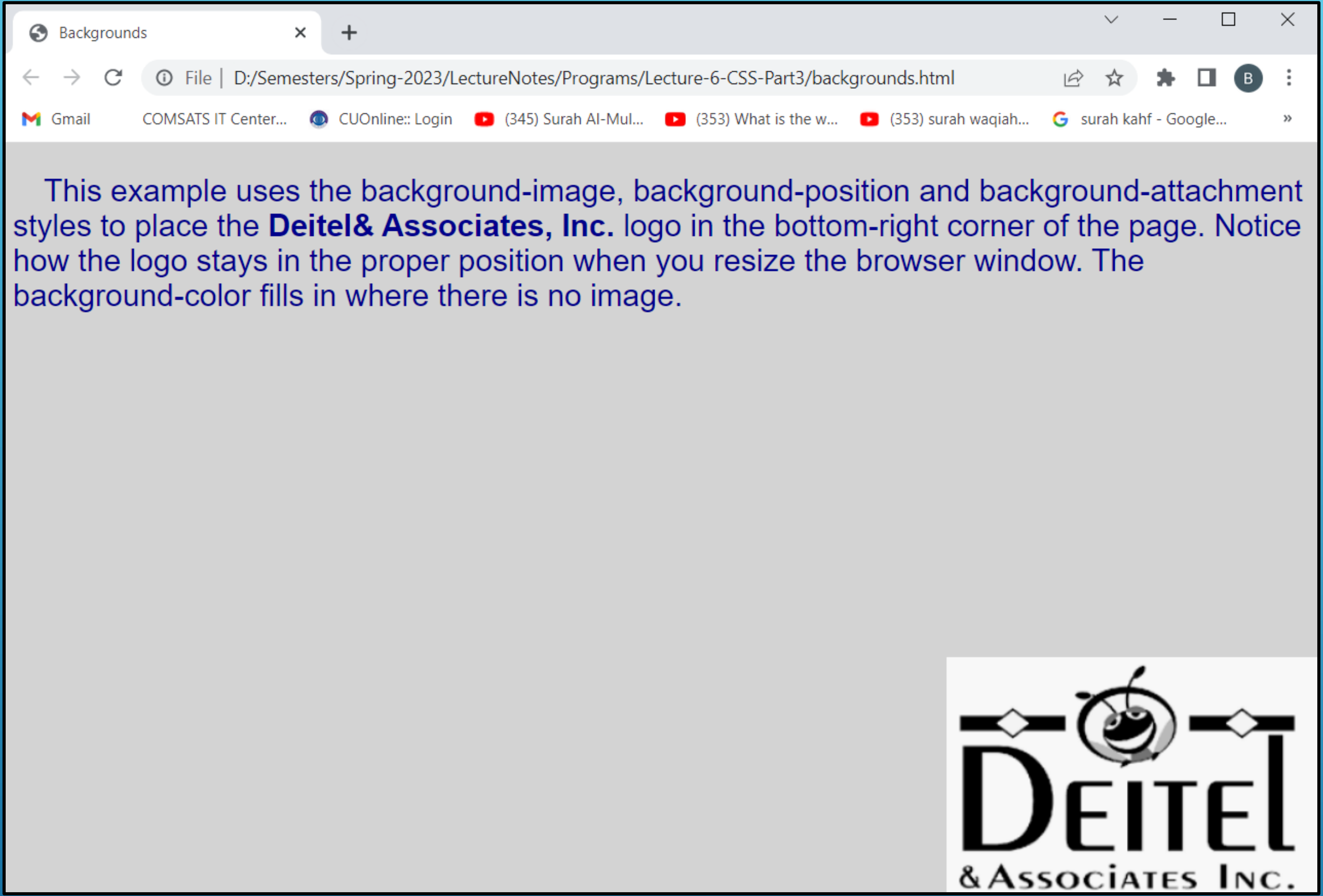
```html
<!DOCTYPE html>
 <html>
 <head> <meta charset = "utf-8"> <title>Backgrounds</title>
    <style type = "text/css">
        body    {    background-image: url(deitelAssociates.png);
                     background-position: bottom right;
                     background-repeat: no-repeat;
                     background-attachment: fixed;
                     background-color: lightgrey; }

        p    {   font-size: 18pt;
                 color: Darkblue;
                 text-indent: 1em;
                 font-family: arial, sans-serif; }

        .dark   {   font-weight: bold; }
     </style>
 </head>
 <body>
<p>
 This example uses the background-image, background-position and background-attachment
 styles to place the <span class = "dark">Deitel& Associates, Inc.</span> logo in the
 bottom-right corner of the page. Notice how the logo  stays in the proper position when
 you resize the  browser window. The background-color fills in where  there is no image.
 </p> </body>
</html>
```

Listing- 6.5-backgrounds.html

BACKGROUNDS

Output-------Listing- 6.5-backgrounds.html

Figure- 6.5-backgrounds



This example uses the background-image, background-position and background-attachment styles to place the **Deitel& Associates, Inc.** logo in the bottom-right corner of the page. Notice how the logo stays in the proper position when you resize the browser window. The background-color fills in where there is no image.

No matter how larger or small the document is ,text and other contents of document the image of Dietel & Associates will always be remain fix in bottom right corner

BACKGROUNDS

- The background-image property (line#5) specifies the image URL for the image ‘ *logo.png* ’ in the format **url**(*fileLocation*).
- Set the background-color property (line#9) in case the image is not found (and to fill in areas the image does not cover).
- The background-position property (line#6) places the image on the page.
- The keywords top, bottom, center, left and right are used individually or in combination for vertical and horizontal positioning.

BACKGROUNDS

- Position an image using lengths by specifying the horizontal length followed by the vertical length.
- **Example:**
  - position the image as horizontally centered (positioned at 50% of the distance across the screen) and 30 pixels from the top.

background-position: 50% 30px;

BACKGROUNDS

- The background-repeat property (line# 7) controls background image **tiling**, which places multiple copies of the image next to each other to fill the background.
- Set the tiling to no-repeat to display only one copy of the background image.
- Other values include:
  - repeat (the default) to tile the image vertically and horizontally,
  - repeat-x to tile the image only horizontally
  - repeat-y to tile the image only vertically.

BACKGROUNDS

- **background-attachment: fixed** (line#8), fixes the image in the position specified by **background-position.**
  - Scrolling the browser window will ***not*** move the image from its position.
- The default value of **background-attachment** is **scroll:**
  - moves the image as the user scrolls through the document.

BACKGROUNDS

- **text-indent** property (in line# 13) indent the first line of text in the element by a specified amount, in this case **1em.**
- Use this property to create a web page that reads more like a novel, in which the first line of every paragraph is indented.

- **font-style** (line#14)property formats text, which allows to set text to none, italic or oblique.
  - oblique is simply more slanted than italic—the browser will display in italic (default) if the system or font does not support oblique text.

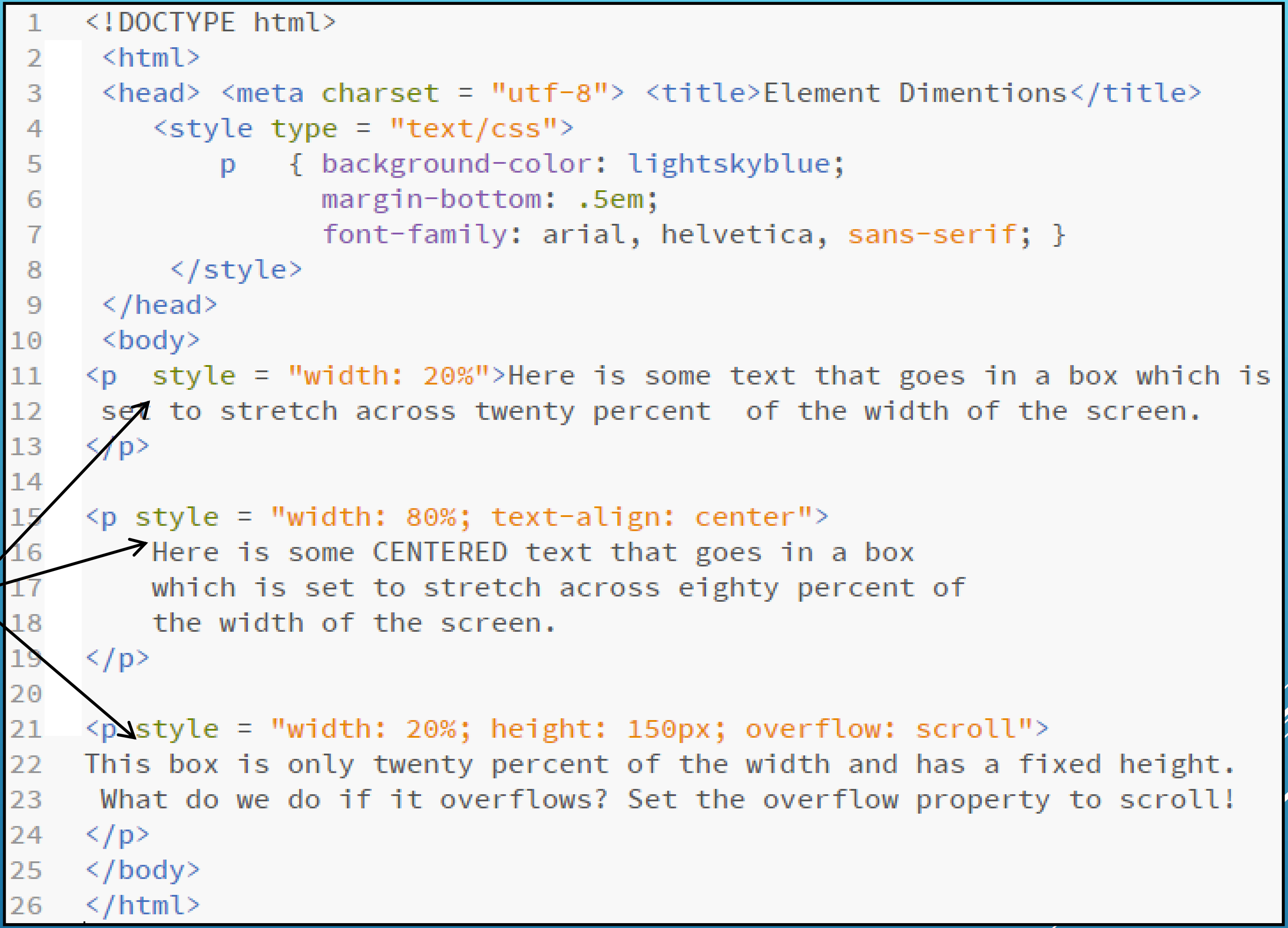
ELEMENT DIMENSIONS

- Through CSS we can specify dimensions of html elements(specifically block level element) on web page.
- Dimensions include width, height, alignment, scrolling etc.
- Block level elements are those elements which have starting and ending tags, so have a virtual box around them. Some examples of block-level elements are: section, div, p, headings(h1 to h6) etc.
- In listing 6.6 we are setting dimensions of three paragraphs with different width, heights, text-alignments and scrolling

ELEMENT DIMENSIONS

Embedded stylesheet line#4-line#8

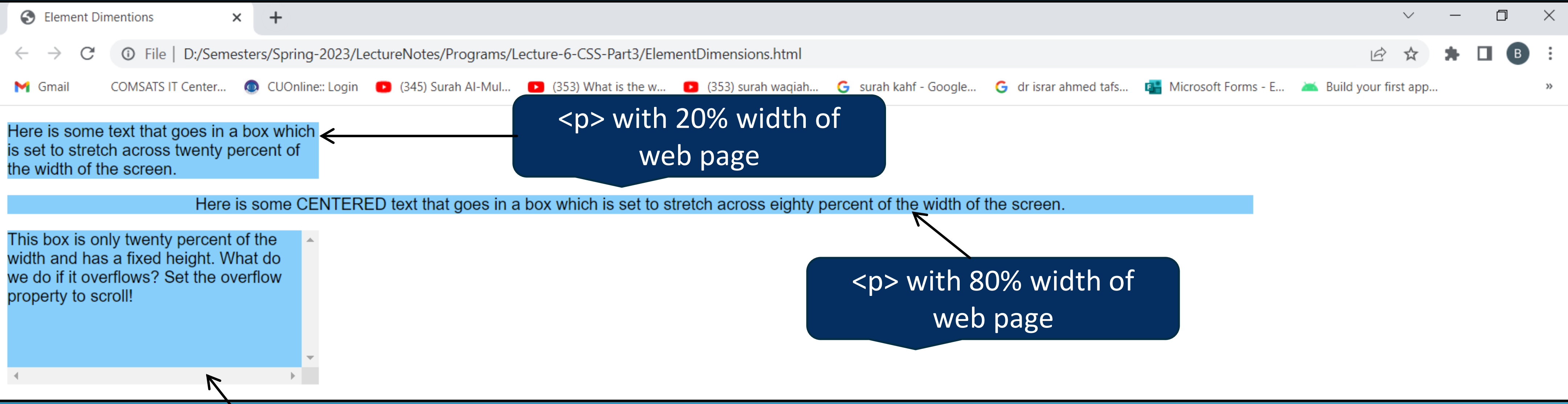
Inline style line#11 line#15 Line#21



```
1   <!DOCTYPE html>
2    <html>
3    <head> <meta charset = "utf-8"> <title>Element Dimentions</title>
4       <style type = "text/css">
5           p   { background-color: lightskyblue;
6                 margin-bottom: .5em;
7                 font-family: arial, helvetica, sans-serif; }
8        </style>
9    </head>
10   <body>
11  <p  style = "width: 20%">Here is some text that goes in a box which is
12   set to stretch across twenty percent  of the width of the screen.
13  </p>
14
15  <p style = "width: 80%; text-align: center">
16     Here is some CENTERED text that goes in a box
17     which is set to stretch across eighty percent of
18     the width of the screen.
19  </p>
20
21  <p style = "width: 20%; height: 150px; overflow: scroll">
22  This box is only twenty percent of the width and has a fixed height.
23   What do we do if it overflows? Set the overflow property to scroll!
24  </p>
25  </body>
26  </html>
```

Listing- 6.6-ElementDimensions.html

ELEMENT DIMENSIONS



Here is some text that goes in a box which is set to stretch across twenty percent of the width of the screen.

Here is some CENTERED text that goes in a box which is set to stretch across eighty percent of the width of the screen.

This box is only twenty percent of the width and has a fixed height. What do we do if it overflows? Set the overflow property to scroll!

<p> with 20% width of web page

<p> with 80% width of web page

<p> with 20% width of web page, fixed height 150 px,
Through scroll bar can see remaining text

Figure- - 6.6-ElementDimensions.html

EXPLANATION LISTING-6.6

Embedded Stylesheet—(line#4-line#8)

- As we know for embedded style is preferred when a common style needed for one or more elements in whole documents.
- Hence a common style set for all <p> elements: lightblue background color, .5em margin-bottom and same font-style(ariel, helvetica, sans serif)

Inline Style—(line#11)

- <p style="width:20%" > This will display this paragraph in 20% width of whole web page

Inline Style—(line#15)

- <p style="width:80% ;text-align=center">
- This will display this parapgraph in 80% width of whole web page with text be in center-aligned of the page

Inline Style—(line#15)

- <p style="width:20% ; height: 150px; overflow: scroll">
- This will display this parapgraph in 20% width of whole web page within fixed height of 150 px. we set the overflow property to scroll, a setting that adds scroll bars if the text overflows the boundaries

EXPLANATION LISTING-6.6

Try yourself the following on listing 6.6

- The width and height values also can be specified as relative or absolute lengths.

For example: width:10em

  - sets the element’s width to 10 times the default font size. This works only for block-level elements.

- other values for the text-align property includes: left and right. Try these

BOX MODEL AND TEXT FLOW

- All HTML5 elements specially block-level have a ***virtual box*** drawn around them, based on what is known as the **box model**.
- In CSS, the term "box model" is used when talking about design and layout.
- Box-model consists of: **margins, borders, padding, and the actual content.**
- Box-Model is described in figure 6.7

BOX MODEL AND TEXT FLOW

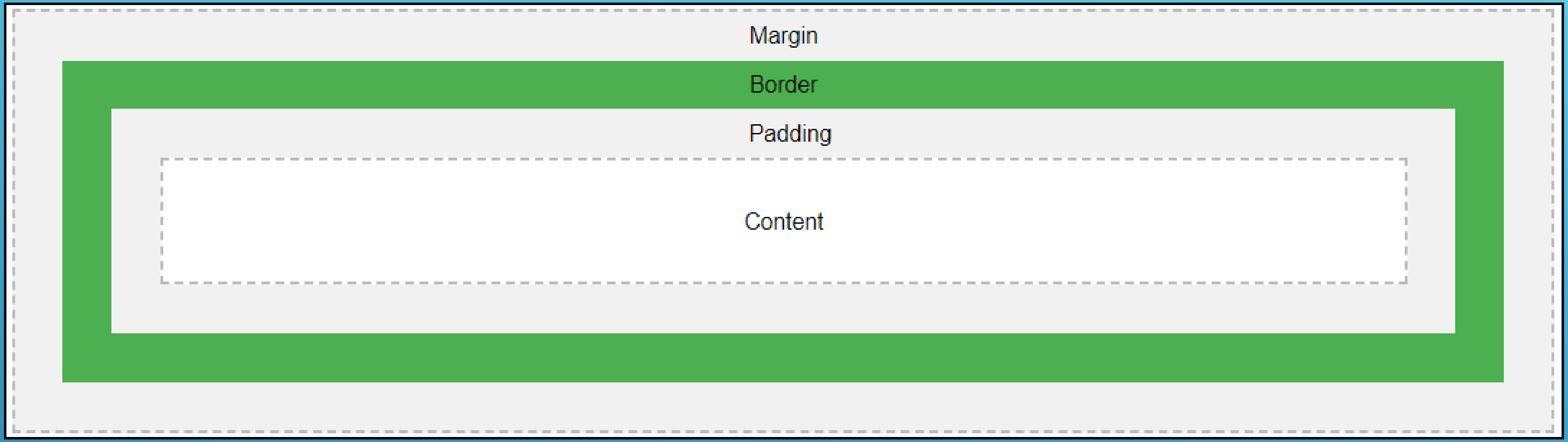


Figure- 6.7

BOX MODEL

Explanation of the different parts in figure-6.7:

- **Content** - The contents of the box, where page text/images/ or any other contents appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent
- The box model allows us to add a border around elements, and to define space between elements.

BOX MODEL-MARGIN

- CSS has properties for specifying the margin for each side of an element:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left
- All the margin properties can have the following values:
  - **auto** - the browser calculates the margin
  - **length** - specifies a margin in number value in absolute or relative units (px, pt, cm, etc…)
  - % - specifies a margin in % of the width of the containing element
  - inherit - specifies that the margin should be inherited from the parent element
- **Tip:** Negative values are allowed.

BOX MODEL-MARGIN

Margin - Shorthand Property

- To shorten the code, it is possible to specify all the margin properties in one property.
- For this **margin** property is a shorthand property for the following individual margin properties:
    - margin-top
    - margin-right
    - margin-bottom
    - margin-left

Example-1

margin: 25px 50px 75px 100px;

It means:

- top margin is 25px
- right margin is 50px
- bottom margin is 75px
- left margin is 100px

BOX MODEL-MARGIN

Margin - Shorthand Property

Example-2

margin: 25px 50px 75px ;

It means:

- top margin is 25px
- right and left margins are 50px
- bottom margin is 75px

Example-3

margin: 25px 50px ;

It means:

- top and bottom margins are 25px
- right and left margins are 50px

Example-4

margin: 25px

It means:

- All four margins are 25px

BOX MODEL-MARGIN

The **auto** Value:

- You can set the margin property to auto to horizontally center the element within its container.
- The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

Example----Use margin: auto:

```
div {
  width: 300px;
  margin: auto;
  border: 1px solid red;
}
```

BOX MODEL-MARGIN

The **inherit** Value:

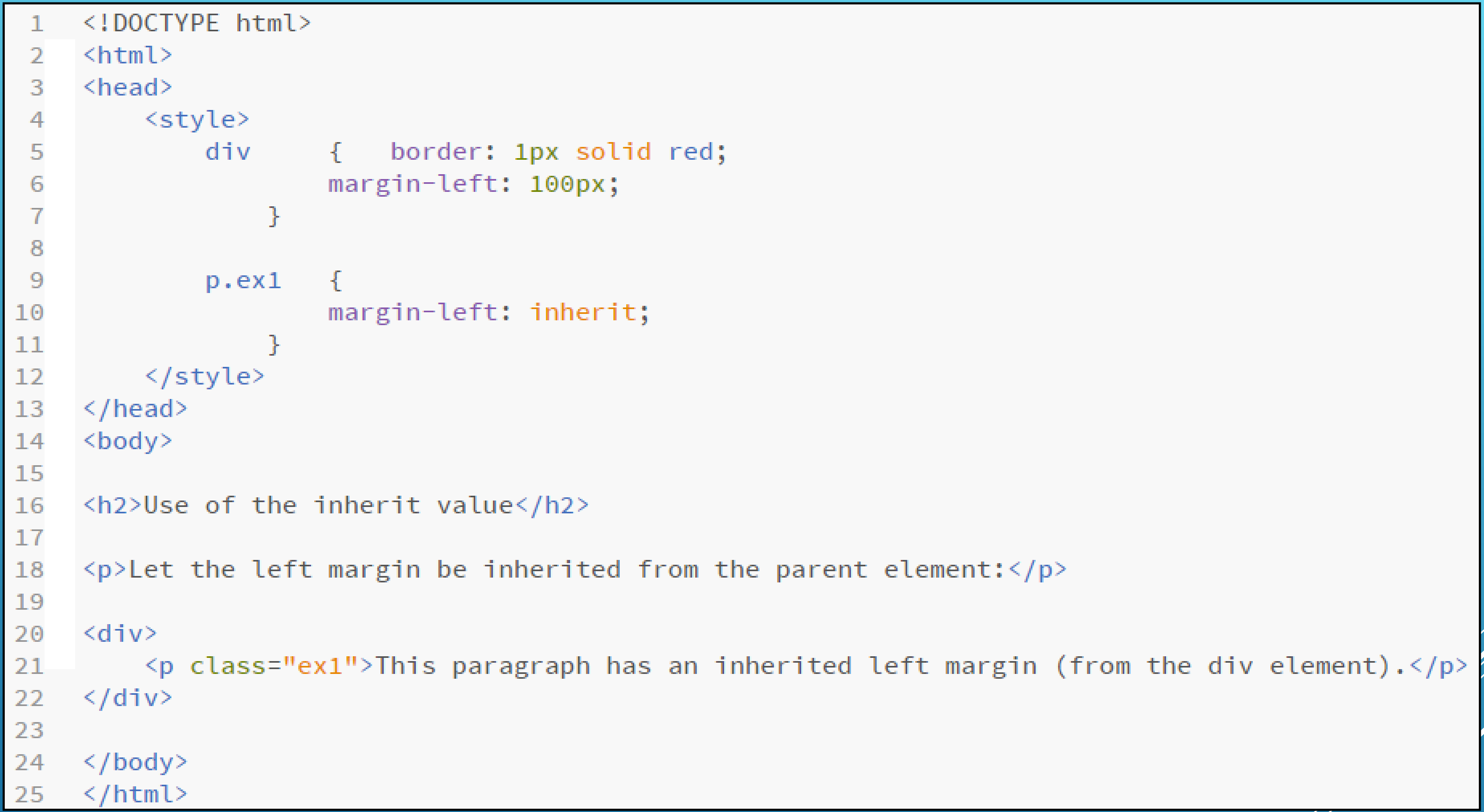
- This example lets the left margin of the <p class="ex1"> element be inherited from the parent element (<div>):

Example----Use margin: inherit

```
div {
  border: 1px solid red;
  margin-left: 100px;
}

p.ex1 {
  margin-left: inherit;
}
```

Listing-6.8 is complete program for above example

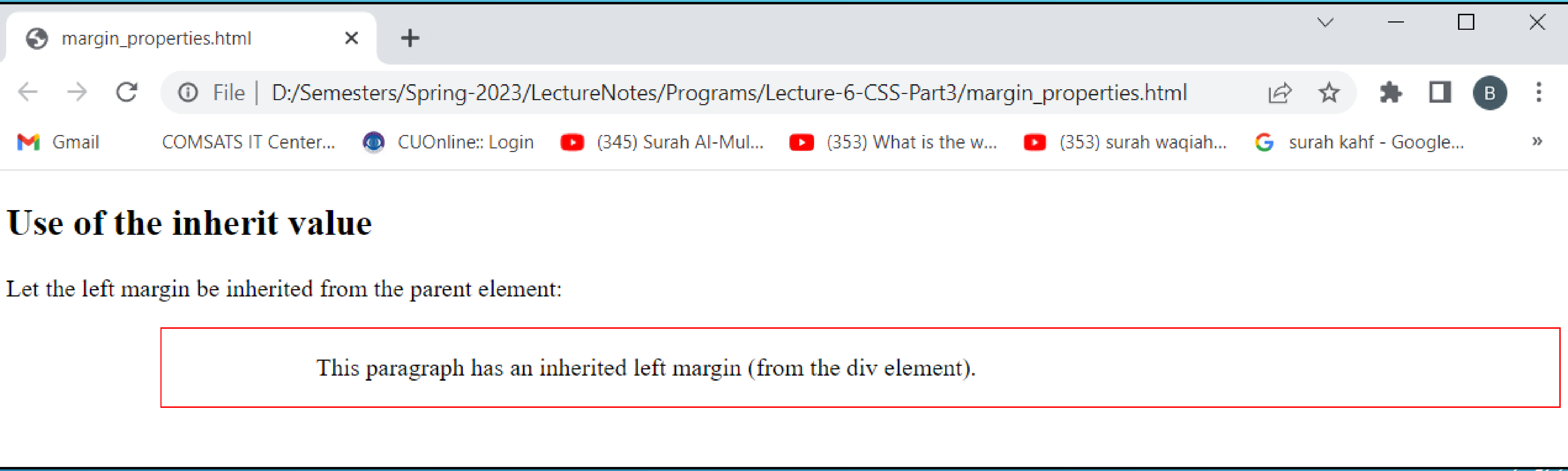
BOX MODEL-MARGIN



```
<!DOCTYPE html>
<html>
<head>
    <style>
        div     {   border: 1px solid red;
                margin-left: 100px;
            }

        p.ex1   {
                margin-left: inherit;
            }
    </style>
</head>
<body>

<h2>Use of the inherit value</h2>

<p>Let the left margin be inherited from the parent element:</p>

<div>
    <p class="ex1">This paragraph has an inherited left margin (from the div element).</p>
</div>

</body>
</html>
```

Listing- 6.8- margin_properties.html

BOX MODEL-MARGIN



Use of the inherit value

Let the left margin be inherited from the parent element:

This paragraph has an inherited left margin (from the div element).

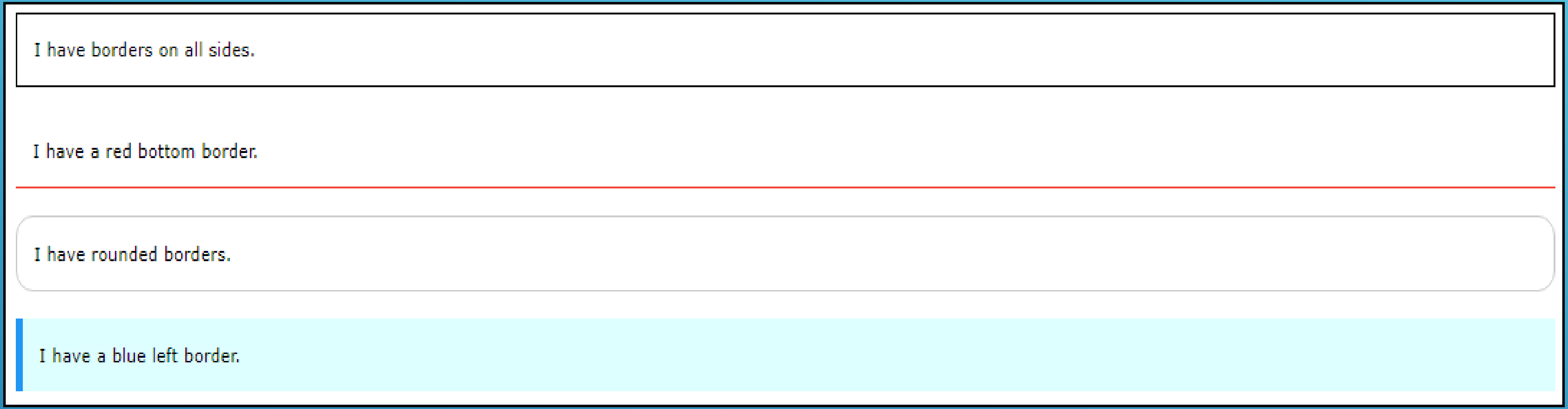
Figure- 6.8- margin_properties.html

BOX MODEL-PADDING

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- Padding - Individual Sides
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left
- All the padding properties can have the following values:
  - **length** - specifies a padding in px, pt, cm, etc.
  - *%* - specifies a padding in % of the width of the containing element
  - inherit - specifies that the padding should be inherited from the parent element
- **Tip:** Negative values are allowed.

BOX MODEL-BORDERS

- CSS border properties allow you to specify the style, width, and color of an element's border, some examples are:


I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

Figure- 6.9-Borders-Padding

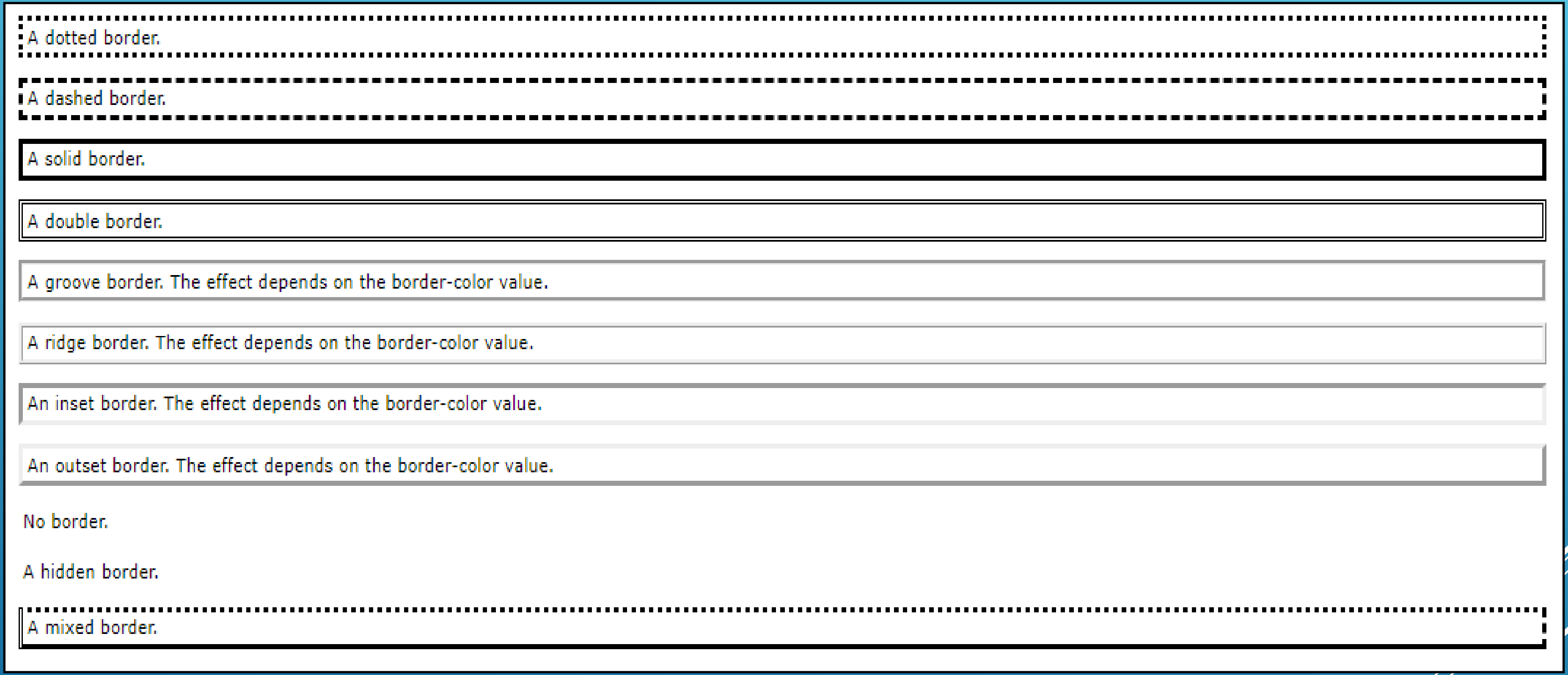
BOX MODEL-BORDERS

CSS border styles:

- 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

BOX MODEL-BORDERS

CSS border styles:


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.

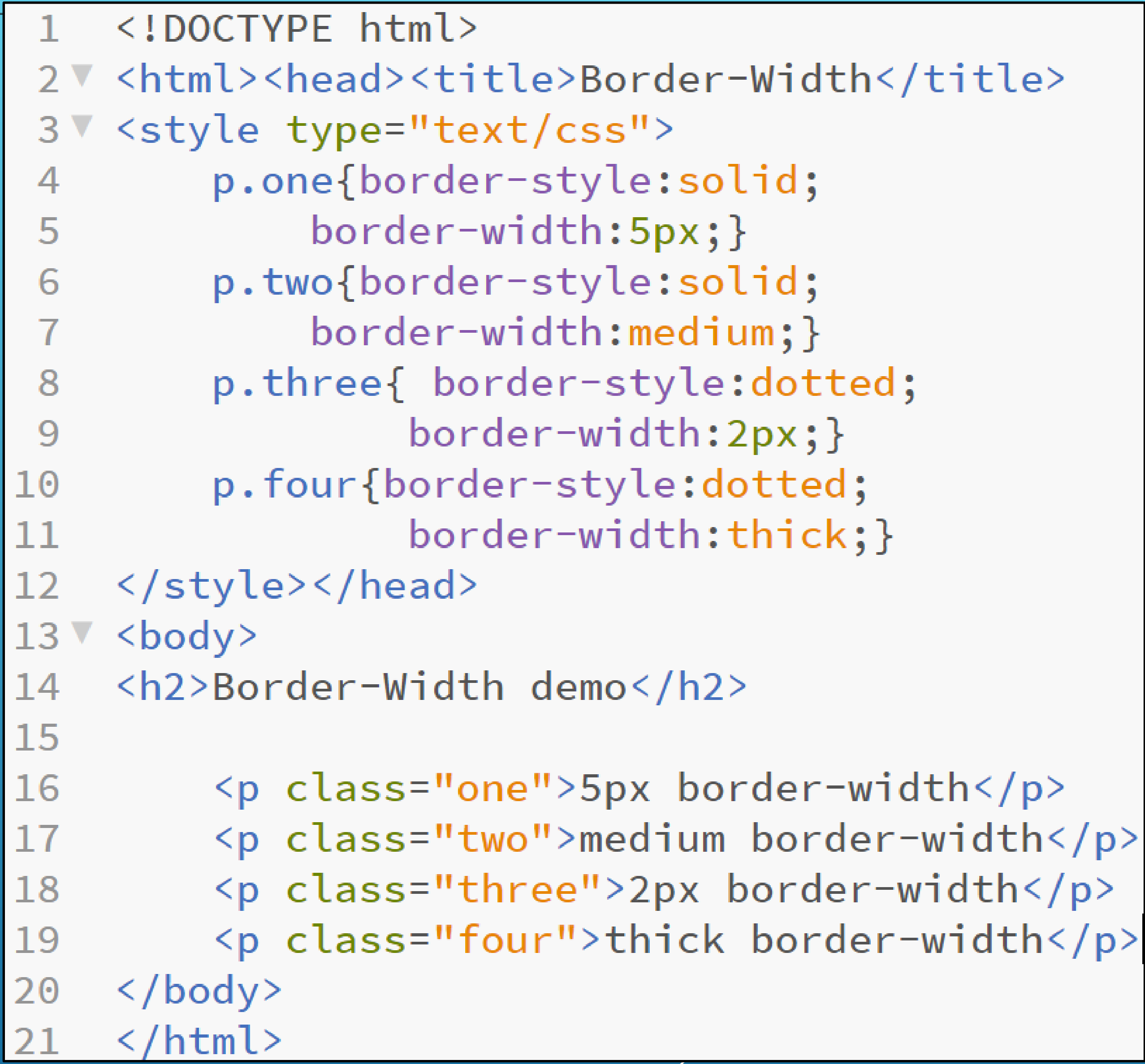
Figure- 6.10- Border Styles

BOX MODEL-BORDERS

Border-width

- The border-width property specifies the width of the four borders.
- The width can be set as one of following two ways:
  - a specific size (in px, pt, cm, em, etc)
  - by using one of the three pre-defined values: thin, medium, or thick:
- Example: Demonstration of the different border widths:

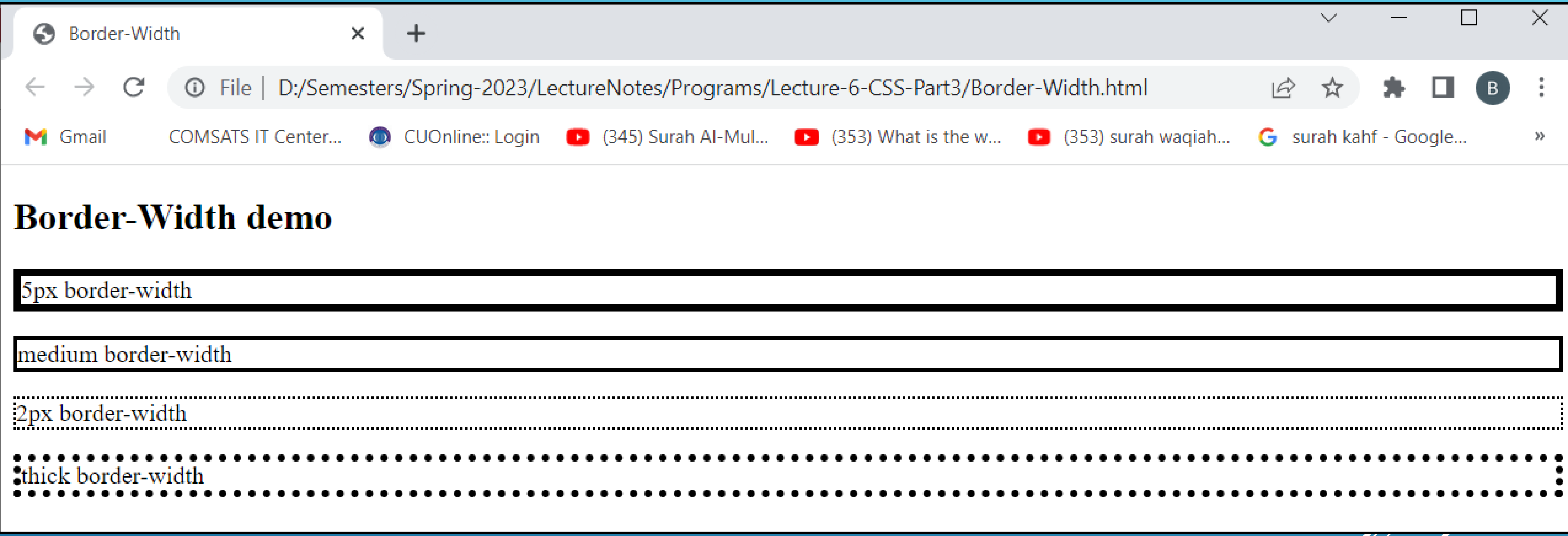
6.11-Border-Width



```
<!DOCTYPE html>
<html><head><title>Border-Width</title>
<style type="text/css">
        p.one{border-style:solid;
                border-width:5px;}
        p.two{border-style:solid;
                border-width:medium;}
        p.three{ border-style:dotted;
                        border-width:2px;}
        p.four{border-style:dotted;
                        border-width:thick;}
</style></head>
<body>
<h2>Border-Width demo</h2>

        <p class="one">5px border-width</p>
        <p class="two">medium border-width</p>
        <p class="three">2px border-width</p>
        <p class="four">thick border-width</p>
</body>
</html>
```

BOX MODEL-BORDERS

Border-width



Border-Width demo

5px border-width

medium border-width

2px border-width

thick border-width

Figure- Output-6.11-Border-Width

BOX MODEL-BORDERS

Border-color

- The border-color property is used to set the color of the four borders.
- The color can be set by:
  - name - specify a color name, like "red"
  - HEX - specify a HEX value, like "#ff0000"
  - RGB - specify a RGB value, like "rgb(255,0,0)"
  - HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
  - transparent
- **Note:** If border-color is not set, it inherits the color of the element.

BOX MODEL-BORDERS

Border-color

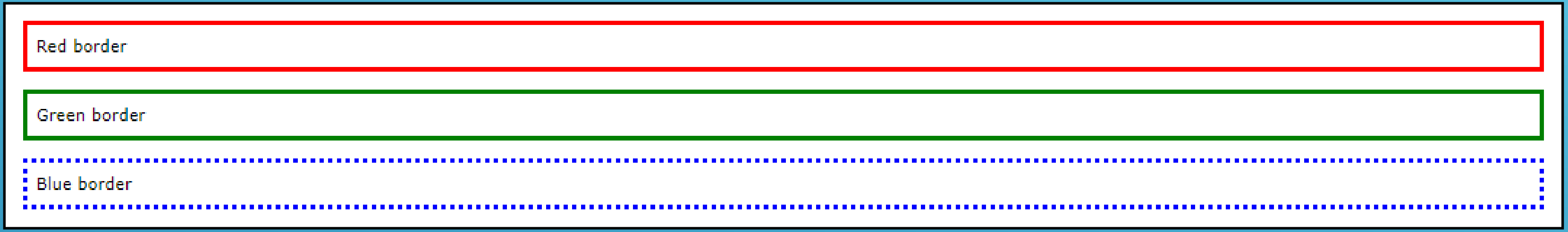
- Example

```
p.one {
  border-style: solid;
  border-color: red;
}

p.two {
  border-style: solid;
  border-color: green;
}

p.three {
  border-style: dotted;
  border-color: blue;
}
```

BOX MODEL-BORDERS

Border-color

- Output-Example


Red border

Green border

Blue border

Figure- 6.12- Border-Color.html

Border-Specific Side Colors

- The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).
- Example

```
p.one {
  border-style: solid;
  border-color: red green blue yellow;

/* red top, green right, blue bottom and yellow left */
}
```

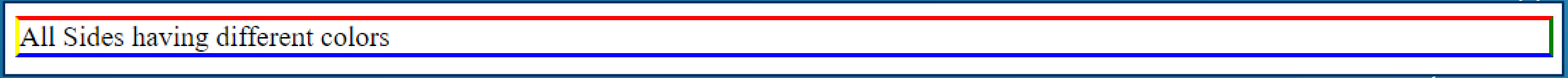

All Sides having different colors

Figure- 6.13- Border-SideSpecific-Color.html

BOX MODEL-BORDERS

CSS Border - Individual Sides

- From the examples on the previous pages, you have seen that it is possible to specify a different border for each side.
- In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left):
- Example:

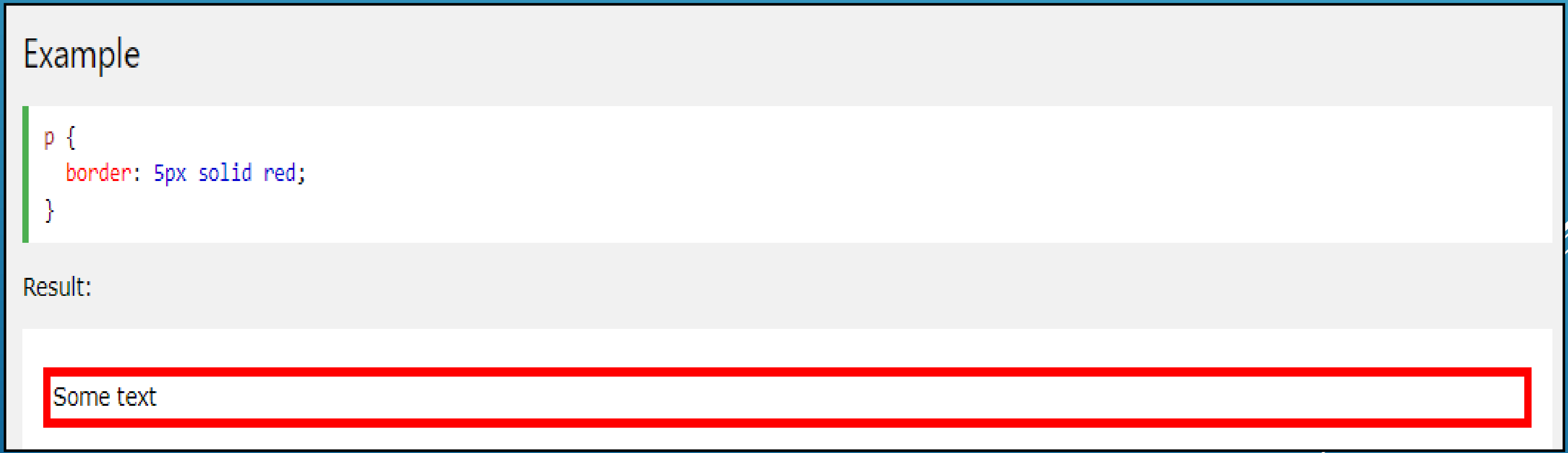
```
p {
  border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

BOX MODEL-BORDERS

CSS Border-Shorthand Property

- To shorten the code, it is also possible to specify all the individual border properties in one property.
- The border property is a shorthand property for the following individual border properties:
  - border-width
  - border-style (required)
  - border-color

Figure- 6.14-border-shorthand-property



Example

```
p {
  border: 5px solid red;
}
```

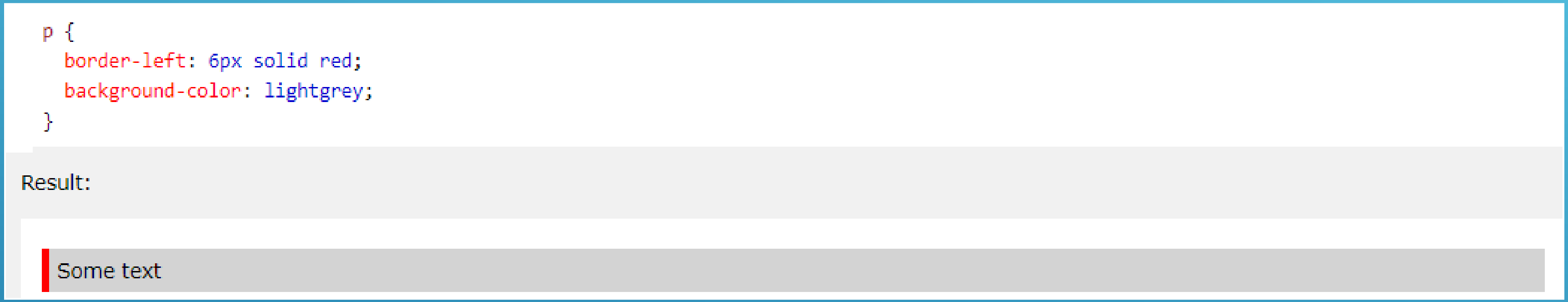
Result:

```
Some text
```

BOX MODEL-BORDERS

CSS *Border-Shorthand* Property

- You can also specify all individual borders
  - **Left Border**



```
p {
  border-left: 6px solid red;
  background-color: lightgrey;
}
```

Result:

```
Some text
```

Figure- 6.15-border-left-shorthand-property

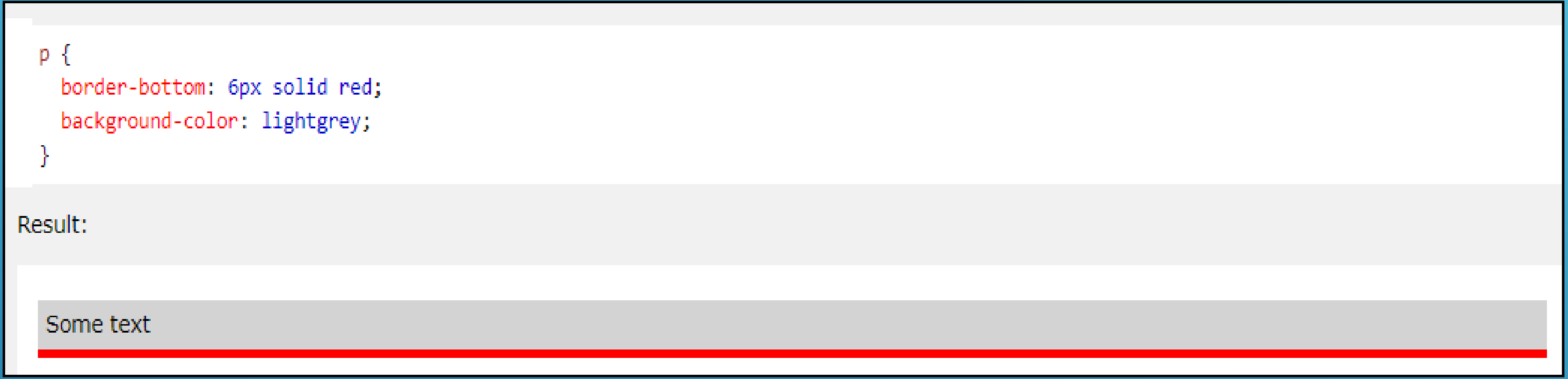
  - **Right Border**

Try yourself

BOX MODEL-BORDERS

CSS Border-Shorthand Property

- You can also specify all individual borders
  - **Bottom Border**



```
p {
  border-bottom: 6px solid red;
  background-color: lightgrey;
}
```

Result:

```
Some text
```

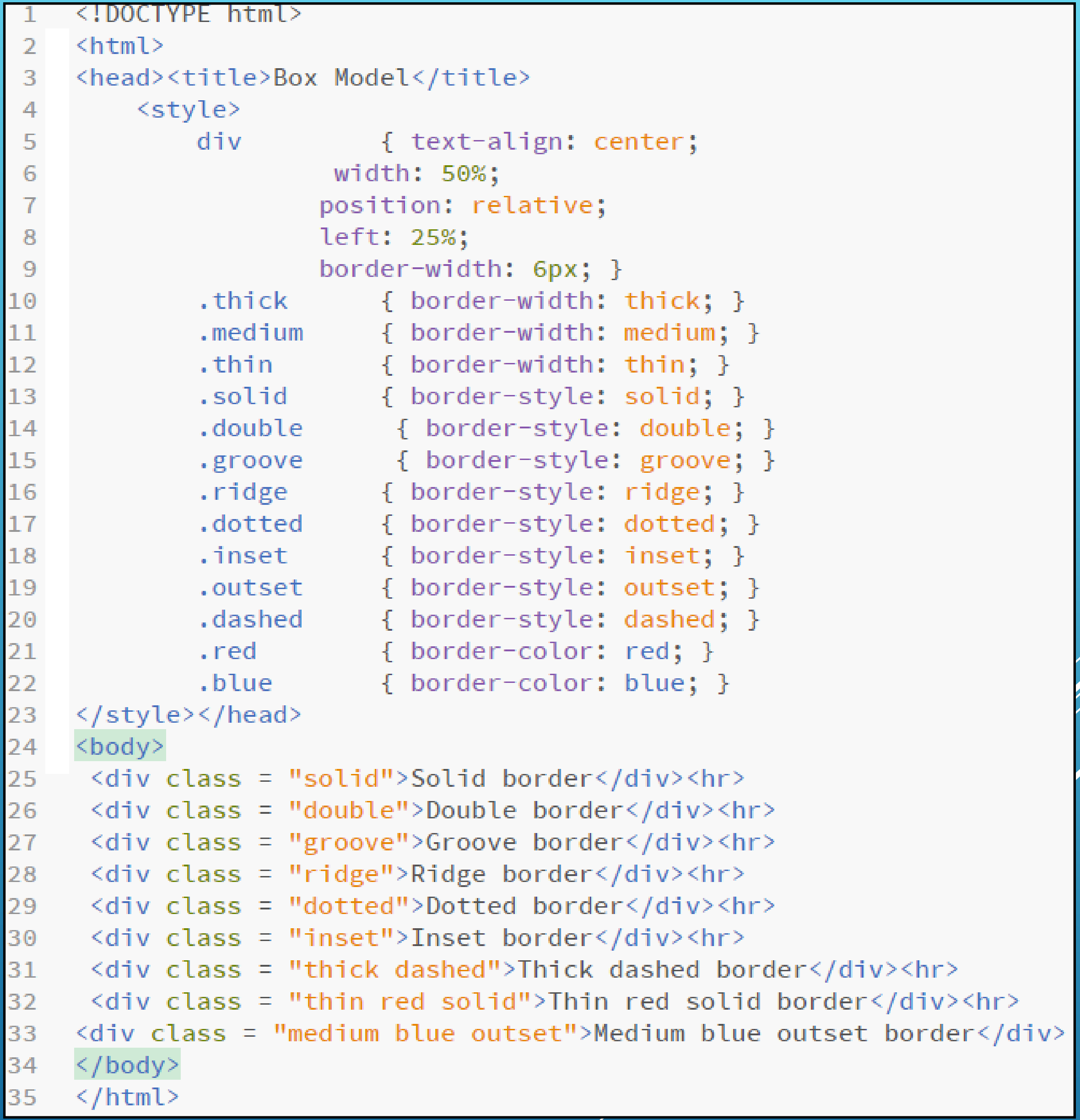
Figure- 6.16-Border-Bottom-Shorthand-Property.html

  - **Top Border**

Try yourself

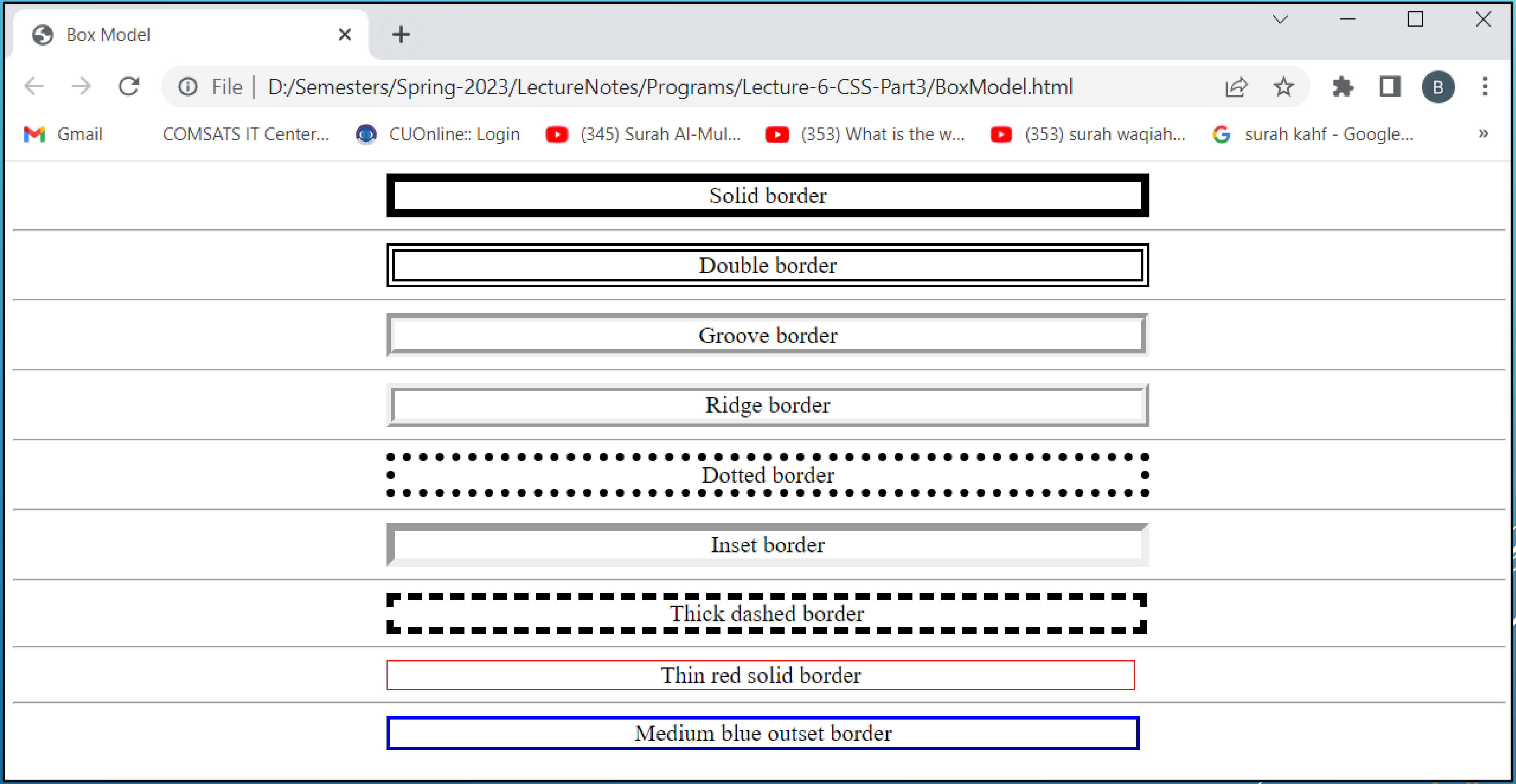
BOX MODEL

Listing—6.17-BoxModel.html



```
<!DOCTYPE html>
<html>
<head><title>Box Model</title>
    <style>
        div          { text-align: center;
                          width: 50%;
                         position: relative;
                         left: 25%;
                         border-width: 6px; }
        .thick       { border-width: thick; }
        .medium      { border-width: medium; }
        .thin        { border-width: thin; }
        .solid       { border-style: solid; }
        .double       { border-style: double; }
        .groove       { border-style: groove; }
        .ridge       { border-style: ridge; }
        .dotted      { border-style: dotted; }
        .inset       { border-style: inset; }
        .outset      { border-style: outset; }
        .dashed      { border-style: dashed; }
        .red         { border-color: red; }
        .blue        { border-color: blue; }
</style></head>
<body>
 <div class = "solid">Solid border</div><hr>
 <div class = "double">Double border</div><hr>
 <div class = "groove">Groove border</div><hr>
 <div class = "ridge">Ridge border</div><hr>
 <div class = "dotted">Dotted border</div><hr>
 <div class = "inset">Inset border</div><hr>
 <div class = "thick dashed">Thick dashed border</div><hr>
 <div class = "thin red solid">Thin red solid border</div><hr>
<div class = "medium blue outset">Medium blue outset border</div>
</body>
</html>
```

BOX MODEL-BORDERS



Solid border

Double border

Groove border

Ridge border

Dotted border

Inset border

Thick dashed border

Thin red solid border

Medium blue outset border

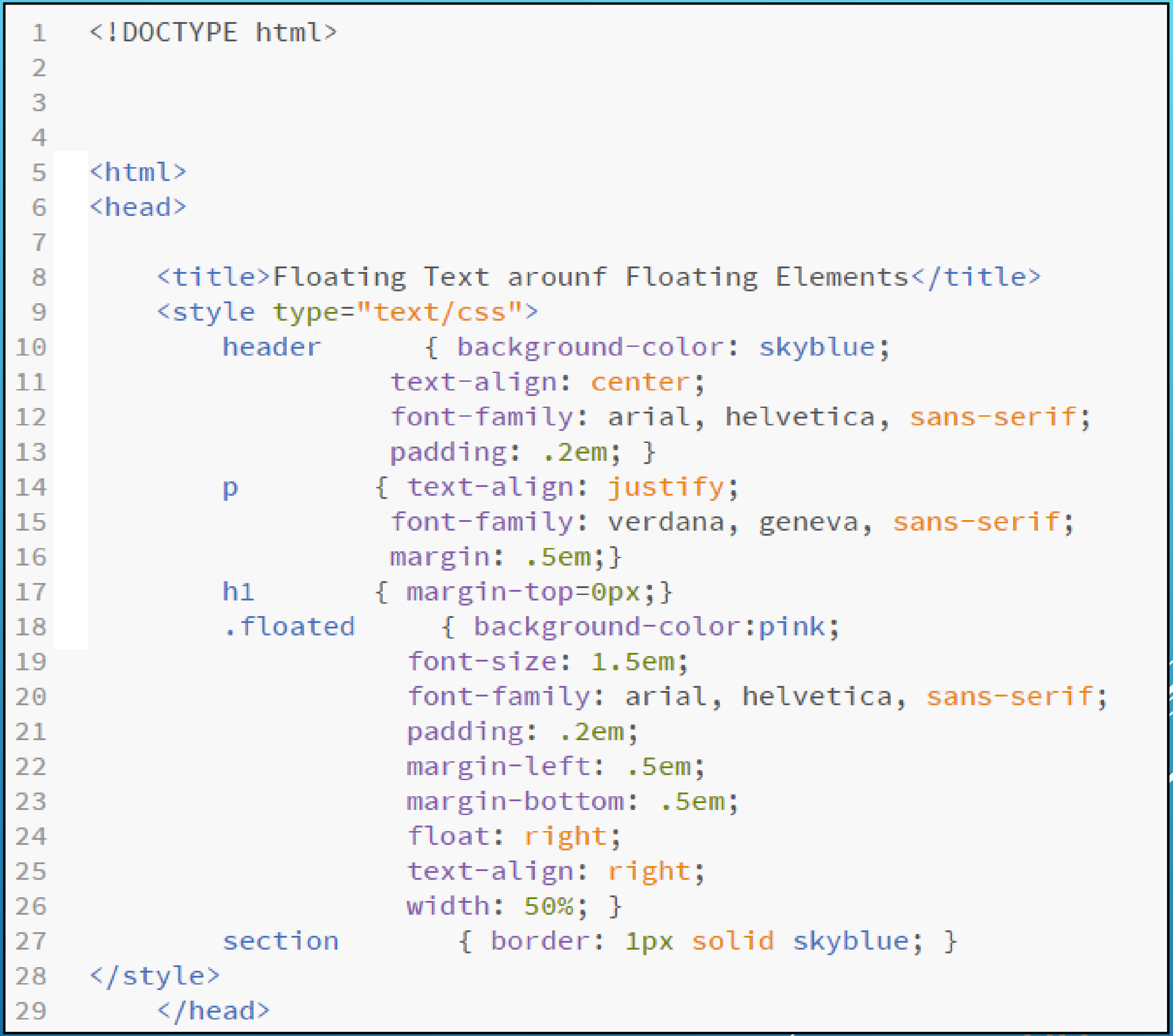
Figure—6.17-BoxModel

FLOATING ELEMENTS

- We’ve seen with absolute positioning that it’s possible to remove elements from the normal flow of text.
- CSS **float** property allows you to move an element to one side of the screen;
- other content in the document then *flows around* the floated element.
- Listing 6.18 demonstrates how **floating elements** and the **box model** can be used to control the layout of an entire page.

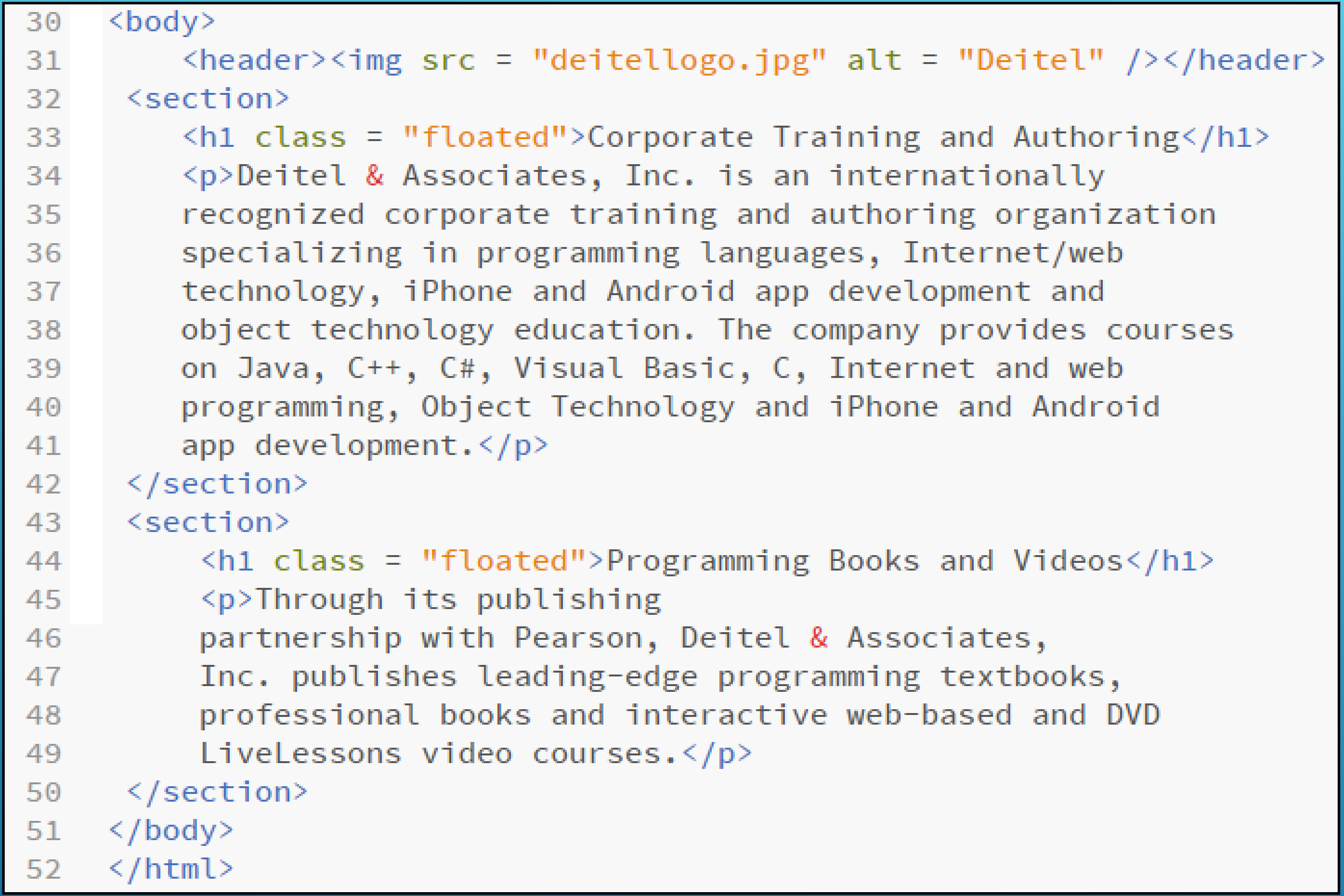
FLOATING ELEMENTS

Listing-6.18
-FloatingElement_and_BoxModel.html



```
<!DOCTYPE html>



<html>
<head>

    <title>Floating Text arounf Floating Elements</title>
    <style type="text/css">
        header      { background-color: skyblue;
                  text-align: center;
                  font-family: arial, helvetica, sans-serif;
                  padding: .2em; }
        p        { text-align: justify;
                  font-family: verdana, geneva, sans-serif;
                  margin: .5em;}
        h1       { margin-top=0px;}
        .floated     { background-color:pink;
                   font-size: 1.5em;
                   font-family: arial, helvetica, sans-serif;
                   padding: .2em;
                   margin-left: .5em;
                   margin-bottom: .5em;
                   float: right;
                   text-align: right;
                   width: 50%; }
        section       { border: 1px solid skyblue; }
</style>
    </head>
```

FLOATING ELEMENTS



```
<body>
    <header><img src = "deitellogo.jpg" alt = "Deitel" /></header>
 <section>
    <h1 class = "floated">Corporate Training and Authoring</h1>
    <p>Deitel &amp; Associates, Inc. is an internationally
    recognized corporate training and authoring organization
    specializing in programming languages, Internet/web
    technology, iPhone and Android app development and
    object technology education. The company provides courses
    on Java, C++, C#, Visual Basic, C, Internet and web
    programming, Object Technology and iPhone and Android
    app development.</p>
 </section>
 <section>
     <h1 class = "floated">Programming Books and Videos</h1>
     <p>Through its publishing
     partnership with Pearson, Deitel &amp; Associates,
     Inc. publishes leading-edge programming textbooks,
     professional books and interactive web-based and DVD
     LiveLessons video courses.</p>
 </section>
</body>
</html>
```

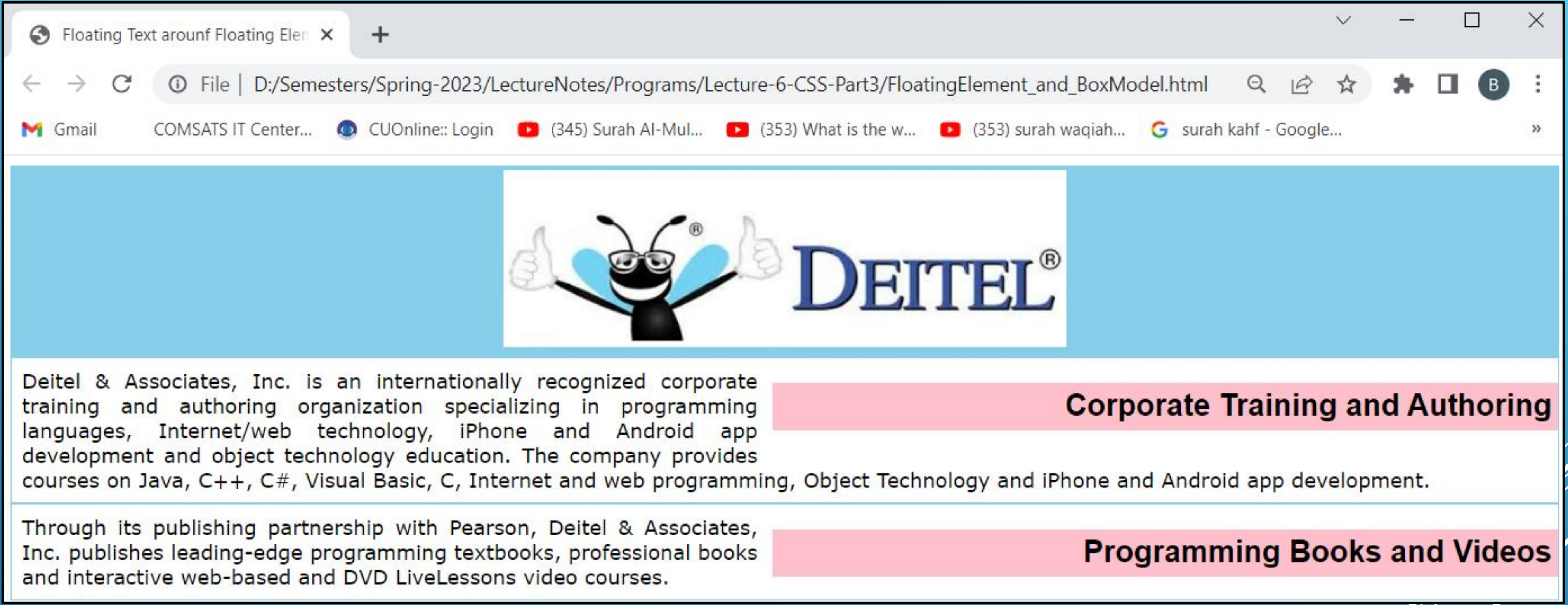
Listing-6.18-FloatingElement_and_BoxModel.html

FLOATING ELEMENTS

- The general structure of **listing-6.18** consists of a header and two main sections.
- Each section contains an **h1** subheading and a paragraph of text.
- Block-level elements (such as sections) render with a *line break before and after their* content, so the header and two sections will render vertically one on top of another.
- In the absence of our styles, the **h1**s that represent our subheadings would also stack vertically on top of the text in the **p** tags.
- However, float property (line#24) to right in the class **floated**, which is applied to the **h1** headings.
    - This causes each **h1** to float to the right edge of its containing element, while the paragraph of text will flow around it.

FLOATING ELEMENTS

Output-Listing-6.18-FloatingElement_and_BoxModel.html


Floating Text arounf Floating Elen
File | D:/Semesters/Spring-2023/LectureNotes/Programs/Lecture-6-CSS-Part3/FloatingElement_and_BoxModel.html
Gmail
COMSATS IT Center...
CUOnline:: Login
(345) Surah Al-Mul...
(353) What is the w...
(353) surah waqiah...
surah kahf - Google...
DEITEL®

Corporate Training and Authoring

Deitel & Associates, Inc. is an internationally recognized corporate training and authoring organization specializing in programming languages, Internet/web technology, iPhone and Android app development and object technology education. The company provides courses on Java, C++, C#, Visual Basic, C, Internet and web programming, Object Technology and iPhone and Android app development.

Programming Books and Videos

Through its publishing partnership with Pearson, Deitel & Associates, Inc. publishes leading-edge programming textbooks, professional books and interactive web-based and DVD LiveLessons video courses.

Figure-6.18-FloatingElement_and_BoxModel.html

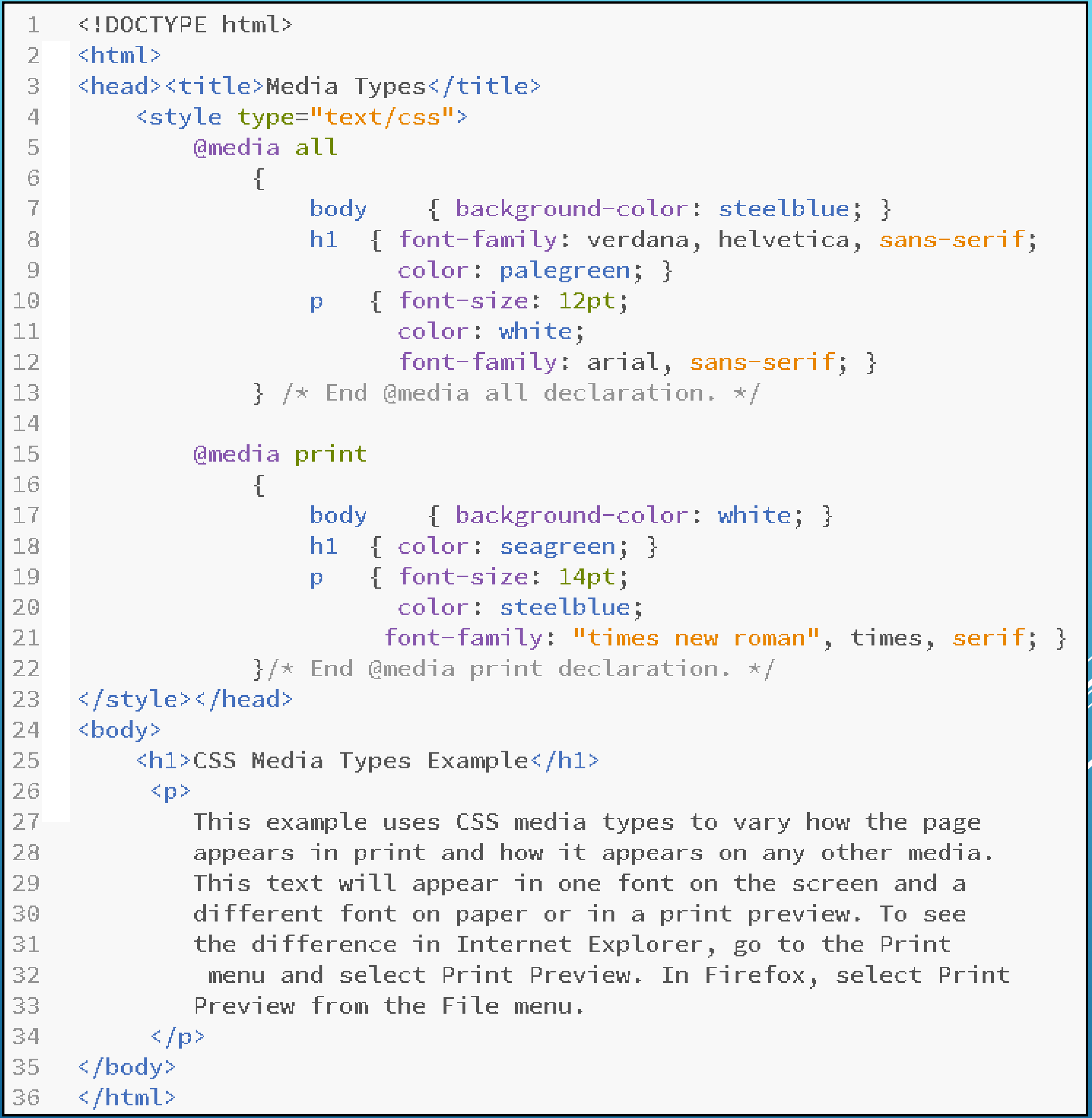
MEDIA TYPES

- The most common media type for a web page is the **screen media type**, which is a standard computer screen.
- Other media types in CSS include **handheld**, **braille**, **speech** and **print**.
  - **handheld** medium is designed for mobile Internet devices such as smartphones,
  - **braille** is for machines that can read or print web pages in braille.
  - **speech** styles allow you to give a speech-synthesizing web browser more information about the content of a page.
  - **print** media type affects a web page's appearance when it's printed.
- Listing-6.19 describing the two types of styles:

1. At line#10---**@media all:** is used for all types of media
2. At line#19---**@media print :** is used for only print media

MEDIA TYPES

Listing-6.19-mediaTypes.html

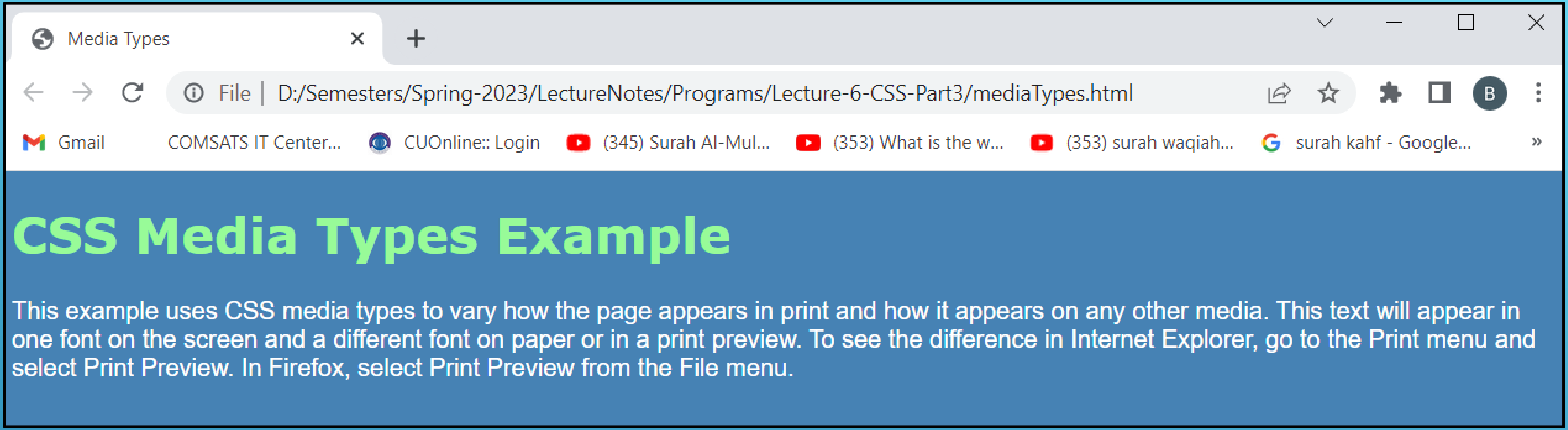


```html
<!DOCTYPE html>
<html>
<head><title>Media Types</title>
    <style type="text/css">
        @media all
            {
                body    { background-color: steelblue; }
                h1  { font-family: verdana, helvetica, sans-serif;
                      color: palegreen; }
                p   { font-size: 12pt;
                      color: white;
                      font-family: arial, sans-serif; }
            } /* End @media all declaration. */

        @media print
            {
                body    { background-color: white; }
                h1  { color: seagreen; }
                p   { font-size: 14pt;
                      color: steelblue;
                     font-family: "times new roman", times, serif; }
            }/* End @media print declaration. */
</style></head>
<body>
    <h1>CSS Media Types Example</h1>
     <p>
        This example uses CSS media types to vary how the page
        appears in print and how it appears on any other media.
        This text will appear in one font on the screen and a
        different font on paper or in a print preview. To see
        the difference in Internet Explorer, go to the Print
         menu and select Print Preview. In Firefox, select Print
        Preview from the File menu.
     </p>
</body>
</html>
```

MEDIA TYPES

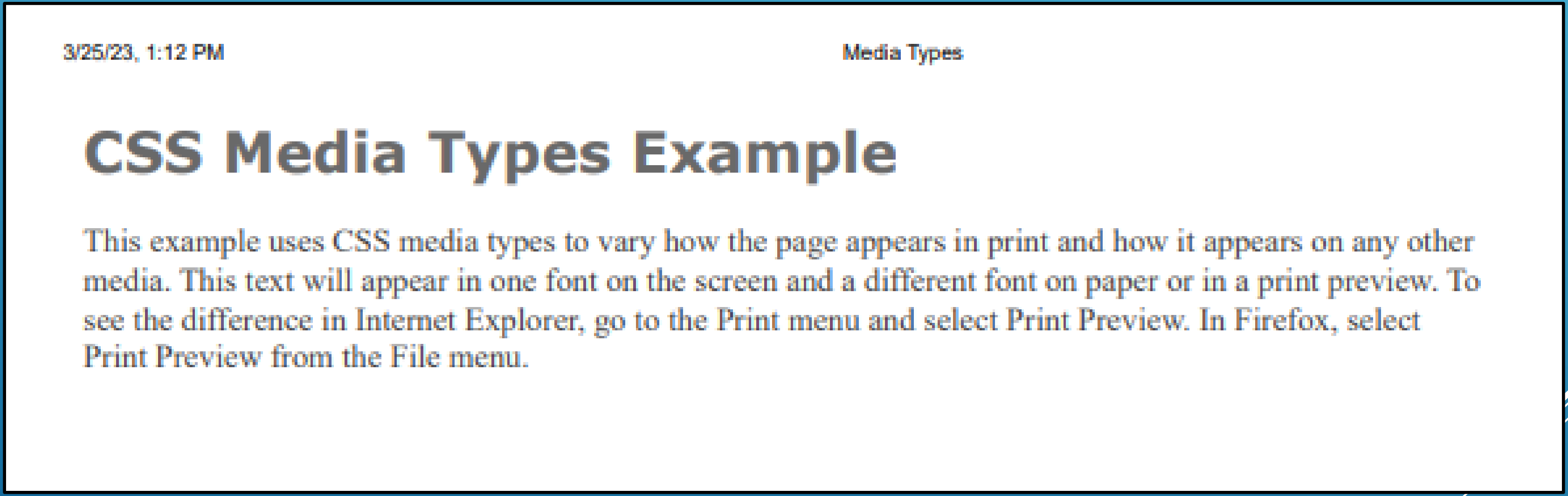
- Note that In listing 6.19- line#5, we use:
  - **@media all:**
    - Set style to define some styles for *all* media types.
- Note that In listing 6.19- line#15, we use:
  - **@media print:**
    - set styles to be applied only when the page is **printed**.
- The styles we applied for all media types look nice on a screen but would ***not*** look good on a printed page.
- A colored background would use a lot of ink, and a black & white printer may print a page that’s hard to read, hence we use @media print style for printing in white background.
- Figure-6.19(a) is displaying output in all media
- Figure-6.19(b) is displaying output in print

MEDIA TYPES



CSS Media Types Example

This example uses CSS media types to vary how the page appears in print and how it appears on any other media. This text will appear in one font on the screen and a different font on paper or in a print preview. To see the difference in Internet Explorer, go to the Print menu and select Print Preview. In Firefox, select Print Preview from the File menu.



3/25/23, 1:12 PM Media Types

CSS Media Types Example

This example uses CSS media types to vary how the page appears in print and how it appears on any other media. This text will appear in one font on the screen and a different font on paper or in a print preview. To see the difference in Internet Explorer, go to the Print menu and select Print Preview. In Firefox, select Print Preview from the File menu.

Figure-6.19(p)----document display for print media

MEDIA QUERIES

- **Media queries** allow you to format content to specific output devices.
- Media queries include a media type and expressions that check the **media features** of the output device.
- Some of the common media features include:
  - **width**—the width of the part of the screen on which the document is rendered, including any scrollbars.
  - **height**—the height of the part of the screen on which the document is rendered, including any scrollbars
  - **device-width**—the width of the screen of the output device
  - **device-height**—the height of the screen of the output device
  - **orientation**—if the height is greater than the width, orientation is portrait, and if the width is greater than the height, orientation is landscape
  - **aspect-ratio**—the ratio of width to height
  - **device-aspect-ratio**—the ratio of device-width to device-height