

CSS TUTORIAL

CHAPTER 1 – CSS INTROUDCTION

WHAT IS CSS?
WHY USE CSS?
ADVANTAGES OF CSS
HISTORY OF CSS

CHAPTER 2 – CSS BACKGROUNDS

BACKGROUND - COLOR
BACKGROUND IMAGE
SET THE BACKGROUND REPEAT
SING BACKGROUND IMAGE-POSITION

CHAPTER 3 - CSS SYNTAX

ELEMENT SELECTOR
THE UNIVERSAL SELECTOR
THE CLASS SELECTORS
THE ID SELECTORS
GROUPING SELECTORS

CHAPTER 4 - CSS STYLES

EXTERNAL STYLE SHEET
INTERNAL STYLE SHEET
INLINE STYLES , MULTIPLE STYLES

CHAPTER 5 - CSS FONTS

SET THE FONT FAMILY
SET THE FONT STYLE
SET THE FONT WEIGHT
SET THE FONT SIZE
SET THE FONT VARIANT

CHAPTER 6 - CSS TEXT

TEXT COLOR
TEXT SHADOW
TEXT ALIGNMENT
TEXT DECORATION
TEXT INDENTATION

CHAPTER 7 - CSS IMAGES

ROUNDED IMAGES
THUMBNAIL IMAGES
RESPONSIVE IMAGES
IMAGE GALLERY
IMAGE OPACITY

CHAPTER 8 - CSS BORDERS

BORDER STYLE
BORDER WIDTH

BORDER COLOR
BORDER - INDIVIDUAL SIDES
BORDER - SHORTHAND PROPERTY

CHAPTER 9 - CSS TABLES

TABLE BORDERS
COLLAPSE TABLE BORDERS
TABLE WIDTH AND HEIGHT
HORIZONTAL ALIGNMENT
VERTICAL ALIGNMENT
TABLE PADDING

HOVERABLE TABLE
TABLE HOVER
RESPONSIVE TABLE

CHAPTER 10 - CSS COLORS

RGB VALUES
RGBA COLORS
HSL COLORS
HSLA COLORS

CHAPTER 11 - CSS MARGINS

INDIVIDUAL SIDES
SHORTHAND PROPERTY
USE OF THE AUTO VALUE

CHAPTER 12 - CSS PADDINGS

PADDING - INDIVIDUAL SIDES
SHORTHAND PROPERTY

CHAPTER 13 - CSS OUTLINES

OUTLINE STYLE
OUTLINE COLOR
OUTLINE WIDTH
OUTLINE - SHORTHAND PROPERTY

CHAPTER 14 - CSS LISTS

DIFFERENT LIST ITEM MAKERS
AN IMAGE AS THE LIST ITEM MAKER
LIST - SHORTHAND PROPERTY
POSITION THE LIST ITEM MAKERS
STYLING LIST WITH COLORS

CHAPTER 15 - CSS DROPPDOWNS

BASIC DROPPDOWN
DROPPDOWN MENU
RIGHT-ALIGNED DROPPDOWN CONTENT

CHAPTER 16 - CSS POSITION

PROPERTY

POSITION: STATIC

POSITION: RELATIVE

POSITION: FIXED

POSITION: ABSOLUTE

OVERLAPPING ELEMENTS

POSITIONING TEXT IN AN IMAGE

CHAPTER 17 - CSS HEIGHT AND WIDTH DIMENSIONS

HEIGHT AND WIDTH PROPERTY

MAX-HEIGHT

SETTING MAX-WIDTH

MIN-HEIGHT PROPERTY

MIN-WIDTH PROPERTY

CHAPTER 18 - CSS LAYOUTS FLOAT AND CLEAR

FLOAT PROPERTY

CLEAR PROPERTY

CLEARFIX HACK-OVERFLOW AUTO

CHAPTER 19 - CSS ANIMATABLE

CHAPTER 20 - CSS ATTRUBUTE SELECTORS

CSS [ATTRIBUTE~="VALUE"] SELECTOR

STYLING FORMS

CHAPTER 21 - CSS THE Z-INDEX PROPERTY

Z-INDEX INTRO

Z-INDEX EXAMPLE

CHAPTER 22 - CSS COMBINATORS

DESCENDANT SELECTOR (SPACE)

CHILD SELECTOR (>)

ADJACENT SIBLING SELECTOR (+)

GENERAL SIBLING SELECTOR (~)

CHAPTER 23 – CSS PSEUDO CLASSES

WHAT ARE PSEUDO CLASSES?

ANCHOR PSEUDO CLASSES

PSEUDO CLASSES & HTML CLASSES

HOVER ON <DIV>

CHAPTER 24 – PSEUDO ELEMENTS

WHAT ARE PSEUDO ELEMENTS?

FIRST PSEUDO ELEMENTS

PSEUDO ELEMENTS & HTML CLASSES

MULTIPLE PSEUDO ELEMENTS

THE :BEFORE PSEUDO ELEMENT

THE :AFTER PSEUDO ELEMENT

THE :MARKER PSEUDO ELEMENT

THE :SELECTION PSEUDO ELEMENT

CHAPTER 25 – NAVIGATION BARS

NAVIGATION BARS = LIST OF LINKS

VERTICAL NAVBAR

HORIZONTAL NAVBAR

CHAPTER 26 – WEBSITE LAYOUT

CREATING HEADER

NAVIGATION BAR

CREATING CONTENT

CREATING FOOTER

FINAL WEBSITE LAYOUT

CHAPTER 1 – CSS INTROUDCTION

WHAT IS CSS?

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using **CSS**, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS saves a lot of work. It can control the layout of multiple web pages all at once. Most commonly, CSS is combined with the markup languages **HTML** or **XHTML**.

WHY USE CSS?

CSS is used to define styles for your **web pages**, including the design, layout and variations in display for different devices and screen sizes.

Before CSS, nearly all of the presentational attributes of **HTML** documents were contained within the **HTML** markup; all font colors, background styles, element alignments, borders and sizes had to be explicitly described, often repeatedly, within the **HTML**. CSS allows authors to move much of that information to another file, the **style sheet**, resulting in considerably simpler **HTML**.

For Example:

under pre-CSS **HTML**, a heading element defined with red text would be written as:

```
<h1><font color="green"> FUTURE VISION </font></h1>
```

Using CSS, the same element can be coded using **style** properties instead of **HTML** presentational attributes:

```
<h1 style="color:green"> FUTURE VISION </h1>
```

ADVANTAGES OF CSS

CSS saves time – You can write CSS once and then reuse the same **sheet** in multiple **HTML** pages. You can define a **style** for each **HTML** element and apply it to as many **web pages** as you want.

CSS saves Lot of Work – The **style** definitions are normally saved in external .CSS files. With an **external stylesheet** file, you can change the look of an entire **website** by changing just one file!

Pages load Faster - If you are using CSS , you do not need to write **HTML tag** attributes every time. Just write one CSS rule of a **tag** and apply it to all the occurrences of that **tag**.

Easy maintenance – To make a **global change**, simply change the **style**, and all the elements in all the **web pages** will be updated automatically.

HISTORY OF CSS

CSS is created and maintained through a group of people within the **W3C** called the CSS Working Group. The CSS Working Group creates documents called specification. When a specification has been discussed and officially ratified by the **W3C** members, it becomes a recommendation.

These ratified specifications are called recommendations because the **W3C** has no control over the actual implementation of the **language**. Independent recommendations about how the **Internet** works and how it should evolve.

CHAPTER 2 – CSS BACKGROUNDS

BACKGROUND - COLOR

The **background-color** property is used to set the background color of an element.

Syntax:

```
Background-color: color_name;
```

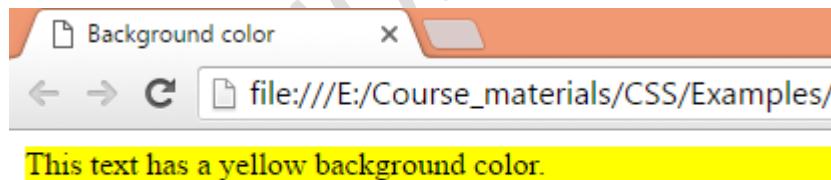
Example:

```
<html>
<head>
<title>Background color</title>
</head>
<body>

<p style="background-color:yellow;">
  This text has a yellow background color.
</p>

</body>
</html>
```

Output:



BACKGROUND IMAGE

The **background-image** property is used to set the background image of an element.

By default, the image is repeated so it covers the entire element.

Syntax:

```
Background-image: url("");
```

Example:

```
<html>
<head>
<title>Background Image</title>
```

```

</head>
<style>
body {
    background-image: url("web_design.jpg");
}
</style>
<body>
<h2> This page has a background image </h2>
</body>
</html>

```

Output:



SET THE BACKGROUND REPEAT

Showing the background image only once is also specified by the `background-repeat` property:

Syntax:

```

background-image: url("");
background-repeat: no-repeat;

```

Example:

```

<html>
<head>
<title>Background No Repeat</title>
</head>
<style>
body {
    background-image: url("web_design.jpg");
    background-repeat: no-repeat;
}

```

```
</style>
<body>
    <h2> This page has a background image </h2>
</body>
</html>
```

Output:



USING BACKGROUND IMAGE-POSITION

The position of the image is specified by the `background-position` property:

Syntax:

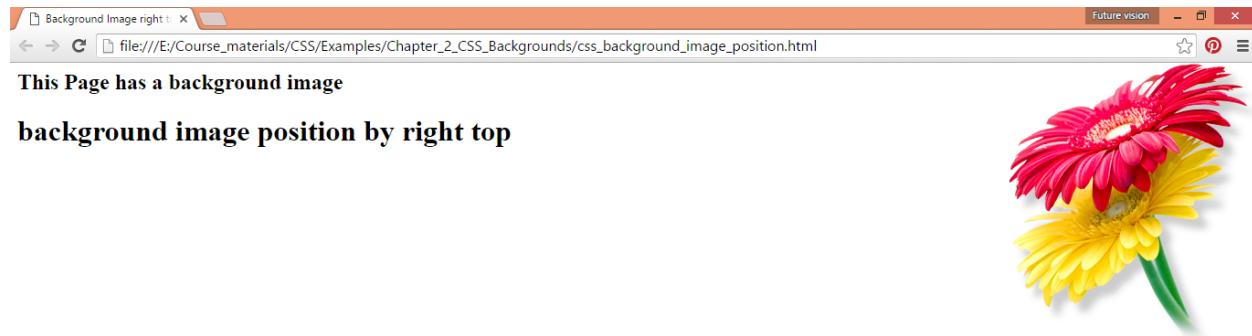
```
background-image: url("");
background-repeat: no-repeat;
background-position: right top;
```

Example:

```
<html>
<head>
<title>Background No Repeat</title>
</head>
<style>
body {
    background-image: url("web_design.jpg");
    background-repeat: no-repeat;
    background-position: right top;
}
</style>
```

```
<body>
<h2> This page has a background image </h2>
</body>
</html>
```

Output:



USING BACKGROUND SHORTHAND PROPERTY

The use the shorthand property to set the background properties in one declaration:

Syntax:

```
background: #ffffff url("img_tree.png") no-repeat right top;
```

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background: #ffffff url("img_tree.png") no-repeat right top;
    margin-right: 200px;
}
</style>
</head>
<body>
<h1>The background Property</h1>
<p>The background property is a shorthand property for specifying all the background properties in one declaration.</p>
```

```
<p>Here, the background image is only shown once, and it is also positioned in  
the top-right corner.</p>  
<p>We have also added a right margin, so that the text will not write over the  
background image.</p>  
</body>  
</html>
```

Output:

The background Property

The background property is a shorthand property for specifying all the background properties in one declaration.

Here, the background image is only shown once, and it is also positioned in the top-right corner.

We have also added a right margin, so that the text will not write over the background image.



CHAPTER 3 - CSS SYNTAX

CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts:

ELEMENT SELECTOR

A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

Property: A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties . They could be color, border, etc.

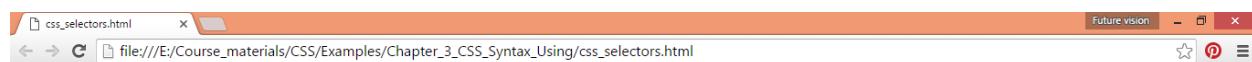
Value : Values are assigned to properties. For example, color property can have the value either blue or # 0000FF etc.

Example:

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

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
    p {  
        color: blue;  
        text-align: center;  
    }  
</style>  
</head>  
<body>  
  
<h1>Future Vision Computers</h1>  
<p>Future Vision Computer Institute was founded in 2006 with the mission of providing  
best quality Computer education to all classes of people at a very reasonable fee  
structure. Thousands of students have already got trained professionally and made their  
career successfully in the past.</p>  
</body>  
  
</html>
```

Output:



Future Vision Computers

Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.

THE UNIVERSAL SELECTOR

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any elements type:

Example:

```
* {  
    color: #ff0000;  
}
```

This rule renders the content of every element in our document in black.

THE CLASS SELECTORS

You can define style rules based on the **class** attribute of the elements. All the elements having that class will be formatted according to the defined rule. we can use **classes** on any elements, multiple times on a single web page

Syntax:

Class is defined as **dot(.)** with class name as follows:

```
.class_name{  
    Color:red;  
}
```

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.red {  
    text-align: center;  
    color: red;  
}  
</style>  
</head>  
<body>
```

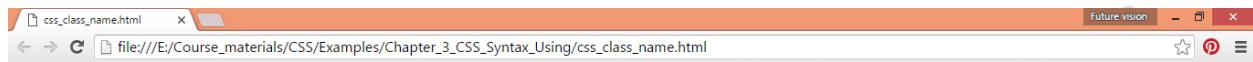
<h1 class="red"> Future Vision Computers </h1>

<p class="red"> Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable

fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.</p>

```
</body>
</html>
```

Output:



Future Vision Computers

Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.

There is also another rule which we are giving below for using css. This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example –

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.color {
  color: #000000;
}
</style>
</head>
<body>

<h1 class="color"> Future Vision Computers </h1>
</body>
</html>
```

Output:

Future Vision Computers

THE ID SELECTORS

You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

Syntax:

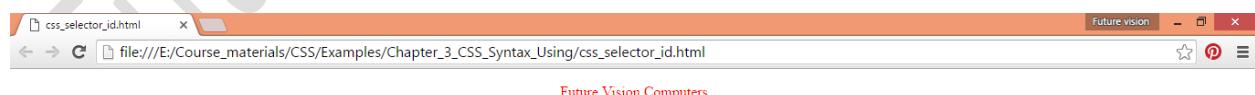
Class is defined as dot(.) with class name as follows:

```
#class_id{  
    Color: #ff0000;  
}
```

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#red {  
    text-align: center;  
    color: #ff0000;  
}  
</style>  
</head>  
<body>  
<p id="red"> Future Vision Computers </p>  
<p> Future Vision Computer Institute was founded in 2006 with the mission of providing  
best quality Computer education to all classes of people at a very reasonable fee  
structure. Thousands of students have already got trained professionally and made their  
career successfully in the past.</p>  
</body>  
</html>
```

Output:



Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.

There is also another rule which we are giving below for using css. This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example –

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1#color {
  color: #000000;
}
</style>
</head>
<body>

<h1 id="color"> Future Vision Computers </h1>
</body>
</html>
```

Output:

Future Vision Computers



GROUPING SELECTORS

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example:

```
h1, h2, h3 {  
    color: #36C;  
    font-weight: normal;  
    letter-spacing: .4em;  
    margin-bottom: 1em;  
    text-transform: lowercase;  
}
```

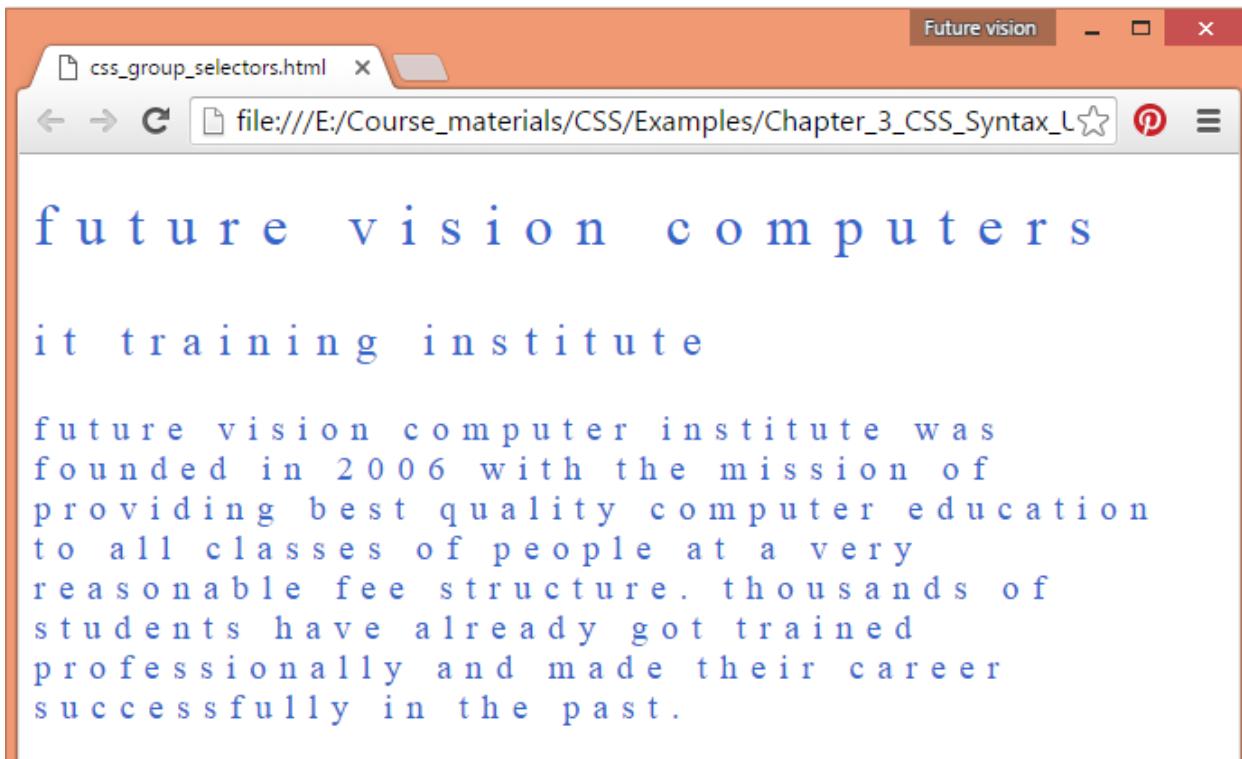
This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *class* selectors together as shown below:

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1, h2, h3 {  
    color: #36C;  
    font-weight: normal;  
    letter-spacing: .4em;  
    margin-bottom: 1em;  
    text-transform: lowercase;  
}  
  
</style>  
</head>  
<body>  
  
<h1>Future Vision Computers </h1>  
<h2>IT Training Institute</h2>  
<h3> Future Vision Computer Institute was founded in 2006 with the mission of providing  
best quality Computer education to all classes of people at a very reasonable fee  
structure. Thousands of students have already got trained professionally and made their  
career successfully in the past.</h3>  
  
</body>  
</html>
```

Output:



The screenshot shows a web browser window titled "Future vision" displaying the content of "css_group_selectors.html". The URL in the address bar is "file:///E:/Course_materials/CSS/Examples/Chapter_3_CSS_Syntax_L". The page content is as follows:

```
future vision computers  
it training institute  
  
future vision computer institute was  
founded in 2006 with the mission of  
providing best quality computer education  
to all classes of people at a very  
reasonable fee structure. thousands of  
students have already got trained  
professionally and made their career  
successfully in the past.
```

CHAPTER 4 - CSS STYLES

When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.

Three ways to insert CSS Style Sheets:

- External style sheet
- Internal style sheet
- Inline style

EXTERNAL STYLE SHEET:

With an external style sheet, you can change the look of an entire website by changing just one file! Each page must include a reference to the external style sheet file inside the `<link>` element. The `<link>` element goes inside the `<head>` section:

Example:

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

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a `.css` extension.

mystyle.css file

```
body{
    background-color: lightgreen;
}

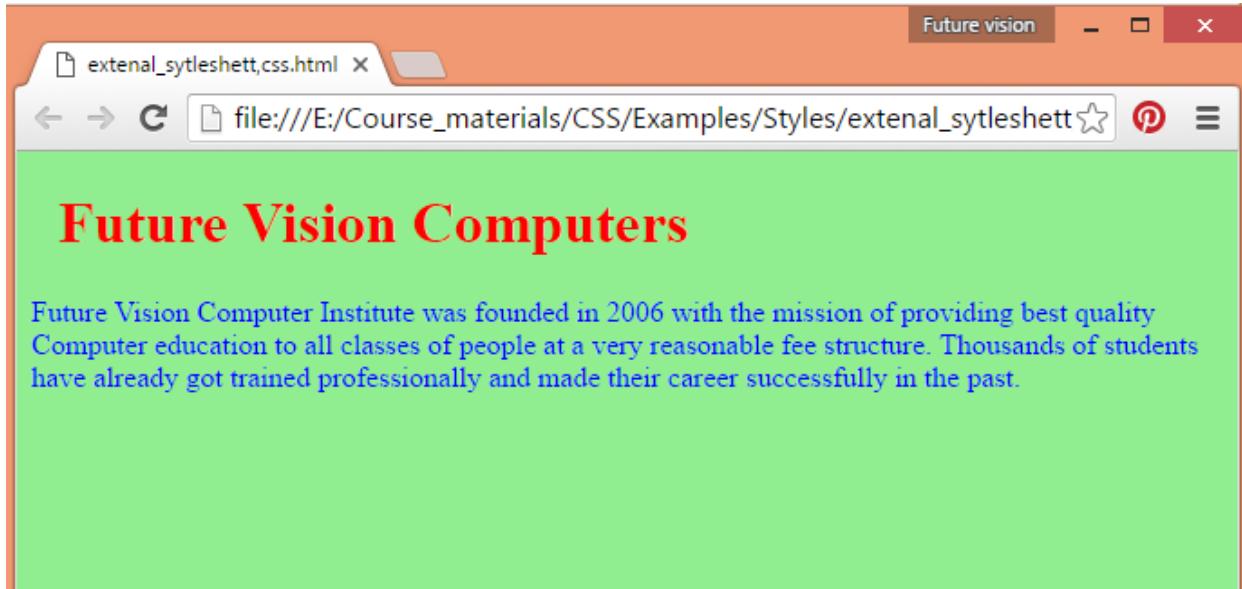
h1{
    color: Red;
    margin-left: 15px;
}

p{
    color:blue;
}

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="mystyle.css"/>
</head>
<body>
    <h1> Future Vision Computers </h1>
    <p> Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee
```

```
structure. Thousands of students have already got trained professionally and made their  
career successfully in the past.</p>  
</body>  
</html>
```

Output:



INTERNAL STYLE SHEET:

An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the `<style>` element, inside the `<head>` section of an HTML page:

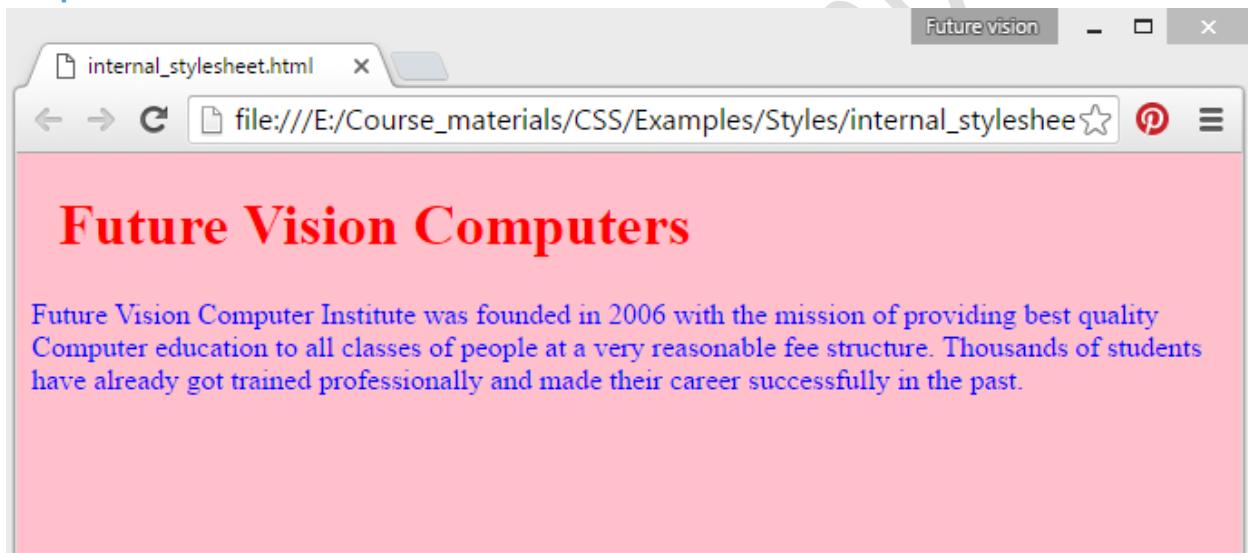
Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body{  
background-color: pink;  
}  
  
h1{  
color: Red;  
margin-left: 15px;  
}  
  
p{
```

```
        color:blue;
    }

</style>
</head>
<body>
<h1> Future Vision Computers </h1>
<p> Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.</p>
</body>
</html>
```

Output:



INLINE STYLE:

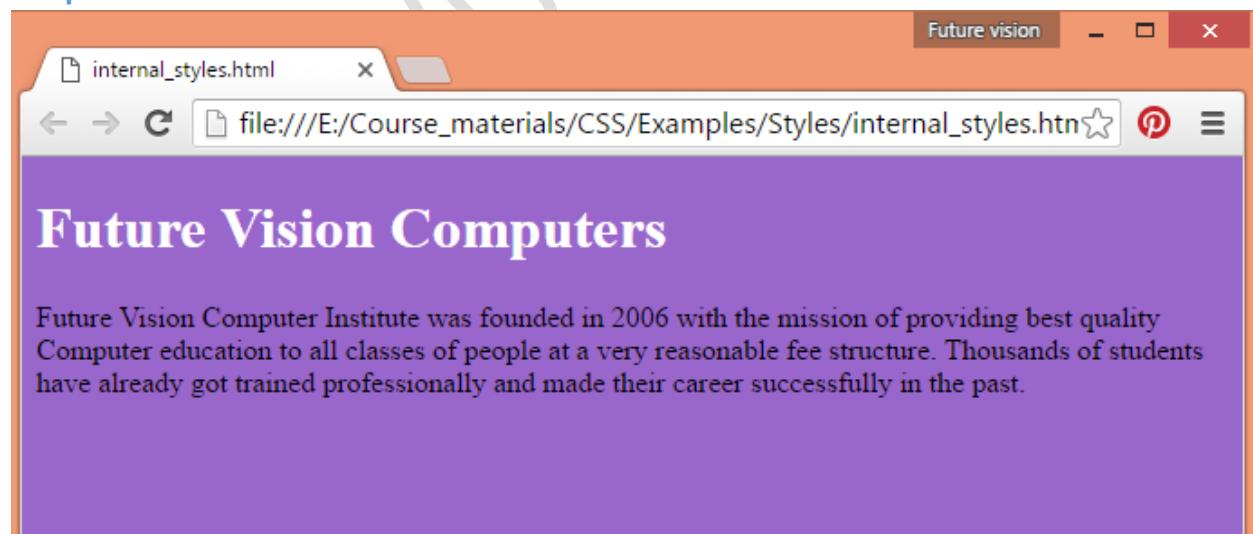
An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property. The example below shows how to change the color and the left margin of a `<h1>` element:

```
<h1  
style="color:red;  
margin-left:50px;">Future Vision.</h1>
```

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
</head>  
<body style="background-color:#96C">  
<h1 style="color:#fff">Future Vision Computers </h1>  
<p> Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.</p>  
</body>  
</html>
```

Output:



MULTIPLE STYLE:

If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

Examples

Assume that an external style sheet has the following style for the `<h1>` element:

Mystyle.css file

```
h1 {  
    color: green;  
}
```

then, assume that an internal style sheet also has the following style for the `<h1>` element:

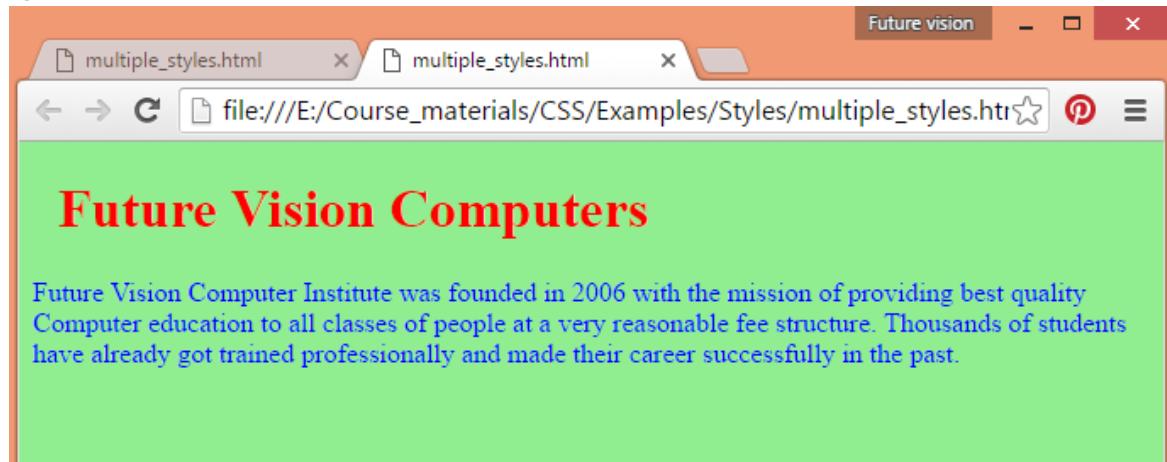
```
h1 {  
    color: red;  
}
```

Example 1:

If the internal style is defined after the link to the external style sheet, the `<h1>` elements will be "red":

```
<!DOCTYPE html>  
<html>  
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
<style>  
h1 {  
    color: red;  
}  
</style>  
</head>  
<body>  
<h1> Future Vision Computers </h1>  
<p> Future Vision Computer Institute was founded in 2006 with the mission of providing  
best quality Computer education to all classes of people at a very reasonable fee  
structure. Thousands of students have already got trained professionally and made their  
career successfully in the past.</p>  
</body>  
</html>
```

Output:



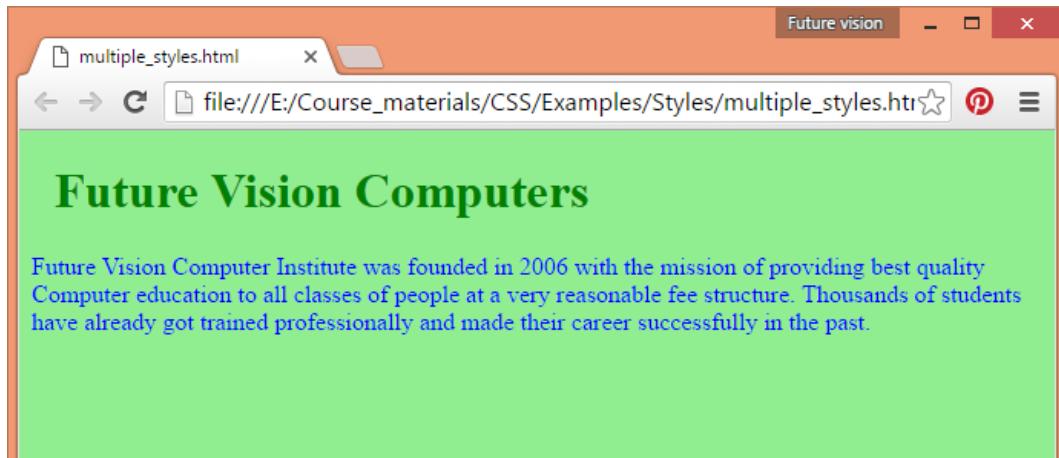
Example 2:

However, if the internal style is defined before the link to the external style sheet, the `<h1>` elements will be "green":

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    color: red;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>

<body>
<h1> Future Vision Computers </h1>
<p> Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.</p>
</body>
</html>
```

Output:



Extra Practice 

FutureVisionComputers

CHAPTER 5 - CSS FONTS

This chapter teaches you how to set fonts of a content, available in an HTML element. You can set the following font properties of an element:

- ✓ The **font-family** property is used to change the face of a font.
- ✓ The **font-style** property is used to make a font italic or oblique.
- ✓ The **font-weight** property is used to increase or decrease how bold or light a font appears.
- ✓ The **font-size** property is used to increase or decrease the size of a font.

- ✓ The **font-variant** property is used to create a small-caps effect.

SET THE FONT FAMILY:

The font family of a text is set with the **font-family** property.

The **font-family** Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Example:

```
p {  
    font-family: "Arial Black", Gadget, sans-serif  
}
```

SET THE FONT STYLE:

The **font-style** property is mostly used to specify italic text.

This property has three values:

- ✓ **normal** - The text is shown normally
- ✓ **italic** - The text is shown in italics
- ✓ **oblique** - The text is "leaning"

Example:

```
p.normal {  
    font-style: normal;  
}  
p.italic {  
    font-style: italic;  
}  
  
p.oblique {
```

```
        font-style: oblique;
    }

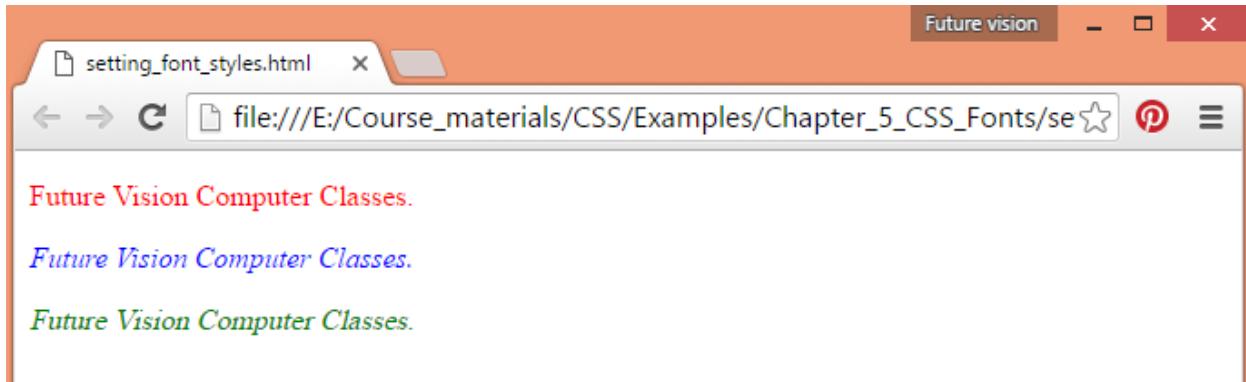
    p.normal {
        font-style: normal;
    }

    p.italic {
        font-style: italic;
    }

    p.oblique {
        font-style: oblique;
    }

<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
    font-style: normal;
    color:red;
}
p.italic {
    font-style: italic;
    color:blue;
}
p.oblique {
    font-style: oblique;
    color:green;
}
</style>
</head>
<body>
<p class="normal">Future Vision Computer Classes.</p>
<p class="italic">Future Vision Computer Classes.</p>
<p class="oblique">Future Vision Computer Classes.</p>
</body></html>
```

Output:



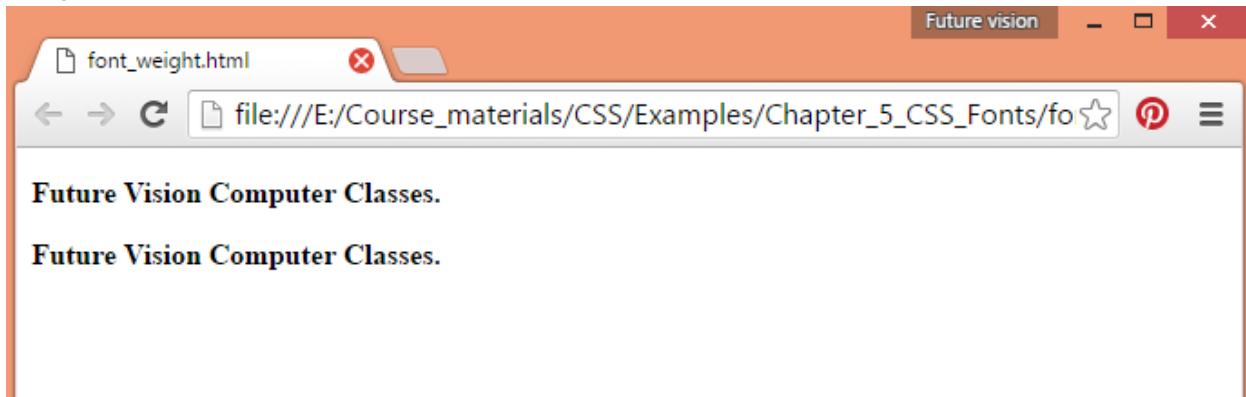
SET THE FONT WEIGHT

The `font-weight` property specifies the weight of a font:

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.bolder {
    font-weight: bolder;
}
p.bold {
    font-weight: bold
}
</style>
</head>
<body>
<p class="bolder">Future Vision Computer Classes.</p>
<p class="bold">Future Vision Computer Classes.</p>
</body>
</html>
```

Output:



SET THE FONT SIZE

The `font-size` property sets the size of the text. Setting the text size with pixels gives you full control over the text size. The following example demonstrates how to set the font size of an element. The font-size property is used to control the size of fonts.

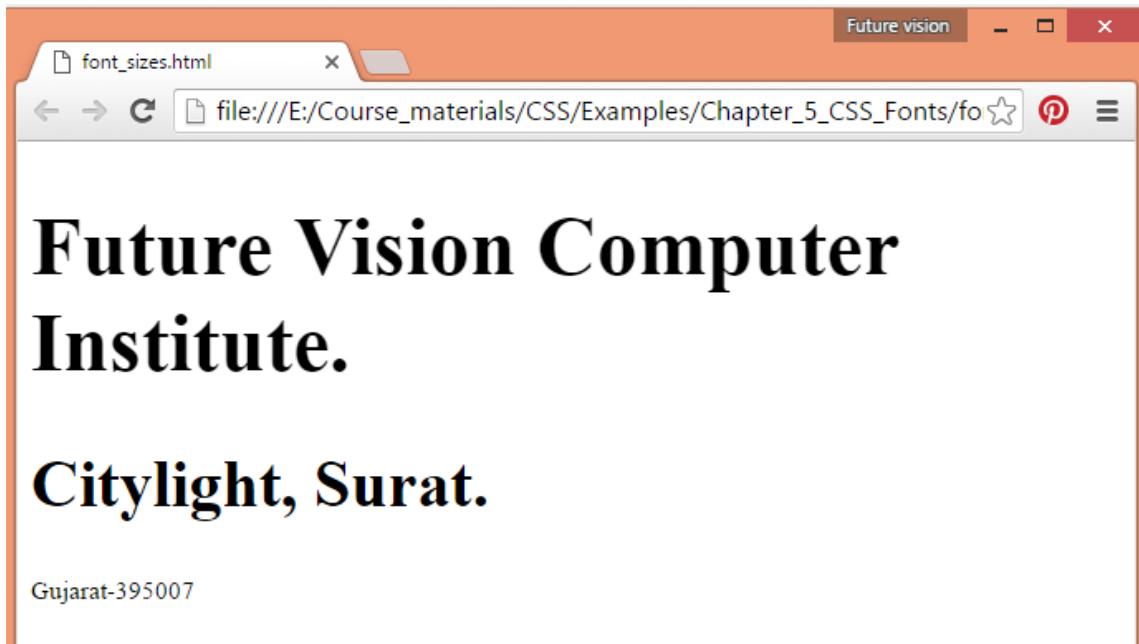
Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
    h1 {
        font-size: 50px;
    }

    h2 {
        font-size: 40px;
    }

    p {
        font-size: 15px;
    }
</style>
</head>
<body>
<h1>Future Vision Computer Institute.</h1>
<h2>Citylight, Surat.</h2>
<p>Gujarat-395007</p>
</body>
</html>
```

Output:



SET THE FONT VARIANT

The **font-variant** property specifies whether or not a text should be displayed in a small-caps font. In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

The following example demonstrates how to set the font variant of an element.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.inherit {
    font-variant: inherit;
}
p.small {
    font-variant: small-caps;
}
</style>
</head>
<body>
<h1>Future Vision Computer Institute.</h1>
<h2>Our Courses</h2>
```

```
<p class="inherit">Web Designing, Web Development, Digital Marketing</p>
<p class="small">Java Programming, PHP Programming</p>
</body>
</html>
```

Output:



FONT SHORTHAND PROPERTY

To shorten the code, it is also possible to specify all the individual font properties in one property.

The following example demonstrates.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
    font: 20px Arial, sans-serif;
}
p.small {
    font-variant: small-caps;
}
</style>
</head>
<body>
<h1>Future Vision Computer Institute.</h1>
<h2>Our Courses</h2>
```

```
<p class="a">Web Designing, Web Development, Digital Marketing</p>
<p class="small">Java Programming, PHP Programming</p>
</body>
</html>
```

Output:



CHAPTER 6 - CSS TEXT

This chapter teaches you how to manipulate text using CSS properties. You can set the following text properties of an element:

- ✓ The color property is used to set the color of a text.
- ✓ The text-shadow property is used to set the text shadow around a text
- ✓ The text-align property is used to align the text of a document
- ✓ The text-decoration property is used to underline, overline, and strikethrough text.
- ✓ The text-indent property is used to indent the text of a paragraph.
- ✓ The text-decoration property is used to underline, overline, and strikethrough text.
- ✓ The text-transform property is used to capitalize text or convert text to uppercase or lowercase letters.
- ✓ The letter-spacing property is used to add or subtract space between the words of a sentence.

TEXT COLOR

The `color` property is used to set the color of the text. With CSS, a color is most often specified by:

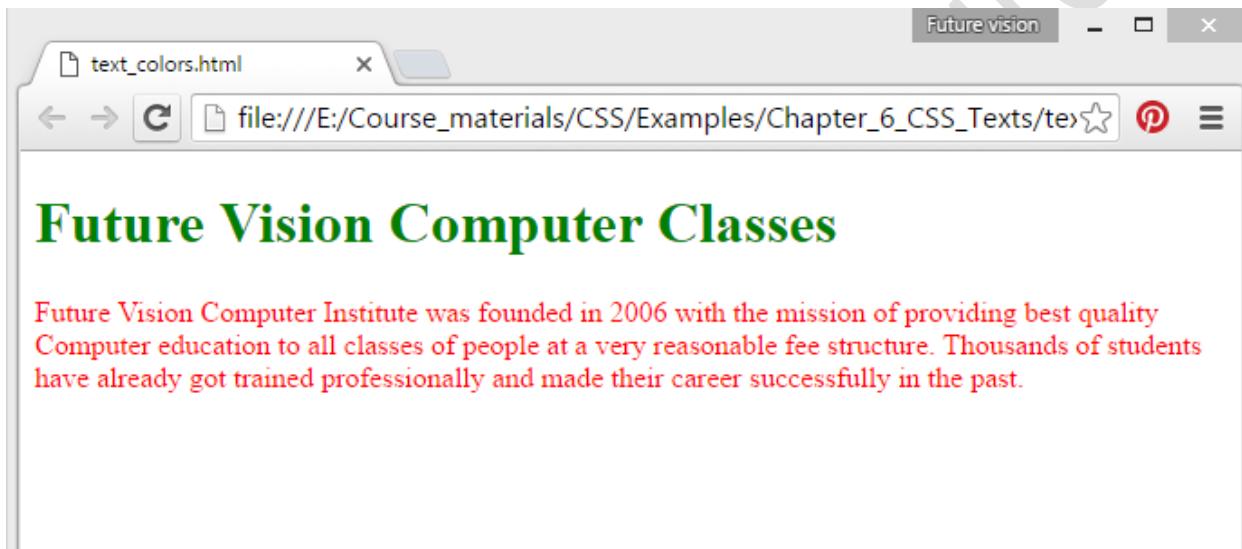
- ✓ a HEX value - like "#00FF00"
- ✓ an RGB value - like "rgb(0,255,0)"
- ✓ a color name - like "green"

Example:

```
body {  
    color: red;  
}  
  
h1 {  
    color: green;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    color: red;  
}  
  
h1 {  
    color: green;  
}  
</style>
```

```
</head>
<body>
<h1>Future Vision Computer Classes</h1>
<p> Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.</p>
</body>
</html>
```

Output:



TEXT SHADOW

The text-shadow property adds shadow to text.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    text-align: center;
    font: italic 70px Sans-Serif;
    padding: 50px 0;
}
p {
    text-shadow : 1px 1px 1px #000;
```



```

}
</style>
</head>
<body>
<h1>Future Vision Computer Classes</h1>
<p> Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.</p>
</body>
</html>

```

Output:



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.

Example:

```

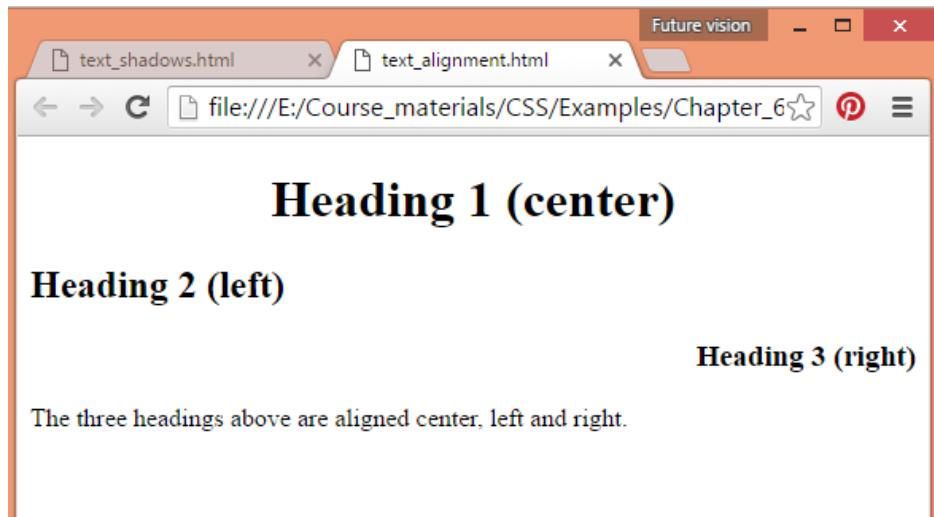
h1 {
    text-align: center;
}

h2 {
    text-align: left;
}

```

```
h3 {  
    text-align: right;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    text-align: center;  
}  
h2 {  
    text-align: left;  
}  
h3 {  
    text-align: right;  
}  
</style>  
</head>  
<body>  
<h1>Heading 1 (center)</h1>  
<h2>Heading 2 (left)</h2>  
<h3>Heading 3 (right)</h3>  
<p>The three headings above are aligned center, left and right.</p>  
</body>  
</html>
```

Output:



TEXT DECORATION

The **text-decoration** CSS property is used to set the text formatting to underline, overline, line-through or blink.

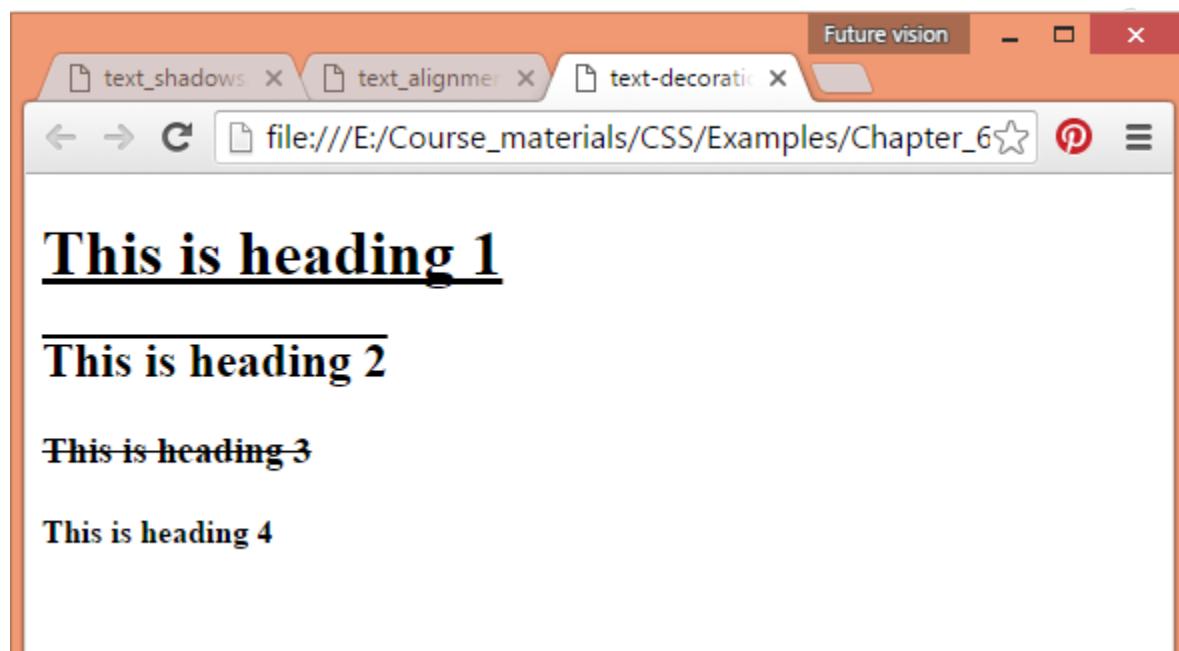
Underline and overline decorations are positioned under the text, line-through over it.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1#under {
text-decoration: underline;
}
H2#over {
text-decoration: overline;
}
H3#line {
text-decoration: line-through;
}
H4#blink {
text-decoration: blink;
}
</style>
</head>
<body>
<h1 id="under">This is heading 1</h1>
```

```
<h2 id="over">This is heading 2</h2>
<h3 id="line">This is heading 3</h3>
<h4 id="blink">This is heading 4</h4>
</body>
</html>
```

Output:



TEXT INDENTATION

The **text-indent** property specifies how much horizontal space should be left before the beginning of the first line of the text content of an element.

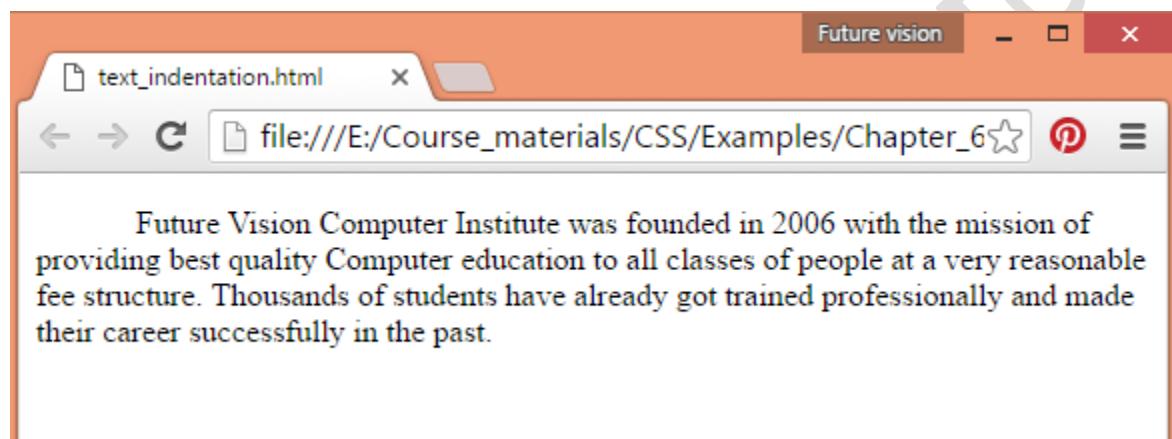
Horizontal spacing is with respect to the left (or right, for right-to-left layout) edge of the containing block element's box.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    text-indent: 50px;
}
</style>
```

```
</head>
<body>
<p>Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past.</p>
</body>
</html>
```

Output:



CHAPTER 7 - CSS IMAGES

Images play an important role in any webpage. Though it is not recommended to include a lot of images, but it is still important to use good images wherever required. CSS plays a good role to control image display.

You can set the following image properties using CSS.

- ✓ The **border** property is used to set the width of an image border.
- ✓ The **height** property is used to set the height of an image.
- ✓ The **width** property is used to set the width of an image.
- ✓ The **Opacity** property is used to set the opacity of an images.

ROUNDED IMAGES

Use the **border-radius** property to create rounded images:

Rounded Images:-

```
img {  
    border-radius: 7px;  
}
```

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    border-radius: 7px;  
}  
</style>  
</head>  
<body>  
<h2>Rounded Images</h2>  
<p>Use the border-radius property to create rounded images:</p>  
  
</body>  
</html>
```

Output:



- Circle Images:-

```
img {  
    border-radius: 60%;  
}
```

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    border-radius: 50%;  
}  
</style>  
</head>  
<body>  
<h2>Rounded Images</h2>
```

```
<p>Use the border-radius property to create rounded images:</p>

</body>
</html>
```

Output:



THUMBNAIL IMAGES

Use the **border** property to create thumbnail images.

Thumbnail Images:-

```
img {
    border: 2px solid #000;
    border-radius: 7px;
    padding: 6px;
}
```

Example:

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

```
<head>
<style>
img {
    border: 2px solid #000;
    border-radius: 7px;
    padding: 6px;
}
</style>
</head>
<body>
<h2>Thumbnail Images</h2>
<p>Use the border property to create thumbnail images:</p>

</body>
</html>
```

Output:



RESPONSIVE IMAGES

Responsive images will automatically adjust to fit the size of the screen.

```
img {  
    max-width: 100%;  
    height: auto;  
}
```

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
    img {  
        max-width: 100%;  
        height: auto;  
    }  
</style>  
</head>  
<body>  
  
<h2>Responsive Images</h2>  
<p>Responsive images will automatically adjust to fit the size of the screen.</p>  
  
</body>  
</html>
```

Output:



IMAGE GALLERY

CSS can be used to create an image gallery.

Example:

```
<html>
<head>
<style>
div.img {
    margin: 4px;
    border: 2px solid #ddd;
    float: left;
    width: 200px;
    padding:1px;
}

div.img:hover {
    border: 2px solid #2ba;
}

div.img img {
    width: 100%;
    height: auto;
}

div.desc {
    padding: 20px;
    text-align: center;
}
</style>
</head>
<body>

<div class="img">
    <a target="_blank" href="#">
         </a>
        <div class="desc">Add a description of the image here</div>
    </div>

<div class="img">
    <a target="_blank" href="#">
        
    </a>
    <div class="desc">Add a description of the image here</div>
</div>
```

```

<div class="img">
  <a target="_blank" href="#">
    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="img">
  <a target="_blank" href="#">
    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

</body>
</html>

```

Output:



IMAGE OPACITY

Creating transparent images with CSS is easy.

Example 1 - Creating a Transparent Image

First we will show you how to create a transparent image with CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    opacity: 0.5;
    filter: alpha(opacity=50);
}
</style>
</head>
<body>
<h1>Image Transparency</h1>

</body>
</html>
```

Output:



Example 2 - Image Transparency - Hover Effect

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    opacity: 0.5;
    filter: alpha(opacity=50);
}
img:hover {
    opacity: 1.0;
    filter: alpha(opacity=100);
}
</style>
</head>
<body>
<h1>Image Transparency</h1>


</body>
</html>
```

Output:



CHAPTER 8 - CSS BORDERS

The CSS border properties allow you to specify the style, width, and color of an element's border.

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

Example:

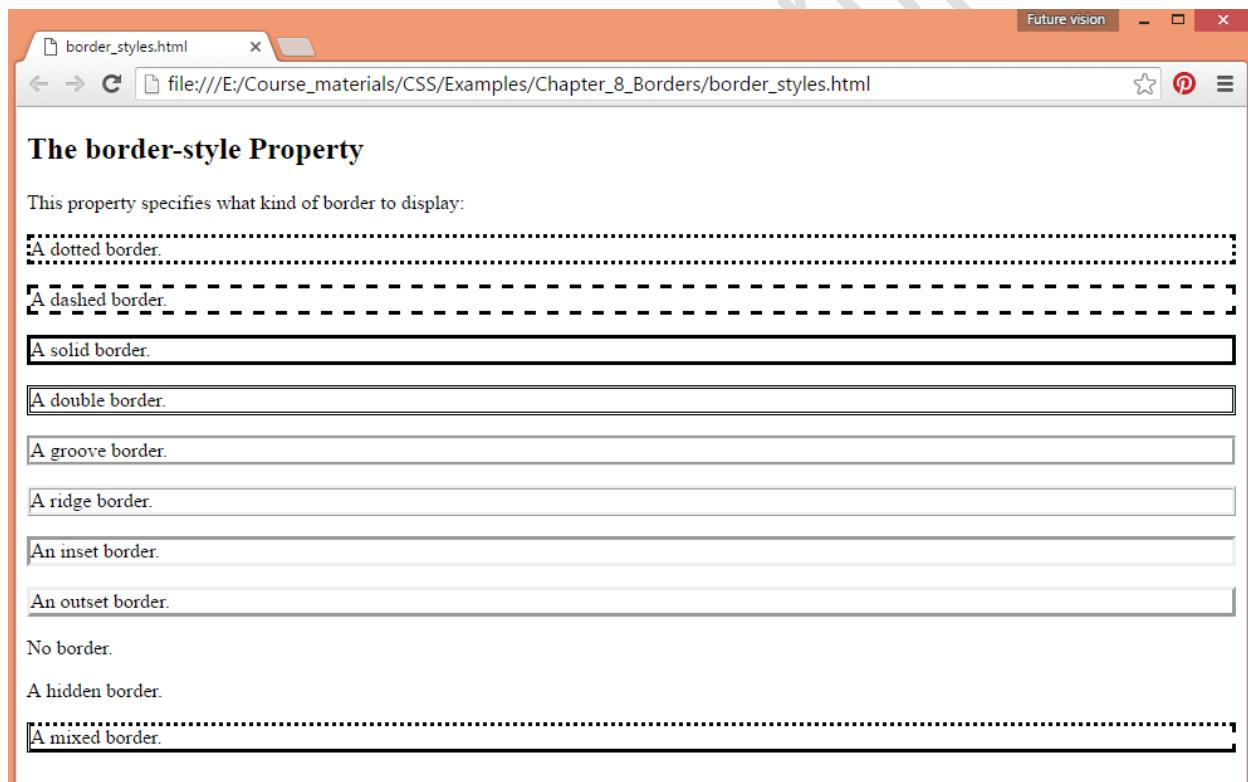
```
<!DOCTYPE html>
<html>
<head>
<style>
    p.dotted {border-style: dotted;}
    p.dashed {border-style: dashed;}
    p.solid {border-style: solid;}
    p.double {border-style: double;}
    p.groove {border-style: groove;}
    p.ridge {border-style: ridge;}
    p.inset {border-style: inset;}
    p.outset {border-style: outset;}
    p.none {border-style: none;}
    p.hidden {border-style: hidden;}
    p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>
```

```
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>

</body>
</html>
```

Output:



BORDER WIDTH

The **border-width** property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
    p.one {
        border-style: solid;
        border-width: 6px;
    }

    p.two {
        border-style: solid;
        border-width: medium;
    }

    p.three {
        border-style: dotted;
        border-width: 3px;
    }

    p.four {
        border-style: dotted;
        border-width: thin;
    }

    p.five {
        border-style: double;
        border-width: 16px;
    }

    p.six {
        border-style: double;
        border-width: thick;
    }

    p.seven {
        border-style: solid;
        border-width: 3px 9px 5px 25px;
    }
}
```

```

        }
</style>
</head>
<body>

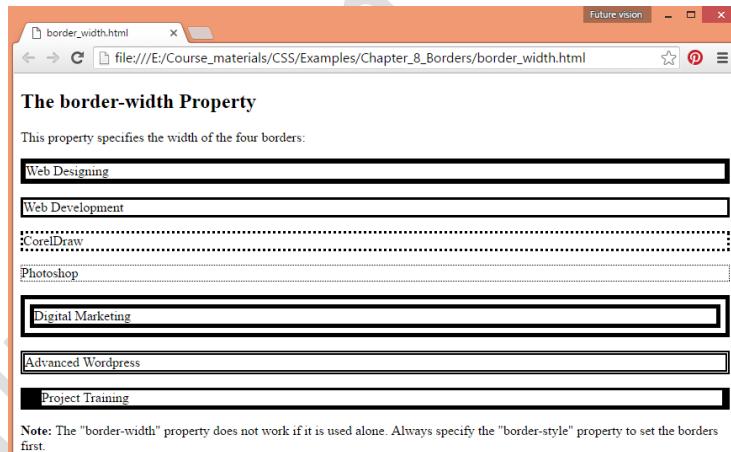
<h2>Future Vision Computer Institute</h2>
<p>This property specifies the width of the four borders:</p>

<p class="one">Web Designing</p>
    <p class="two">Web Development</p>
    <p class="three">CorelDraw</p>
    <p class="four">Photoshop</p>
    <p class="five">Digital Marketing</p>
<p class="six">Advanced Wordpress </p>
<p class="seven"> Project Training </p>

<p><b>Note:</b> The "border-width" property does not work if it is used alone.  
Always specify the "border-style" property to set the borders first.</p>
</body>
</html>

```

Output:



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"
- ✓ RGB - specify a RGB value, like "rgb(255,0,0)"
- ✓ Hex - specify a hex value, like "#ff0000"
- ✓ transparent

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

If **border-color** is not set, it inherits the color of the element.

Border-Color Styles:

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

BORDER - INDIVIDUAL SIDES

In CSS, these properties are for specifying each of the borders (top, right, bottom, and left):

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

The example above gives the same result as this:

- ```

p {
 border-style: dotted solid groove;
}

```
- ✓ If the **border-style** property has four values:
  - ✓ **border-style: dotted solid double dashed;**
    - o top border is dotted
    - o right border is solid
    - o bottom border is double
    - o left border is dashed
  - ✓ If the **border-style** property has three values:
  - ✓ **border-style: dotted solid double;**
    - o top border is dotted
    - o right and left borders are solid
    - o bottom border is double
  - ✓ If the **border-style** property has two values:
  - ✓ **border-style: dotted solid;**
    - o top and bottom borders are dotted
    - o right and left borders are solid
  - ✓ If the **border-style** property has one value:
  - ✓ **border-style: dotted;**
    - o all four borders are dotted

### Example:

```

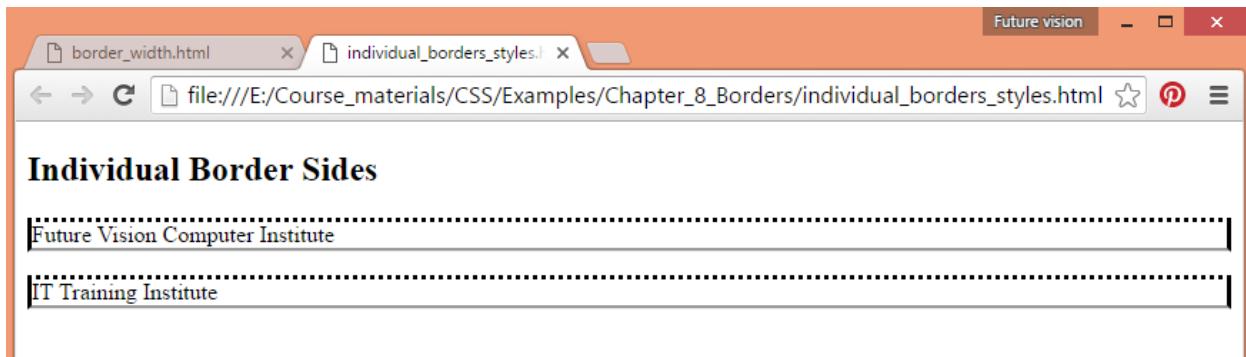
<!DOCTYPE html>

<html>
<head>
<style>
p {
 border-style: dotted solid groove;
}

</style>
</head>
<body>
 <h2>Individual Border Sides</h2>
 <p>Future Vision Computer Institute</p>
 <p>IT Training Institute</p>
</body>
</html>

```

## Output:



## BORDER - SHORTHAND PROPERTY

To shorten the code, it is also possible to specify all the individual border properties in one property.

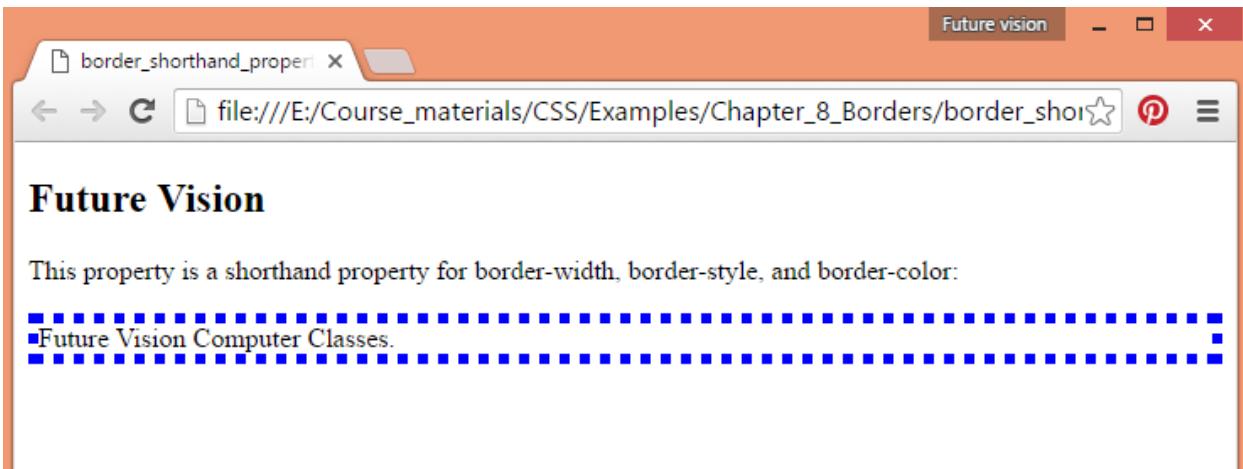
The **border** property is a shorthand property for the following individual border properties:

- ✓ **border-width**
- ✓ **border-style** (required)
- ✓ **border-color**

## Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 div {
 border: 6px dotted blue;
 }
</style>
</head>
<body>
<h2>Future Vision</h2>
<p>This property is a shorthand property for border-width, border-style, and border-color:</p>
<div> Future Vision Computer Classes.</div>
</body>
</html>
```

## Output:



## CHAPTER 9 - CSS TABLES

This chapter teaches you how to set different properties of an HTML table using CSS. You can set the following properties of a table:

The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.

The **border-spacing** specifies the width that should appear between table cells.

The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.

The **empty-cells** specifies whether the border should be shown if a cell is empty.

The **table-layout** allows browsers to speed up the layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

### TABLE BORDERS

To specify table borders in CSS, use the **border** property. The example below specifies a black border for <table>, <th>, and <td> elements:

```
table,th,td {
 border: 2px solid red;
}
```

**Example:**

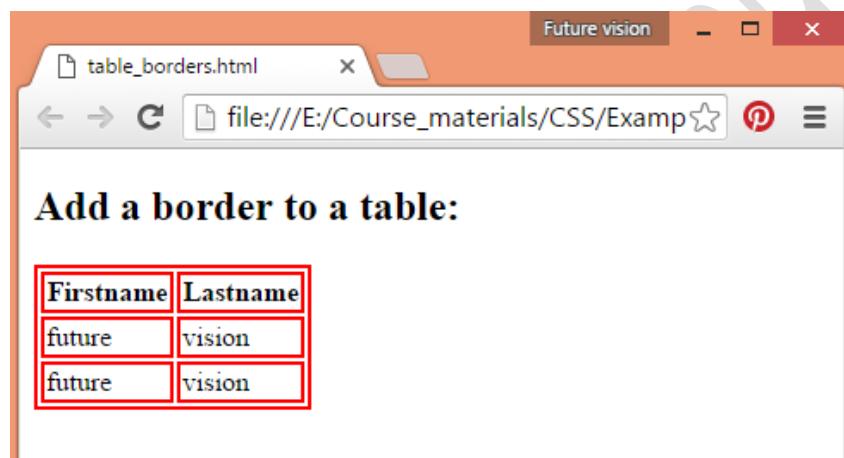
```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
 border: 2px solid red;
}
</style>
</head>
<body>
<h2>Add a border to a table:</h2>
<table>
<tr>
 <th>Firstname</th>
 <th>Lastname</th>
</tr>
```

```

<tr>
 <td>future</td>
 <td>vision</td>
</tr>
<tr>
 <td>future</td>
 <td>vision</td>
</tr>
</table>
</body>
</html>

```

#### Output:



Notice that the table in the example above has double borders. This is because both the table and the `<th>` and `<td>` elements have separate borders.

### COLLAPSE TABLE BORDERS

The `border-collapse` property sets whether the table borders should be collapsed into a single border:

#### Example:

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
 border-collapse: collapse;
}

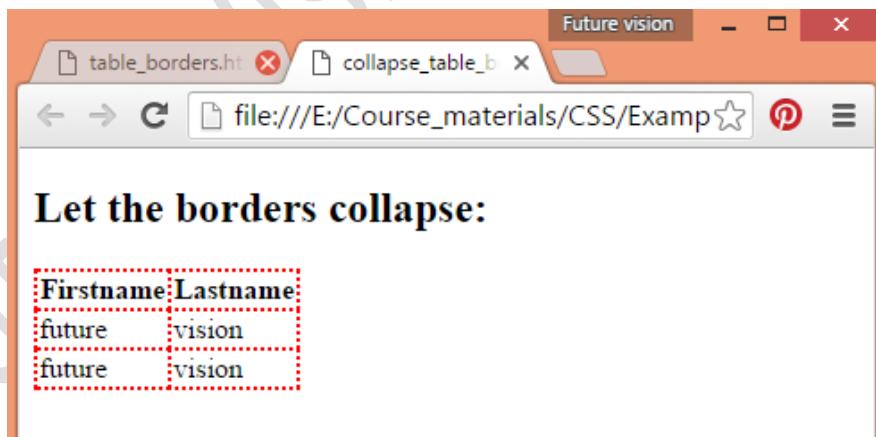
```

```

table, td, th {
 border: 2px dotted red;
}
</style>
</head>
<body>
<h2>Let the borders collapse:</h2>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>future</td>
<td>vision</td>
</tr>
<tr>
<td>future</td>
<td>vision</td>
</tr>
</table>
</body>
</html>

```

**Output:**



## TABLE WIDTH AND HEIGHT

Width and height of a table are defined by the `width` and `height` properties.

```

table {
 width: 90%;
}

```

```
}
```

```
th {
```

```
 height: 60px;
```

```
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 table, td, th {
 border: 2px solid red;
 }
 table {
 border-collapse: collapse;
 width: 90%;
 }
 th {
 height: 60px;
 }
</style>
</head>
<body>
<h2>The width and height Properties</h2>
<p>Set the width of the table, and the height of the table header row:</p>
<table>
 <tr>
 <th>Firstname</th>
 <th>Lastname</th>
 <th>Savings</th>
 </tr>
 <tr>
 <td>Siddharth</td>
 <td>parakh</td>
 <td>Rs.20,000</td>
 </tr>
 <tr>
 <td>Eswar</td>
 <td>Kumar</td>
 <td>Rs.15,000</td>
 </tr>
 <tr>
 <td>Future</td>
```

```

<td>vision</td>
<td>Rs.1,00,000</td>
</tr>
</table>
</body>
</html>

```

### Output:

The width and height Properties

Set the width of the table, and the height of the table header row:

<b>Firstname</b>	<b>Lastname</b>	<b>Savings</b>
Siddharth	parakh	Rs.20.000
Eswar	Kumar	Rs.15,000
Future	vision	Rs.1,00,000

### HORIZONTAL ALIGNMENT

The **text-align** property sets the horizontal alignment (like left, right, or center) of the content in **<th>** or **<td>**. By default, the content of **<th>** elements are center-aligned and the content of **<td>** elements are left-aligned.

```

th {
 text-align: center;
}

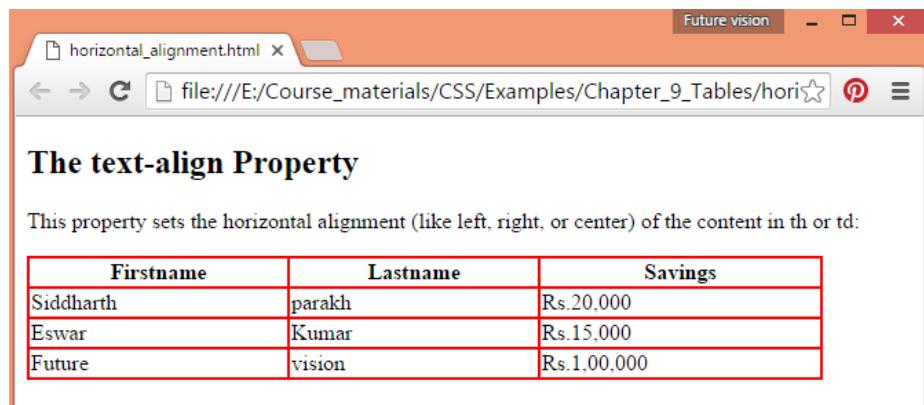
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 #table, td, th {
 border: 2px solid red;
 }
 table {
 border-collapse: collapse;
 width: 90%;
 }
 th {
 text-align: center;
 }
</style>
</head>
<body>
<h2>The text-align Property</h2>
<p>This property sets the horizontal alignment (like left, right, or center) of the content in th or td:</p>
<table>
 <tr>
 <th>Firstname</th>
 <th>Lastname</th>
 <th>Savings</th>
 </tr>
 <tr>
 <td>Siddharth</td>
 <td>parakh</td>
 <td>Rs.20,000</td>
 </tr>
 <tr>
 <td>Eswar</td>
 <td>Kumar</td>
 <td>Rs.15,000</td>
 </tr>
 <tr>
 <td>Future</td>
 <td>vision</td>
 <td>Rs.1,00,000</td>
 </tr>
</table>
</body>
```

```
</html>
```

## Output:



The screenshot shows a browser window titled "Future vision" displaying a table with horizontal alignment. The table has three columns: "Firstname", "Lastname", and "Savings". The rows contain the following data:

Firstname	Lastname	Savings
Siddharth	parakh	Rs.20,000
Eswar	Kumar	Rs.15,000
Future	vision	Rs.1,00,000

## VERTICAL ALIGNMENT

The **vertical-align** property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>. By default, the vertical alignment of the content in a table is middle (for both <th> and <td> elements).

```
td {
 height: 60px;
 vertical-align: top;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 table, td, th {
 border: 2px solid red;
 }
 table {
 border-collapse: collapse;
 width: 100%;
 }
 td {
 height: 40px;
 vertical-align: top;
 }
</style>
</head>
<body>
```

```
<h2>The vertical-align Property</h2>
<p>This property sets the vertical alignment (like top, bottom, or middle) of the content in th or td.</p>
<table>
 <tr>
 <th>Firstname</th>
 <th>Lastname</th>
 <th>Savings</th>
 </tr>
 <tr>
 <td>Siddharth</td>
 <td>Parakh</td>
 <td>Rs.20,000</td>
 </tr>
 <tr>
 <td>Eswar</td>
 <td>Kumar</td>
 <td>Rs.15,000</td>
 </tr>
 <tr>
 <td>Future</td>
 <td>vision</td>
 <td>Rs.1,00,000</td>
 </tr>
</table>
</body>
</html>
```

Output:

## TABLE PADDING

To control the space between the border and the content in a table, use the `padding` property on `<td>` and `<th>` elements:

```
th,
 padding: 10px;
 text-align: center;
}

td {
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 table, td, th {
 border: 2px solid red;
 text-align: center;
 }
 table {
 border-collapse: collapse;
 width: 100%;
 }
 th, td {
 padding: 10px;
 }
</style>
</head>
<body>
<h2>The padding Property</h2>
<p>This property adds space between the border and the content in a table.</p>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Siddharth</td>
<td>Parakh</td>
<td>Rs.20,000</td>
</tr>
<tr>
```

```

<td>Eswar</td>
<td>Kumar</td>
<td>Rs.15,000</td>
</tr>
<tr>
<td>Future</td>
<td>Vision</td>
<td>Rs.1,00,000</td>
</tr>
</table>
</body>
</html>

```

### Output:

Firstname	Lastname	Savings
Siddharth	Parakh	Rs.20.000
Eswar	Kumar	Rs.15,000
Future	vision	Rs.1,00,000

### HOVERABLE TABLE

Use the `:hover` selector on `<tr>` to highlight table rows on mouse over:

```
tr:hover {background-color: # b9afaf}
```

### Example:

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
 border-collapse: collapse;

```

```

 width: 100%;
 }
 th, td {
 padding: 10px;
 text-align: center;
 border-bottom: 2px dotted red;
 }
 tr:hover {
 background-color: #b9afaf
 }
</style>
</head>
<body>
<h2>Hoverable Table</h2>
<p>Move the mouse over the table rows to see the effect.</p>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Siddharth</td>
<td>Parakh</td>
<td>Rs.20,000</td>
</tr>
<tr>
<td>Eswar</td>
<td>Kumar</td>
<td>Rs.15,000</td>
</tr>
<tr>
<td>Future</td>
<td>vision</td>
<td>Rs.1,00,000</td>
</tr>
</table>
</body>
</html>

```

## Output:

The screenshot shows a web browser window titled "Future vision" displaying a table. The title bar also includes the file path "file:///E:/Course\_materials/CSS/Examples/Chapter\_9\_Tables/hov". The page content is titled "Hoverable Table" with the instruction "Move the mouse over the table rows to see the effect.". The table has three columns: "Firstname", "Lastname", and "Savings". The first row, containing "Siddharth", "Parakh", and "Rs.20,000", has a gray background color. The other two rows have white backgrounds. All rows are separated by dotted horizontal borders.

Firstname	Lastname	Savings
Siddharth	Parakh	Rs.20,000
Eswar	Kumar	Rs.15,000
Future	vision	Rs.1,00,000

## TABLE COLOR

This table color property specifies the background color and text color of <th> elements:

```
th {
 background-color: # 5a52f4;
 color: black;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 table {
 border-collapse: collapse;
 width: 100%;
 }

 th, td {
 text-align: center;
 padding: 10px;
 }

 tr:nth-child(even){background-color: #5a52f4}

 th {
 background-color: #f72f5e;
```

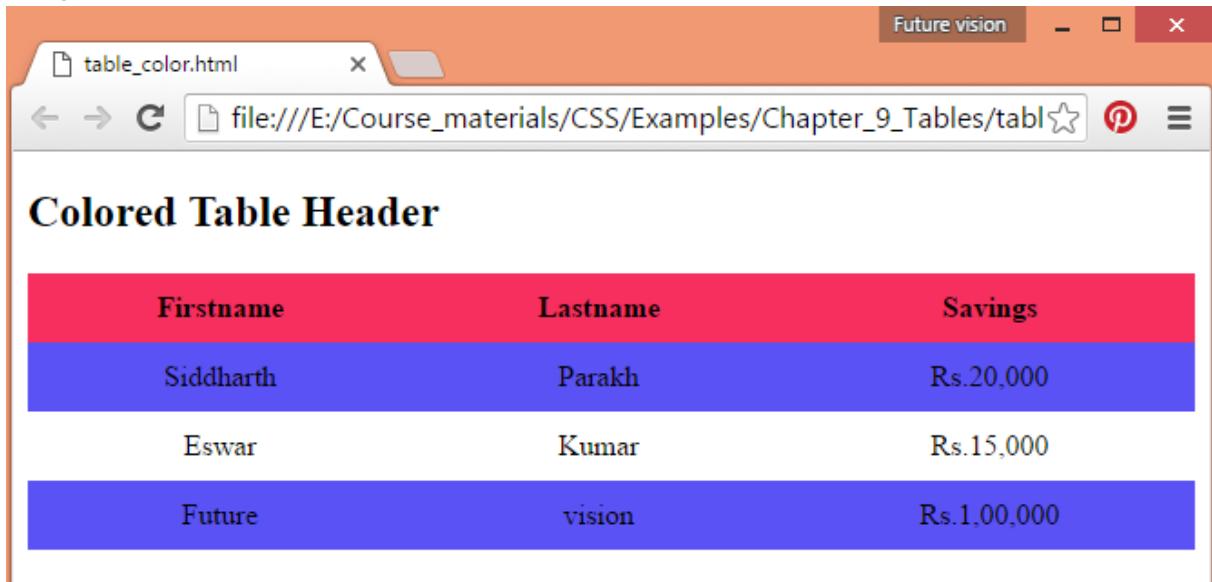
```
 color: black;
 }
</style>
</head>
<body>

<h2>Colored Table Header</h2>

<table>
<tr>
 <th>Firstname</th>
 <th>Lastname</th>
 <th>Savings</th>
</tr>
<tr>
 <td>Siddharth</td>
 <td>Parakh</td>
 <td>Rs.20,000</td>
</tr>
<tr>
 <td>Eswar</td>
 <td>Kumar</td>
 <td>Rs.15,000</td>
</tr>
<tr>
 <td>Future</td>
 <td>vision</td>
 <td>Rs.1,00,000</td>
</tr>
</table>

</body>
</html>
```

## Output:



The screenshot shows a web browser window titled "Future vision" displaying a table. The browser's address bar shows the URL "file:///E:/Course\_materials/CSS/Examples/Chapter\_9\_Tables/table\_color.html". The table has a red header row with columns for "Firstname", "Lastname", and "Savings". Below it are three blue rows containing data: "Siddharth Parakh Rs.20,000", "Eswar Kumar Rs.15,000", and "Future vision Rs.1,00,000".

Firstname	Lastname	Savings
Siddharth	Parakh	Rs.20,000
Eswar	Kumar	Rs.15,000
Future	vision	Rs.1,00,000

## RESPONSIVE TABLE

Add a container element (like <div>) with `overflow-x:auto` around the <table> element to make it responsive:

```
<div style="overflow-x:auto;">
 <table>
 ... table content ...
 </table>
</div>
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
 border-collapse: collapse;
 width: 100%;
}
th, td {
 text-align: center;
 padding: 10px;
}
tr:nth-child(even) {
 background-color: #d9d7d8
}
```

```
 }
 </style>
</head>
<body>
<h2>Responsive Table</h2>
<div style="overflow-x:auto;">
 <table>
 <tr>
 <th>First Name</th>
 <th>Last Name</th>
 <th>Points</th>
 <th>Points</th>
 <th>Points</th>
 <th>Points</th>
 <th>Points</th>
 <th>Points</th>
 <th>Points</th>
 </tr>
 <tr>
 <td>Nishi</td>
 <td>agarwal</td>
 <td>60</td>
 <td>60</td>
 <td>60</td>
 <td>60</td>
 <td>60</td>
 <td>60</td>
 <td>60</td>
 </tr>
 <tr>
 <td>Dolly</td>
 <td>agarwal</td>
 <td>94</td>
 <td>94</td>
 <td>94</td>
 <td>94</td>
 <td>94</td>
 <td>94</td>
 <td>94</td>
 </tr>
 <tr>
 <td>Minakshi</td>
```

```
<td>agarwal</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
</tr>
</table>
</div>
</body>
</html>
```

### Output:

The screenshot shows a web browser window titled "Future vision" with three tabs open: "table\_color.html", "responsive\_table.html", and "responsive\_table.html". The active tab displays a responsive table with the following data:

First Name	Last Name	Points							
Nishi	agarwal	60	60	60	60	60	60	60	60
Dolly	agarwal	94	94	94	94	94	94	94	94
Minakshi	agarwal	67	67	67	67	67	67	67	67

## CHAPTER 10 - CSS COLORS

CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or for the background of the element. They can also be used to affect the color of borders and other decorative effects.

Colors in CSS can be specified by the following methods:

- ✓ RGB colors
- ✓ RGBA colors
- ✓ HSL colors
- ✓ HSLA colors
- ✓ Hexadecimal colors
- ✓ Predefined/Cross-browser color names

### RGB VALUES

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

```
#p1 {background-color: rgb(255, 0, 0);} /* red */
#p2 {background-color: rgb(255,255,0);} /* yellow */
#p3 {background-color:rgb(255,0,255);} /* cerise */
```

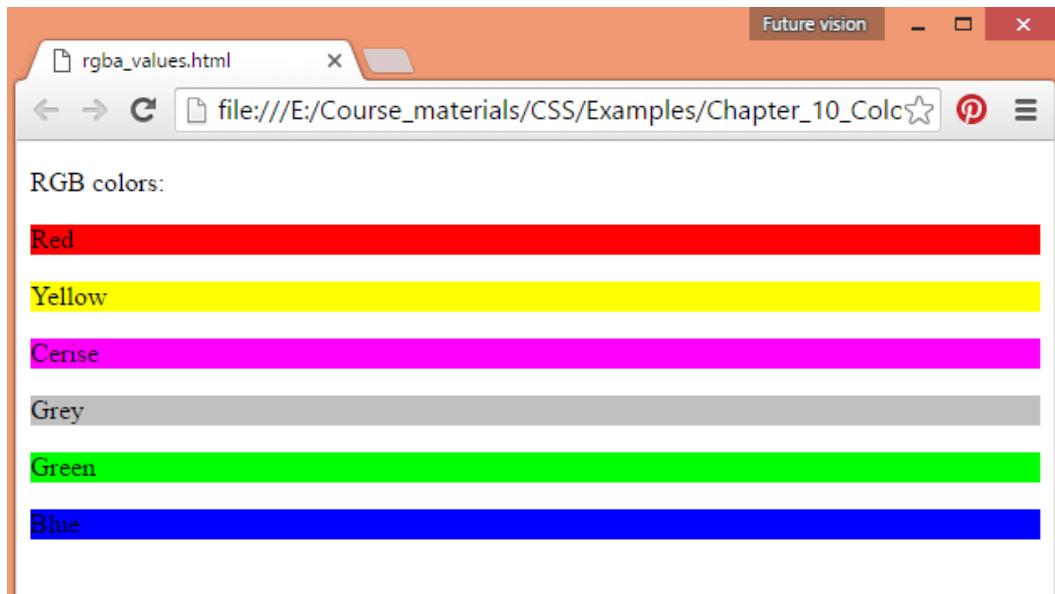
**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
 #p1 {background-color:rgb(255,0,0);}
 #p2 {background-color:rgb(255,255,0);}
 #p3 {background-color:rgb(255,0,255);}
 #p4 {background-color:rgb(192,192,192);}
 #p5 {background-color:rgb(0,255,0);}
 #p6 {background-color:rgb(0,0,255);}
</style>
</head>

<body>
<p>RGB colors:</p>
<p id="p1">Red</p>
<p id="p2"> Yellow </p>
<p id="p3"> Cerise </p>
```

```
<p id="p4">Grey</p>
<p id="p5"> Green</p>
<p id="p6">Blue </p>
</body>
</html>
```

#### Output:



## RGBA COLORS

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity of the object.

An RGBA color value is specified with: `rgba(red, green, blue, alpha)`. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

```
#p1 {background-color: rgba(255, 0, 0, 0.3);} /* red with opacity */
#p2 {background-color: rgba(0, 255, 0, 0.3);} /* yellow with opacity */
#p3 {background-color: rgba(0, 0, 255, 0.3);} /* cerise with opacity */
```

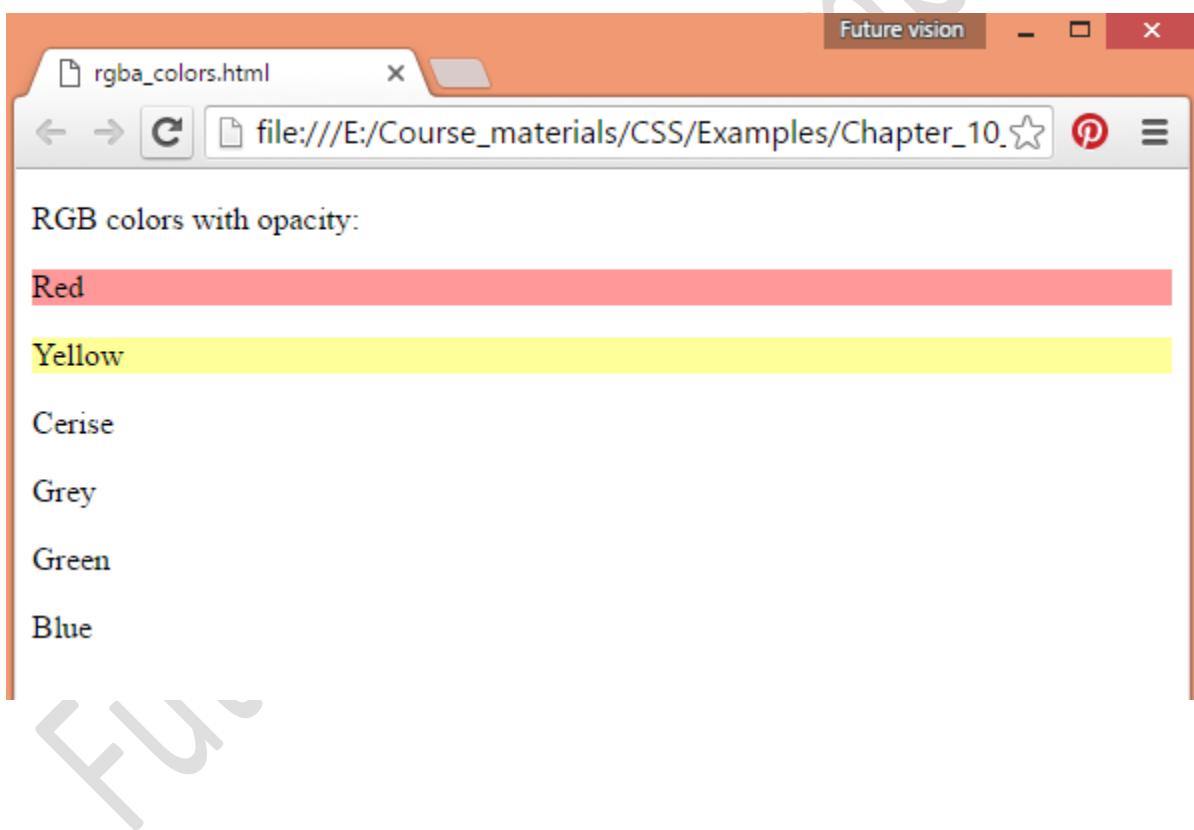
#### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 #p1 {background-color:rgb(255,0,0);}
 #p2 {background-color:rgb(255,255,0);}
 #p3 {background-color:rgb(255,0,255);}
 #p4 {background-color:rgb(192,192,192);}
 #p5 {background-color:rgb(0,255,0);}

```

```
#p6 {background-color:rgb(0,0,255);}
</style>
</head>
<body>
 <p>RGB colors:</p>
 <p id="p1">Red</p>
 <p id="p2"> Yellow </p>
 <p id="p3"> Cerise </p>
 <p id="p4">Grey</p>
 <p id="p5"> Green</p>
 <p id="p6">Blue </p>
</body>
</html>
```

### Output:



## HSL COLORS

HSL stands for hue, saturation, and lightness - and represents a cylindrical-coordinate representation of colors.

An HSL color value is specified with: hsl(hue, saturation, lightness).

Hue is a degree on the color wheel (from 0 to 360) - 0 (or 360) is red, 120 is green, 240 is blue. Saturation is a percentage value; 0% means a shade of gray and 100% is the full color. Lightness is also a percentage; 0% is black, 100% is white.

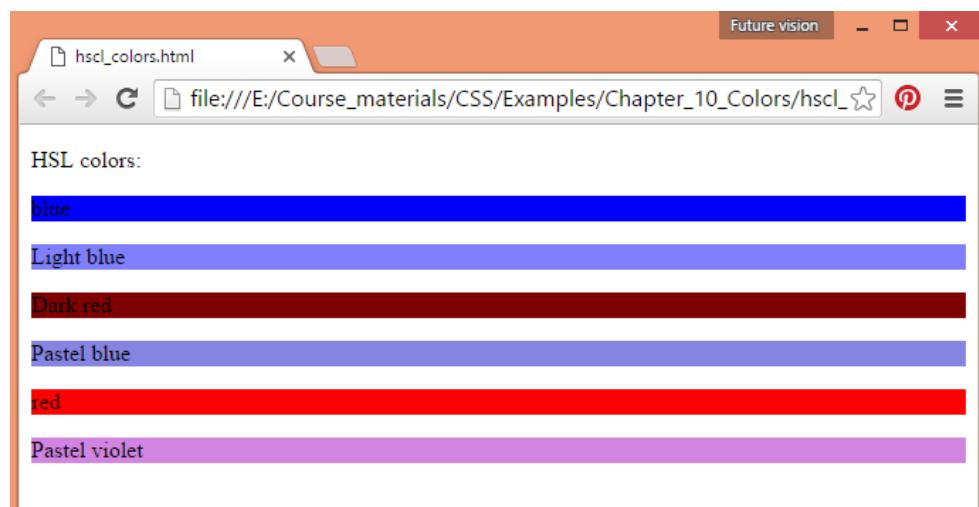
```
#p1 {background-color: hsl(240, 100%, 50%); /* blue */
#p2 {background-color: hsl(240, 100%, 75%); /* light blue */
#p3 {background-color: hsl(240, 100%, 25%); /* dark blue */
#p4 {background-color: hsl(240, 60%, 70%); /* pastel blue */
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
 table {
 border-collapse: collapse;
 }
 table, td, th {
 border: 2px dotted red;
 }
</style>
</head>
<body>
<h2>Let the borders collapse:</h2>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>future</td>
<td>vision</td>
</tr>
<tr>
<td>future</td>
<td>vision</td>
</tr>
</table>
</body>
```

```
</html>
```

## Output:



## HSLA COLORS

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity of the object.

An HSLA color value is specified with: hsla(hue, saturation, lightness, alpha), where the alpha parameter defines the opacity. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

```
#p1 {background-color: hsla(240, 100%, 50%, 0.4); /* blue */
#p2 {background-color: hsla(240, 100%, 75%, 0.4); /* light blue */
#p3 {background-color: hsla(240, 100%, 25%, 0.4); /* dark blue */
#p4 {background-color: hsla(240, 60%, 70%, 0.4); /* pastel blue */
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 #p1 {background-color:hsla(240,100%,50%, 0.4);}
 #p2 {background-color:hsla(240,100%,75%, 0.4);}
 #p3 {background-color:hsla(360,100%,25%, 0.4);}
 #p4 {background-color:hsla(240,60%,70%, 0.4);}
 #p5 {background-color:hsla(360,100%,50%, 0.4);}
 #p6 {background-color:hsla(290,60%,70%, 0.4);}
</style>
</head>
<body>
```

```
<p>HSLA colors with opacity:</p>
<p id="p1">blue</p>
<p id="p2">Light blue</p>
<p id="p3">Dark red</p>
<p id="p4">Pastel blue</p>
<p id="p5">red</p>
<p id="p6">Pastel violet</p>
</body>
</html>
```

### Output:



## CHAPTER 11- CSS MARGINS

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

With CSS, you have full control over the margins. There are CSS properties for setting the margin for each side of an element (top, right, bottom, and left).

### INDIVIDUAL SIDES

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 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

```
p {
 margin-top: 90px;
 margin-bottom: 90px;
 margin-right: 140px;
 margin-left: 70px;
}
```

#### **Example:**

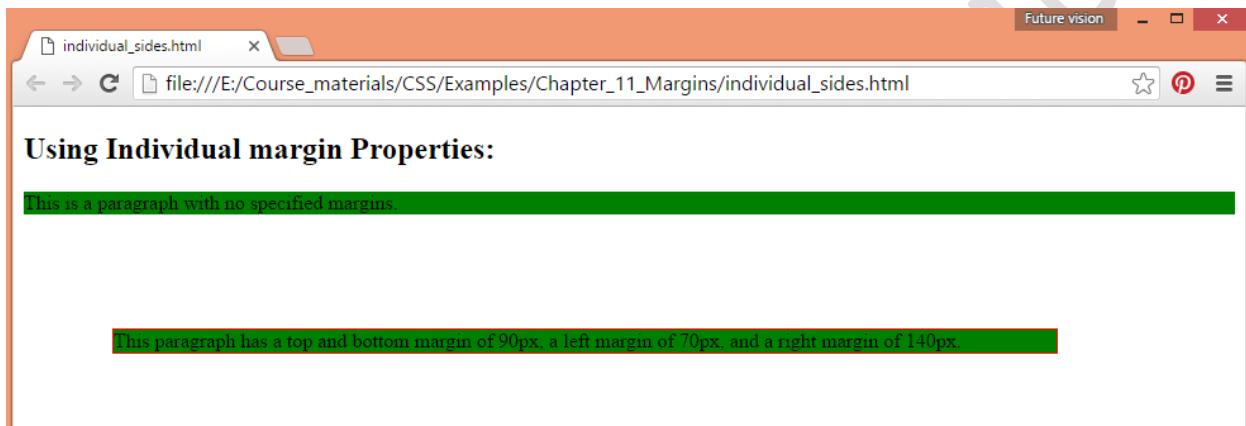
```
<!DOCTYPE html>
<html>
<head>
<style>
p#green {
 background-color: green;
}
p.ex {
 border: 1px solid red;
 margin-top: 90px;
 margin-bottom: 90px;
 margin-right: 140px;
 margin-left: 70px;
}
```

```

</style>
</head>
<body>
<h2>Using Individual margin Properties:</h2>
<p id="green">This is a paragraph with no specified margins.</p>
<p class="ex" id="green">This paragraph has a top and bottom margin of 90px, a left
margin of 70px, and a right margin of 140px.</p>
</body>
</html>

```

### Output:



## SHORTHAND PROPERTY

To shorten the code, it is possible to specify all the margin properties in one property. The `margin` property is a shorthand property for the following individual margin properties:

```

margin-top
margin-right
margin-bottom
margin-left
p{
 margin: 90px 140px 90px 70px;
}

```

### Example:

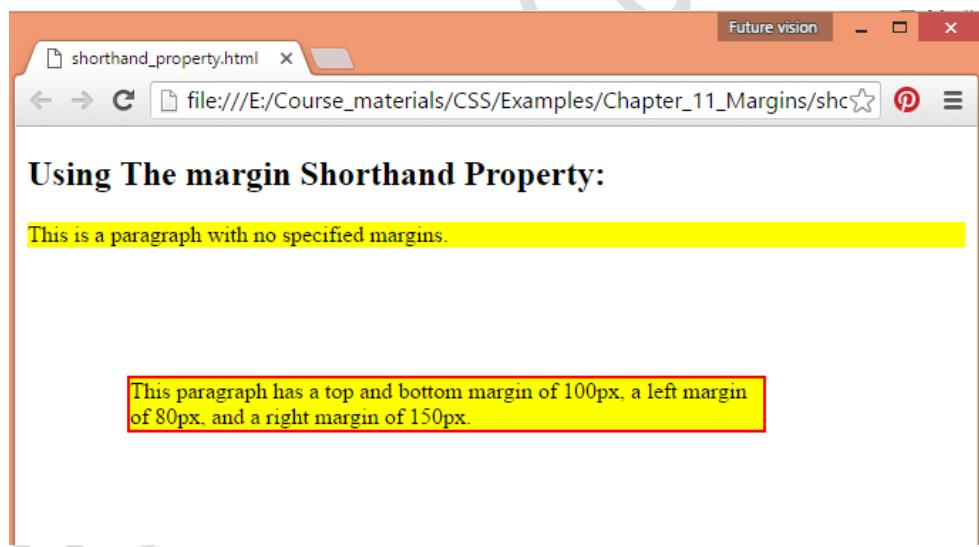
```

<!DOCTYPE html>
<html>

```

```
<head>
<style>
 p#yellow{
 background-color: yellow;
 }
 p.ex {
 border: 2px solid red;
 margin: 90px 140px 90px 70px;
 }
</style>
</head>
<body>
<h2>Using The margin Shorthand Property:</h2>
<p id="yellow">This is a paragraph with no specified margins.</p>
<p class="ex" id="yellow">This paragraph has a top and bottom margin of 100px, a left margin of 80px, and a right margin of 150px.</p>
</body>
</html>
```

#### Output:



If the **margin** property has four values:

**margin: 25px 50px 75px 100px;**

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

If the **margin** property has three values:

**margin: 25px 50px 75px;**

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

If the **margin** property has two values:

**margin: 25px 50px;**

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

If the **margin** property has one value:

**margin: 25px;**

- ✓ all four margins are 25px

## USE OF THE AUTO VALUE

You can set the margin property to **auto** to horizontally center the element within its container.

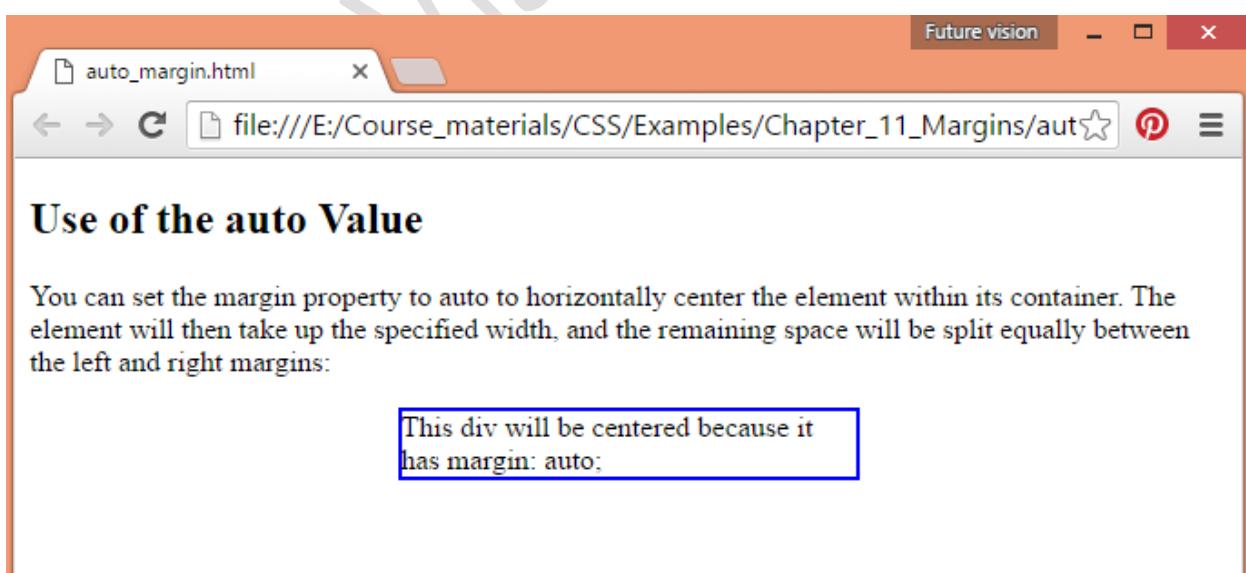
The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:

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

## Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 div {
 width:250px;
 margin: auto;
 border: 2px solid blue;
 }
</style>
</head>
<body>
<h2>Use of the auto Value</h2>
<p>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:</p>
<div>
 This div will be centered because it has margin: auto;
</div>
</body>
</html>
```

## Output:



## CHAPTER 12 – CSS PADDINGS

The **padding** property allows you to specify how much space should appear between the content of an element and its border –

The value of this attribute should be either a length, a percentage, or the **word *inherit***. If the value is *inherit*, it will have the same padding as its parent element. If a percentage is used, the percentage is of the containing box.

The following CSS properties can be used to control lists. You can also set different values for the padding on each side of the box using the following properties –

- ✓ The **padding-bottom** specifies the bottom padding of an element.
- ✓ The **padding-top** specifies the top padding of an element.
- ✓ The **padding-left** specifies the left padding of an element.
- ✓ The **padding-right** specifies the right padding of an element.
- ✓ The **padding** serves as shorthand for the preceding properties.

### PADDING - INDIVIDUAL SIDES

CSS has properties for specifying the padding for each side of an element:

- ✓ **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

```
p {
 padding-top: 60px;
 padding-right: 40px;
 padding-bottom: 60px;
 padding-left: 70px;
}
```

#### **Example:**

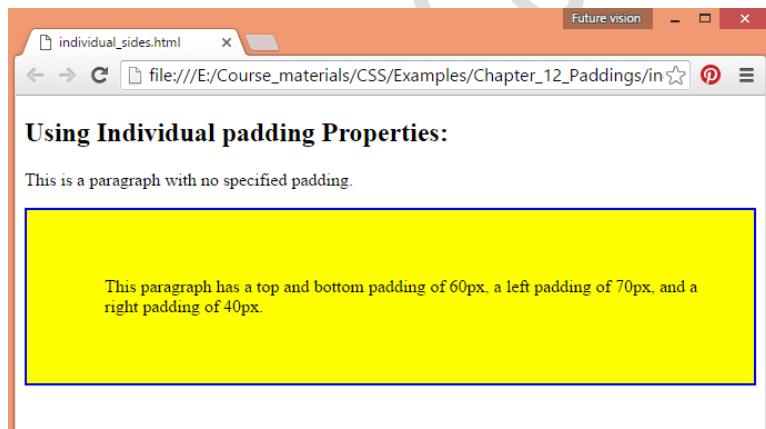
```
<!DOCTYPE html>
<html>
<head>
```

```

<style>
 p.one {
 border: 2px solid blue;
 background-color: yellow;
 padding-top: 60px;
 padding-right: 40px;
 padding-bottom: 60px;
 padding-left: 70px;
 }
</style>
</head>
<body>
<h2>Using Individual padding Properties:</h2>
<p>This is a paragraph with no specified padding.</p>
<p class="one">This paragraph has a top and bottom padding of 60px, a left padding of 70px, and a right padding of 40px.</p>
</body>
</html>

```

#### **Output:**



## **SHORTHAND PROPERTY**

To shorten the code, it is possible to specify all the padding properties in one property. The **padding** property is a shorthand property for the following individual padding properties:

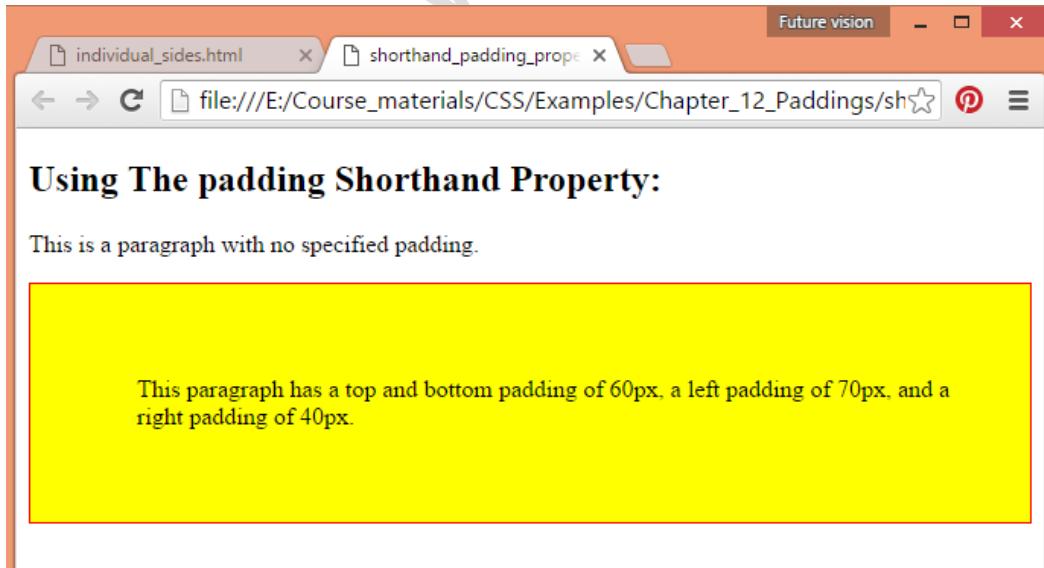
- ✓ **padding-top**
- ✓ **padding-right**
- ✓ **padding-bottom**
- ✓ **padding-left**

```
p {
 padding: 60px 40px 60px 70px;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
 border: 1px solid red;
 background-color: yellow;
 padding: 60px 40px 60px 70px;
}
</style>
</head>
<body>
<h2>Using The padding Shorthand Property:</h2>
<p>This is a paragraph with no specified padding.</p>
<p class="one">This paragraph has a top and bottom padding of 60px, a left padding of
70px, and a right padding of 40px.</p>
</body>
</html>
```

### Output:



If the **padding** property has four values:

- ✓ **padding: 25px 50px 75px 100px;**
  - o top padding is 25px
  - o right padding is 50px
  - o bottom padding is 75px
  - o left padding is 100px
- ✓ If the **padding** property has three values:
- ✓ **padding: 25px 50px 75px;**
  - o top padding is 25px
  - o right and left paddings are 50px
  - o bottom padding is 75px
- ✓ If the **padding** property has two values:
- ✓ **padding: 25px 50px;**
  - o top and bottom paddings are 25px
  - o right and left paddings are 50px
- ✓ If the **padding** property has one value:
- ✓ **padding: 25px;**
  - o all four paddings are 25px

## CHAPTER 13- CSS OUTLINES

An outline is a line drawn around an element - outside the border. This can be used to make an element "stand out". The CSS outline properties specify the style, color, and width of an outline.

**Outlines are very similar to borders, but there are few major differences as well:**

- ✓ An outline does not take up space.
- ✓ Outlines do not have to be rectangular.
- ✓ Outline is always the same on all sides; you cannot specify different values for different sides of an element.

**You can set the following outline properties using CSS.**

- ✓ The **outline-width** property is used to set the width of the outline.
- ✓ The **outline-style** property is used to set the line style for the outline.
- ✓ The **outline-color** property is used to set the color of the outline.
- ✓ The **outline** property is used to set all the above three properties in a single statement.

### OUTLINE STYLE

The **outline-style** property specifies the style for the line (solid, dotted, or dashed) that goes around an element.

The **outline-style** property can have one of the following values:

- ✓ **dotted** - Defines a dotted outline
- ✓ **dashed** - Defines a dashed outline
- ✓ **solid** - Defines a solid outline
- ✓ **double** - Defines a double outline
- ✓ **groove** - Defines a 3D grooved outline. The effect depends on the outline-color value
- ✓ **ridge** - Defines a 3D ridged outline. The effect depends on the outline-color value
- ✓ **inset** - Defines a 3D inset outline. The effect depends on the outline-color value
- ✓ **outset** - Defines a 3D outset outline. The effect depends on the outline-color value
- ✓ **none** - Defines no outline
- ✓ **hidden** - Defines a hidden outline

```
#one {
 border: 2px solid black;
 outline-color: blue;
}
```

```
p.dotted {outline-style: dotted;}
p.dashed {outline-style: dashed;}
p.solid {outline-style: solid;}
p.double {outline-style: double;}
```

```
p.groove {outline-style: groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
```

### Example:

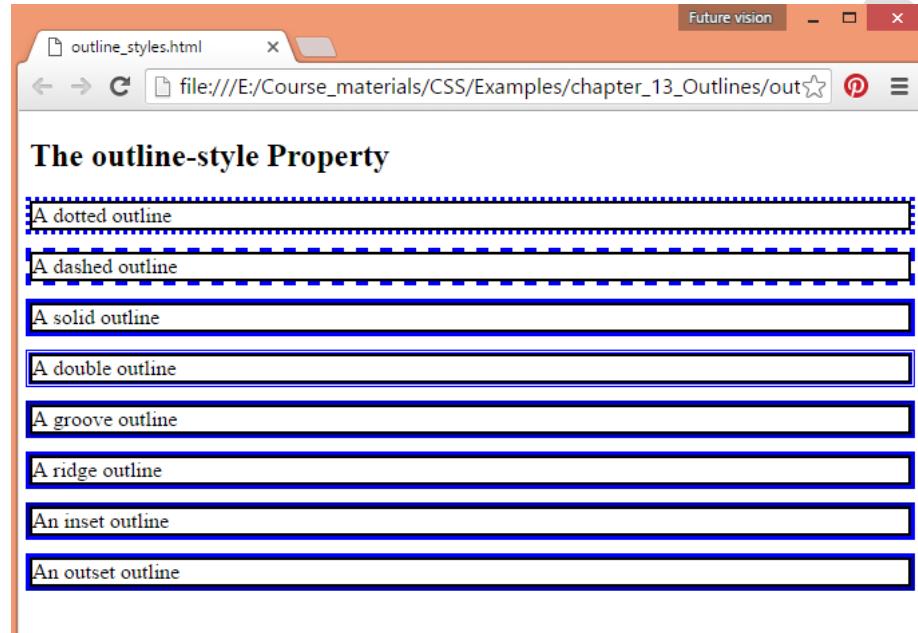
```
<!DOCTYPE html>
<html>
<head>
<style>
 #one {
 border: 2px solid black;
 outline-color: blue;
 }
 p.dotted {
 outline-style: dotted;
 }
 p.dashed {
 outline-style: dashed;
 }
 p.solid {
 outline-style: solid;
 }
 p.double {
 outline-style: double;
 }
 p.groove {
 outline-style: groove;
 }
 p.ridge {
 outline-style: ridge;
 }
 p.inset {
 outline-style: inset;
 }
 p.outset {
 outline-style: outset;
 }
</style>
</head>
<body>
<h2>The outline-style Property</h2>
<p id="one" class="dotted">A dotted outline</p>
```

```

<p id="one" class="dashed">A dashed outline</p>
<p id="one" class="solid">A solid outline</p>
<p id="one" class="double">A double outline</p>
<p id="one" class="groove">A groove outline</p>
<p id="one" class="ridge">A ridge outline</p>
<p id="one" class="inset">An inset outline</p>
<p id="one" class="outset">An outset outline</p>
</body>
</html>

```

## Output:



## OUTLINE COLOR

The **outline-color** property is used to set the color of the outline.

The color can be set by:

- ✓ name - specify a color name, like "red"
- ✓ RGB - specify a RGB value, like "rgb(255,0,0)"
- ✓ Hex - specify a hex value, like "#ff0000"
- ✓ invert - performs a color inversion (which ensures that the outline is visible, regardless of color background)

```

#two {
 border: 2px solid black;
 outline-style: double;
}

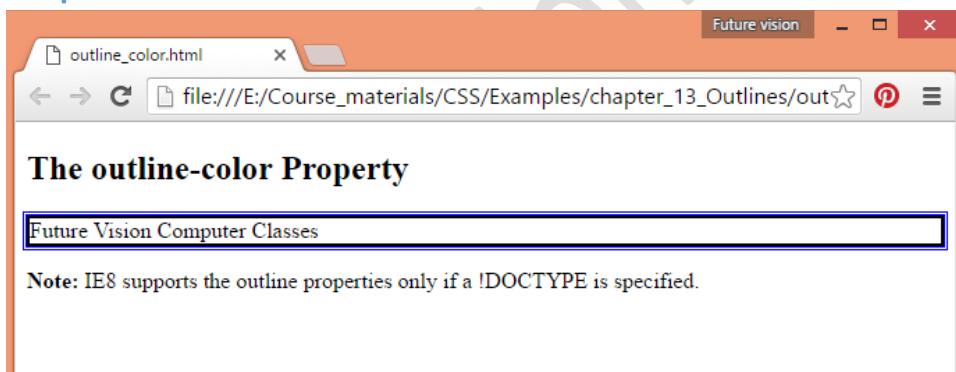
```

```
 outline-color: blue;
 }
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
#two {
 border: 2px solid black;
 outline-style: double;
 outline-color: blue;
}
</style>
</head>
<body>
<h2>The outline-color Property</h2>
<p id="two">Future Vision Computer Classes</p>
<p>Note: IE8 supports the outline properties only if a !DOCTYPE is specified.</p>
</body>
</html>
```

### Output:



## OUTLINE WIDTH

The outline-width property specifies the width of the outline to be added to the box. Its value should be a length or one of the values thin, medium, or thick, just like the border-width attribute.

```
#four {
 border: 2px solid black;
}

p.one {
```

```
outline-style: double;
outline-color: red;
outline-width: thick;
}

p.two {
```

```
outline-style: double;
outline-color: green;
outline-width: 4px;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 #four {border: 2px solid black;}

 p.one {
 outline-style: double;
 outline-color: red;
 outline-width: thick;
 }

 p.two {
 outline-style: double;
 outline-color: green;
 outline-width: 4px;
 }
</style>
</head>
<body>

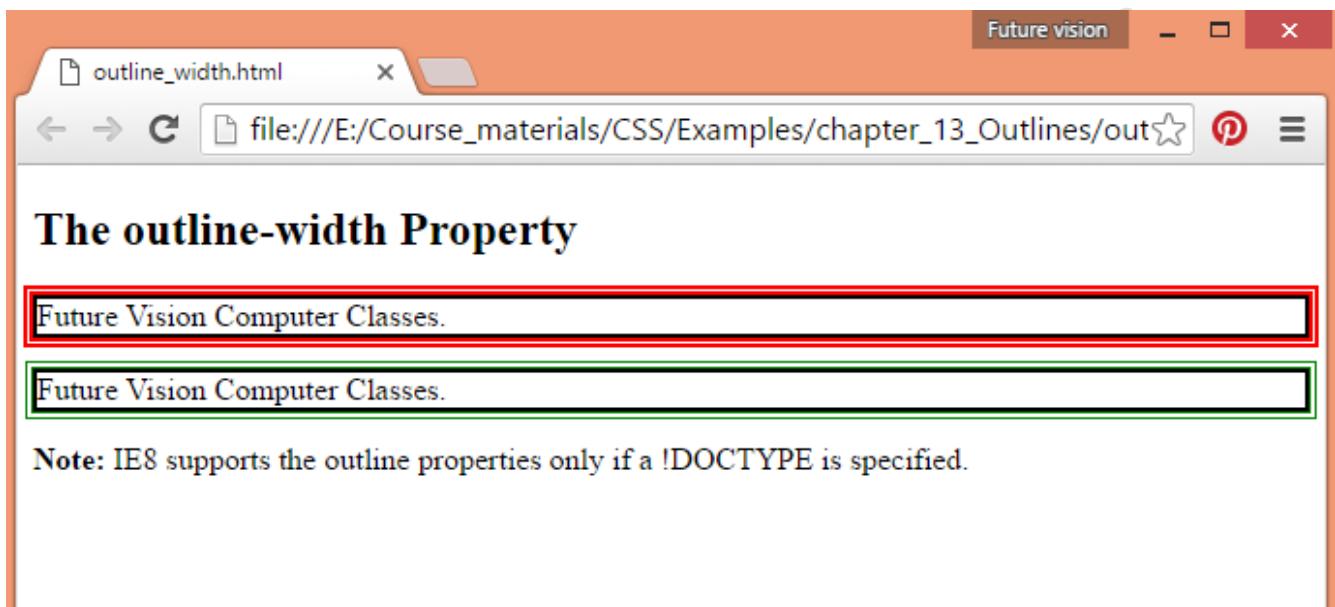
<h2>The outline-width Property</h2>

<p id="four" class="one"> Future Vision Computer Classes.</p>
<p id="four" class="two"> Future Vision Computer Classes.</p>
```

**Note:** IE8 supports the outline properties only if a !DOCTYPE is specified.

```
</body>
</html>
```

**Output:**



## OUTLINE - SHORTHAND PROPERTY

To shorten the code, it is also possible to specify all the individual outline properties in one property.

The **outline** property is a shorthand property for the following individual outline properties:

- ✓ **outline-width**
- ✓ **outline-style** (required)
- ✓ **outline-color**

```
#five {
 border: 2px solid black;
 outline: 6px dotted green;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
#five {
 border: 2px solid black;
 outline: 6px dotted green;
}
</style>
</head>
<body>

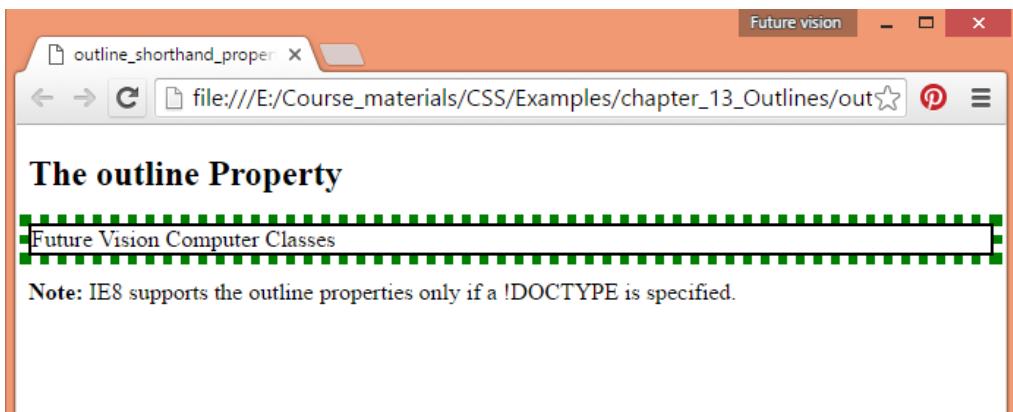
<h2>The outline Property</h2>

<p id="five">Future Vision Computer Classes</p>

<p>Note: IE8 supports the outline properties only if a !DOCTYPE is specified.</p>

</body>
</html>
```

**Output:**



**Extra Practice**



FutureVisionComputerClasses

## CHAPTER 14 - CSS LISTS

Lists are very helpful in conveying a set of either numbered or bulleted points.

This chapter teaches you how to control list type, position, style, etc., using CSS. The CSS list properties allow you to:

- ✓ Set different list item markers for ordered lists
- ✓ Set different list item markers for unordered lists
- ✓ Set an image as the list item marker
- ✓ Add background colors to lists and list items

### DIFFERENT LIST ITEM MARKERS

The ***list-style-type*** property allows you to control the shape or style of a bullet point (also known as a marker) in case of unordered lists and the style of numbering characters in ordered lists.

```
ul.a {
 list-style-type: circle;
}

ul.b {
 list-style-type: square;
}

ol.c {
 list-style-type: upper-roman;
}

ol.d {
 list-style-type: lower-alpha;
}
```

#### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 Ul#a {
 list-style-type: square;
 }
 Ul#b {
 list-style-type: circle;
 }
 ol.c {
```

```

 list-style-type: lower-alpha;
 }
 ol.d {
 list-style-type: upper-roman;
 }
</style>
</head>
<body>
<p>Example of unordered lists:</p>
<ul id="a">
 Photoshop
 CorelDraw
 Web Designing

<ul id="b">
 C/C++
 Java/Javascript
 PHP/MYSQL

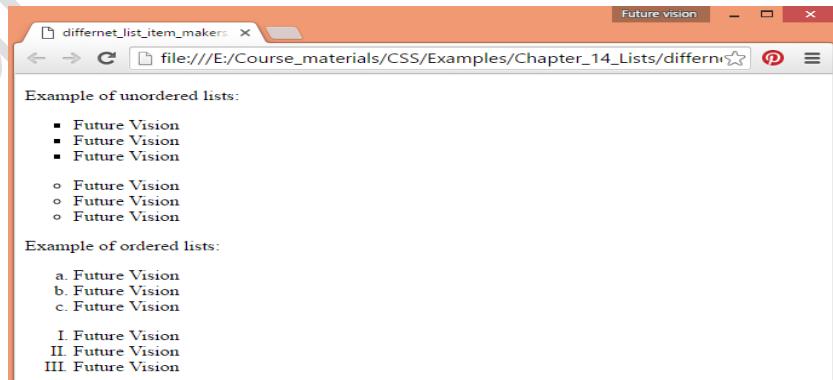
<p>Example of ordered lists:</p>
<ol class="c">
 Jquery
 Bootstrap
 Wordpress

<ol class="d">
 Web Development
 Digital Marketing
 Live Project Training

</body>
</html>

```

## Output:



## AN IMAGE AS THE LIST ITEM MARKER

The ***list-style-image*** allows you to specify an image so that you can use your own bullet style. The syntax is similar to the background-image property with the letters url starting the value of the property followed by the URL in brackets. If it does not find the given image then default bullets are used.

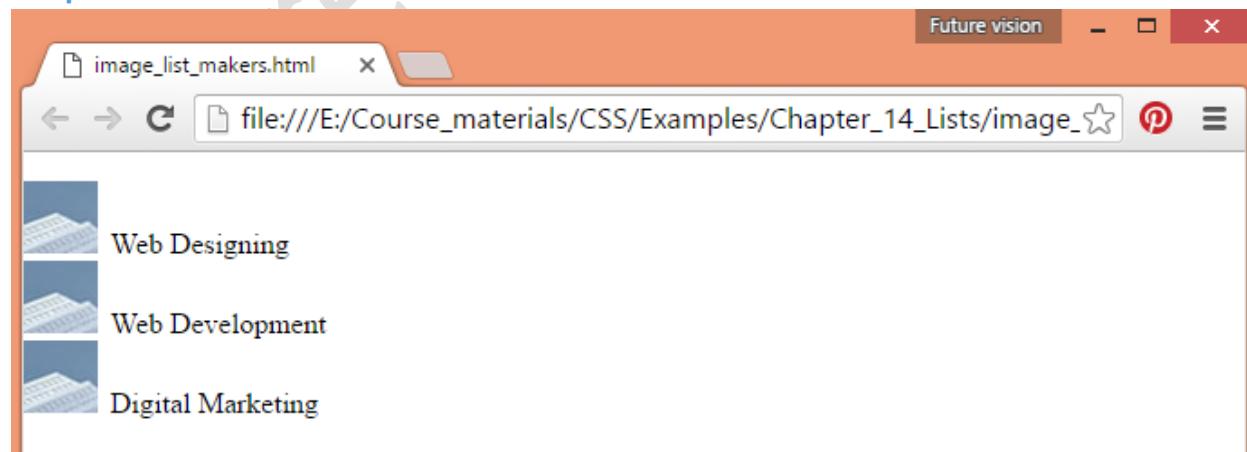
```
Ul#square {
 list-style-image: url('square.png');
}
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
 Ul#square {
 list-style-image: url('../images/list_img.png');
 }
</style>
</head>
<body>
<ul id="square">
 Web Designing
 Web Development
 Digital Marketing

</body>
</html>
```

**Output:**



## LIST - SHORTHAND PROPERTY

The **list-style** property is a shorthand property. It is used to set all the list properties in one declaration:

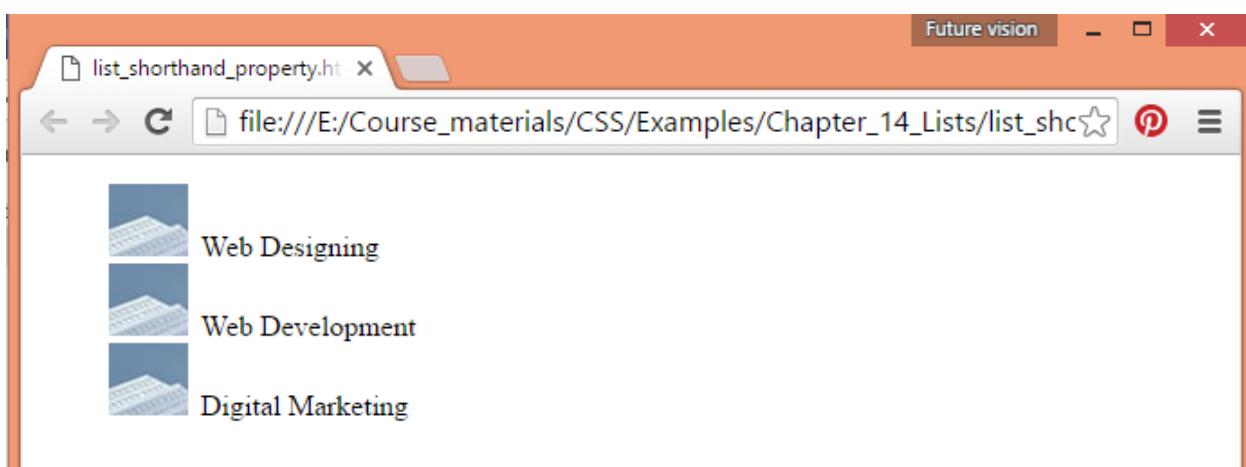
```
ul#square {
 list-style: square inside url('square.png');
}
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
 ul#square {
 list-style: square inside url("../images/list_img.png");
 }
</style>
</head>
<body>
<ul id="square">
 Web Designing
 Web Development
 Digital Marketing

</body>
</html>
```

**Output:**



When using the shorthand property, the order of the property values are:

- ✓ **list-style-type** (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)
- ✓ **list-style-position** (specifies whether the list-item markers should appear inside or outside the content flow)
- ✓ **list-style-image** (specifies an image as the list item marker)

If one of the property values above are missing, the default value for the missing property will be inserted, if any.

## POSITION THE LIST ITEM MARKERS

The **list-style-position** property specifies whether the list-item markers should appear inside or outside the content flow:

```
ul {
 list-style-position: inside;
}
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
 ul.inside {list-style-position:inside;}
 ul.outside {list-style-position:outside;}
</style>
</head>
<body>
 <p>The following list has list-style-position: inside:</p>
 <ul class="inside">
 Web Designing
 Web Development
 Digital Marketing

 <p>The following list has list-style-position: outside:</p>
 <ul class="outside">
 Web Designing
 Web Development
 Digital Marketing

 <p>"list-style-position: outside" is the default setting.</p>
</body>
</html>
```

## Output:

The following list has list-style-position: inside:

- Web Designing
- Web Development
- Digital Marketing

The following list has list-style-position: outside:

- Web Designing
- Web Development
- Digital Marketing

"list-style-position: outside" is the default setting.

## STYLING LIST WITH COLORS

We can also style lists with colors, to make them look a little more interesting.

Anything added to the <ol> or <ul> tag, affects the entire list, while properties added to the <li> tag will affect the individual list items:

```
ol {
 background: #ff9999;
 padding: 20px;
}

ul {
 background: #3399ff;
 padding: 20px;
}

ol li {
 background: #ffe5e5;
 padding: 5px;
 margin-left: 35px;
}

ul li {
```

```
background: #cce5ff;
margin: 5px;
}
```

**Example:**

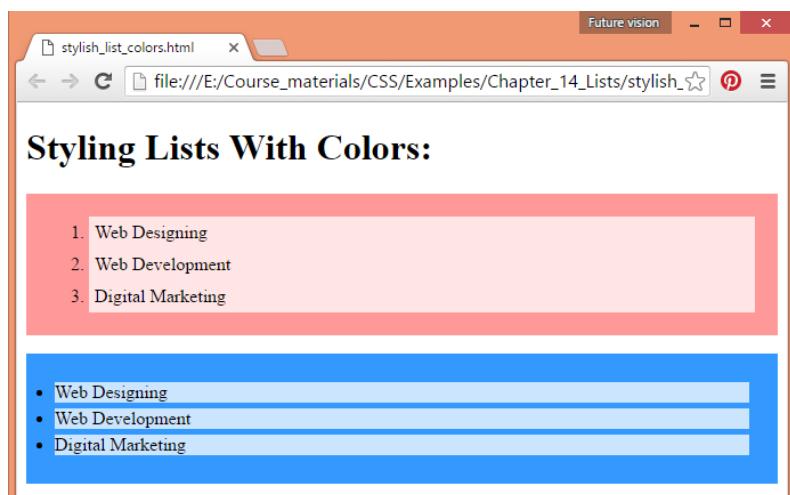
```
<!DOCTYPE html>
<html>
<head>
<style>
 ol {
 background: #ff9999;
 padding: 20px;
 }
 ul {
 background: #3399ff;
 padding: 20px;
 }
 ol li {
 background: #ffe5e5;
 padding: 5px;
 margin-left: 35px;
 }
 ul li {
 background: #cce5ff;
 margin: 5px;
 }
</style>
</head>
<body>
 <h1>Styling Lists With Colors:</h1>

 Web Designing
 Web Development
 Digital Marketing

 Web Designing
 Web Development
 Digital Marketing

</body>
</html>
```

**Output:**



# CHAPTER 15 - CSS DROPSHADOWS

## BASIC DROPSHADOW

Create a dropdown box that appears when the user moves the mouse over an element.

```
<style>
 .dropdown {
 position: fixed;
 display: inline-block;
 }

 .dropdown-content {
 display: none;
 position: absolute;
 background-color: #bdb6b6;
 min-width: 150px;
 box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
 padding: 10px 15px;
 z-index: 1;
 }

 .dropdown:hover .dropdown-content {
 display: block;
 }
</style>

<div class="dropdown">
 Mouse over me
 <div class="dropdown-content">
 <p>Hello World!</p>
 </div>
</div>
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
 .dropdown {
 position: fixed;
 display: inline-block;
 }

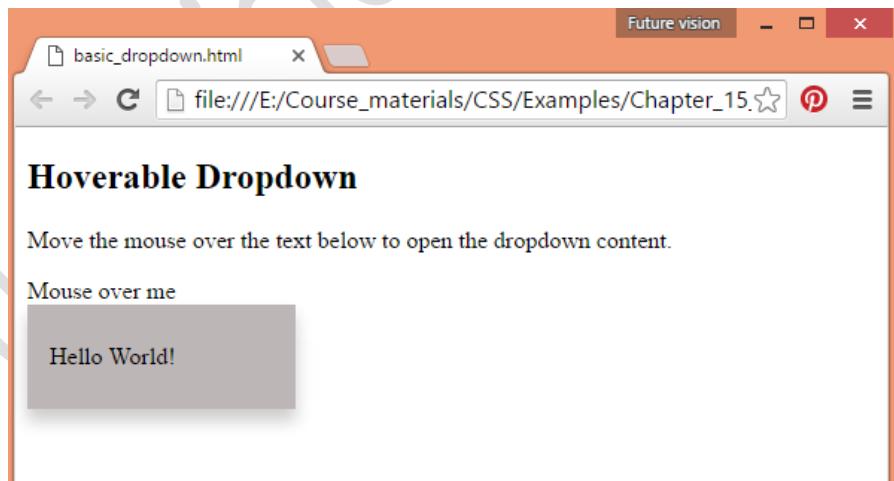
 .dropdown-content {
```

```

 display: none;
 position: absolute;
 background-color: #bdb6b6;
 min-width: 150px;
 box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
 padding: 10px 15px;
 }
.dropdown:hover .dropdown-content {
 display: block;
}
</style>
</head>
<body>
<h2>Hoverable Dropdown</h2>
<p>Move the mouse over the text below to open the dropdown content.</p>
<div class="dropdown"> Mouse over me
 <div class="dropdown-content">
 <p>Hello World!</p>
 </div>
</div>
</body>
</html>

```

### Output:



## Example Explained

**HTML)** Use any element to open the dropdown content, e.g. a <span>, or a <button> element.

Use a container element (like <div>) to create the dropdown content and add whatever you want inside of it.

Wrap a <div> element around the elements to position the dropdown content correctly with CSS.

**CSS)** The **.dropdown** class use **position:fixed**, which is needed when we want the dropdown content to be placed right below the dropdown button (using **position:absolute**).

The **.dropdown-content** class holds the actual dropdown content. It is hidden by default, and will be displayed on hover (see below). Note the **min-width** is set to 150px. Feel free to change this. **Tip:** If you want the width of the dropdown content to be as wide as the dropdown button, set the **width** to 100% (and **overflow:auto** to enable scroll on small screens).

## DROPDOWN MENU

Create a dropdown menu that allows the user to choose an option from a list:

### Example:

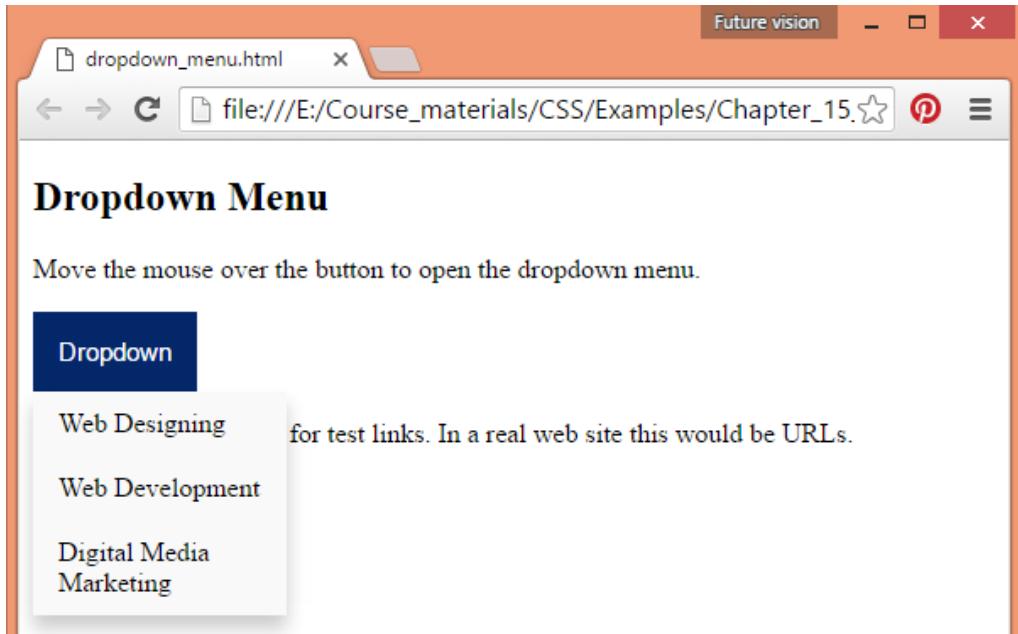
```
<!DOCTYPE html>
<html>
<head>
<style>
.dropbtn {
 background-color: #4c53af;
 color: white;
 padding: 15px;
 font-size: 15px;
 border: none;
 cursor: pointer;
}
.dropdown {
 position: relative;
 display: inline-block;
}
.dropdown-content {
 display: none;
 position: absolute;
 background-color: #f9f9f9;
 min-width: 150px;
 box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
}
.dropdown-content a {
```

```

 color: black;
 padding: 10px 15px;
 text-decoration: none;
 display: block;
 }
.dropdown-content a:hover {background-color: #bcd1f8}
.dropdown:hover .dropdown-content {
 display: block;
}
.dropdown:hover .dropbtn {
 background-color: #032769;
}
</style>
</head>
<body>
<h2>Dropdown Menu</h2>
<p>Move the mouse over the button to open the dropdown menu.</p>
<div class="dropdown">
 <button class="dropbtn">Dropdown</button>
 <div class="dropdown-content">
 Web Designing
 Web Development
 Digital Media Marketing
 </div>
</div>
<p>Note: We use href="#" for test links. In a real web site this would be URLs.</p>
</body>
</html>

```

**Output:**



## RIGHT-ALIGNED DROPDOWN CONTENT

If you want the dropdown menu to go from right to left, instead of left to right, add `right: 0;`

```
.dropdown-content {
 right: 0;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 .dropbtn {
 background-color: #4c53af;
 color: white;
 padding: 15px;
 font-size: 15px;
 border: none;
 cursor: pointer;
 }
 .dropdown {
 position: relative;
 }
```

```

 display: inline-block;
 }
 .dropdown-content {
 display: none;
 position: absolute;
 right: 0;
 background-color: #f9f9f9;
 min-width: 150px;
 box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
 }
 .dropdown-content a {
 color: black;
 padding: 10px 15px;
 text-decoration: none;
 display: block;
 }
 .dropdown-content a:hover {background-color: # bcd1f8}
 .dropdown:hover .dropdown-content {
 display: block;
 }
 .dropdown:hover .dropbtn {
 background-color: #032769;
 }

```

</style>

</head>

<body>

<h2>Aligned Dropdown Content</h2>

<p>Determine whether the dropdown content should go from left to right or right to left with the left and right properties.</p>

<div class="dropdown" style="float:left;">

    <button class="dropbtn">Left</button>

    <div class="dropdown-content" style="left:0;">

        <a href="#">List 1</a>

        <a href="#">List 2</a>

        <a href="#">List 3</a>

    </div>

</div>

<div class="dropdown" style="float:right;">

    <button class="dropbtn">Right</button>

    <div class="dropdown-content">

        <a href="#">List 1</a>

        <a href="#">List 2</a>

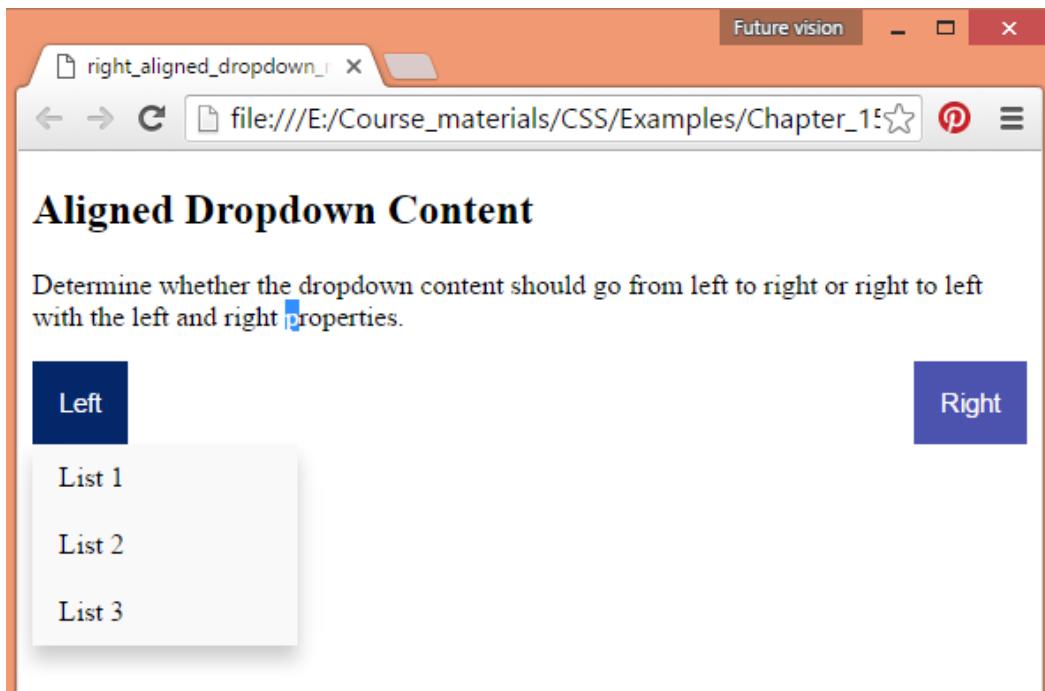
        <a href="#">List 3</a>

    </div>

</div>

```
</body>
</html>
```

**Output:**



# CHAPTER 16 - CSS POSITION PROPERTY

The **position** property specifies the type of positioning method used for an element.

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:

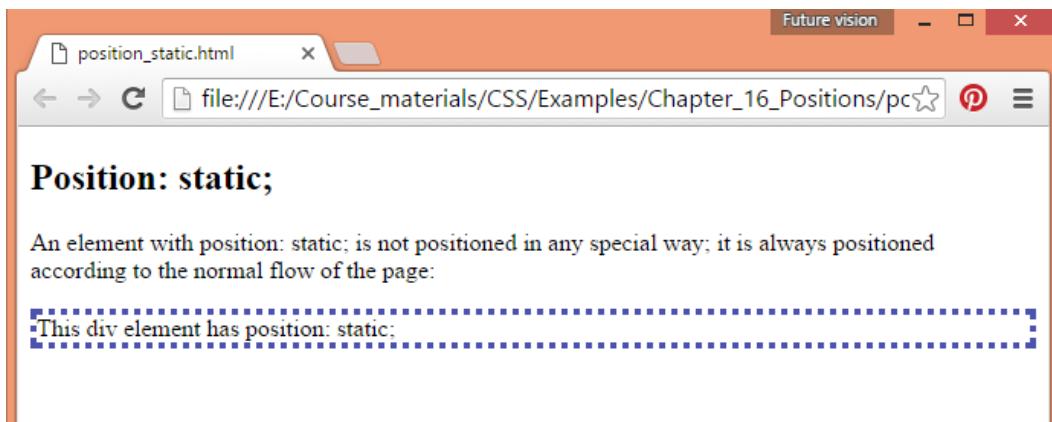
```
div.static {
 position: static;
 border: 4px dotted #4c53af;
}
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
 position: static;
 border: 4px dotted #4c53af;
}
</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>
```

## Output:



## 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.

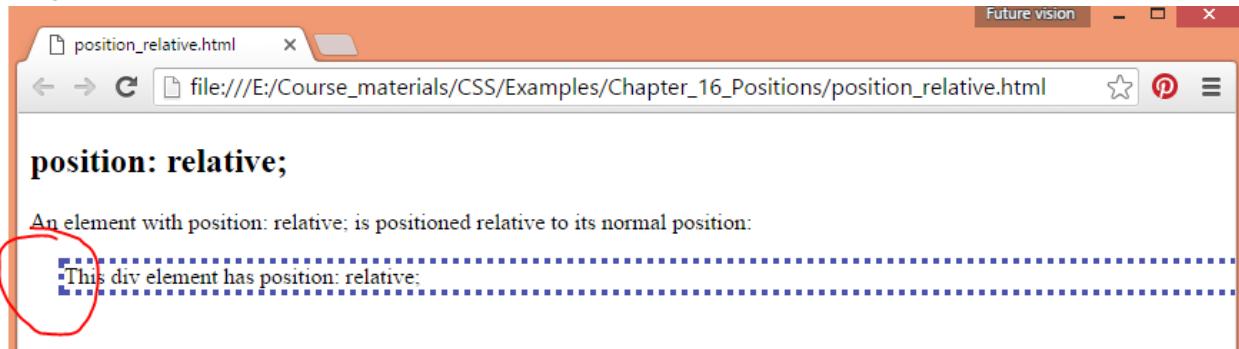
```
div.relative {
 position: relative;
 left: 20px;
 border: 4px dotted #4c53af;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
 position: relative;
 left: 20px;
 border: 4px dotted #4c53af;
}
</style>
</head>
<body>
<h2>position: relative;</h2>
<p>An element with position: relative; is positioned relative to its normal position:</p>
<div class="relative"> This div element has position: relative; </div>
</body>
```

```
</html>
```

## Output:



## 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;
 left: 0;
 width: 400px;
 border: 4px dotted #4c35af;
}
```

## Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
 position: fixed;
 bottom: 0;
 left: 0;
 width: 400px;
 border: 4px dotted #4c35af;
}
</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>
```

## Output:



## POSITION: ABSOLUTE

An element with **position: absolute;** is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

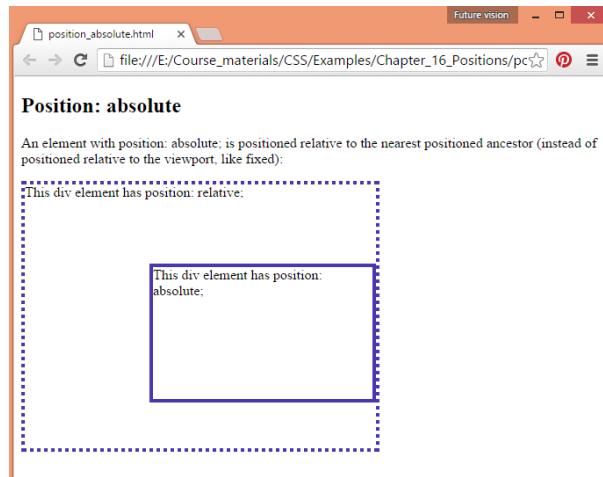
However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

**Note:** A "positioned" element is one whose position is anything except **static**.

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative1 {
 position: relative;
 width: 400px;
 height: 300px;
 border: 4px dotted #4c35af;
}
div.absolute {
 position: absolute;
 top: 90px;
 right: 0;
 width: 250px;
 height: 150px;
 border: 4px solid #4c35af;
}
</style>
</head>
<body>
<h2>Position: absolute</h2>
<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>
<div class="relative1">This div element has position: relative;
<div class="absolute">This div element has position: absolute;</div>
</div>
</body>
</html>
```

## Output:



## OVERLAPPING ELEMENTS

When elements are positioned, they can overlap other elements.

The **z-index** property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

```
#img3 {
 z-index: -1;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 #img3{
 z-index: -1;
 }
</style>
</head>
<body>
 <h1>This is a heading</h1>

 <p>Because the image has a z-index of -1, it will be placed behind the text.</p>
</body>
</html>
```

## Output:



## POSITIONING TEXT IN AN IMAGE

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
 position: relative;
}
.topleft {
 position: absolute;
 font-size: 20px;
 top: 10px;
 left: 15px;
}
#img5 {
 width: 100%;
 height: auto;
 opacity: 0.4;
}
</style>
</head>
<body>
<h2>Image Text</h2>
<div class="container">

 <div class="topleft">Top Left</div>
</div>
```

```
</body>
</html>
```

**Output:**



## CHAPTER 17- CSS HEIGHT AND WIDTH DIMENSIONS:

You have seen the border that surrounds every box i.e. element, the padding that can appear inside each box, and the margin that can go around them. In this chapter, we will learn how to change the dimensions of boxes.

We have the following properties that allow you to control the dimensions of a box.

- ✓ The **height** and **width** property is used to set the height of a box.
- ✓ The **max-height** property is used to set a maximum height that a box can be.
- ✓ The **min-height** property is used to set the minimum height that a box can be.
- ✓ The **max-width** property is used to set the maximum width that a box can be.
- ✓ The **min-width** property is used to set the minimum width that a box can be.

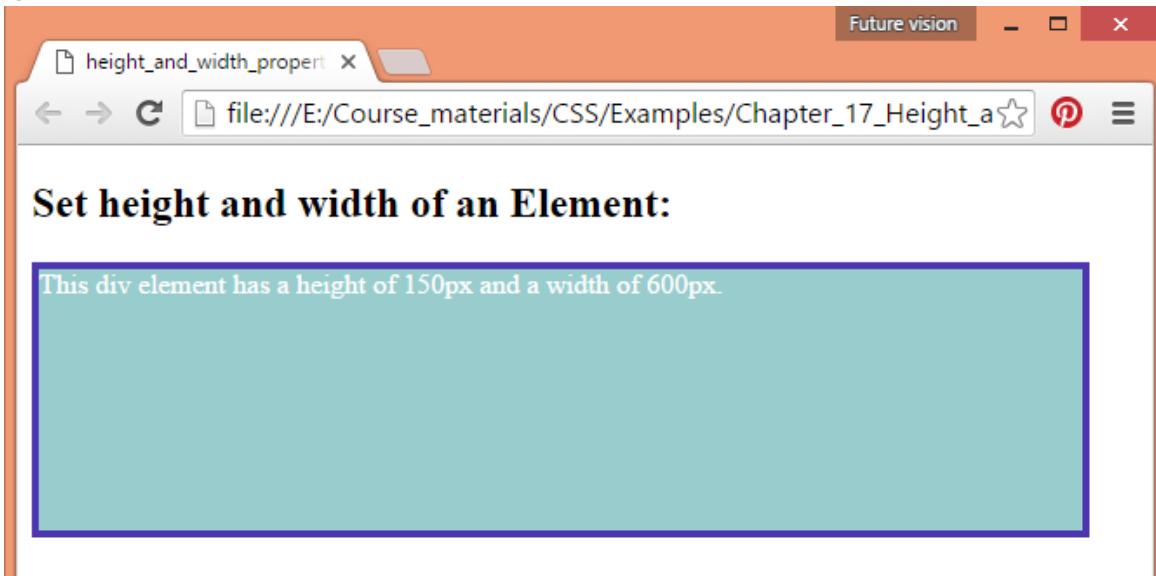
### HEIGHT AND WIDTH PROPERTY

```
div.a {
 width: 600px;
 height: 150px;
 border: 4px solid #4c35af;
}
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
div.a{
 background-color:#9CC;
 color:#FFF;
 width: 600px;
 height: 150px;
 border: 4px solid #4c35af;
}
</style>
</head>
<body>
<h2>Set height and width of an Element:</h2>
<div class="a">
This div element has a height of 150px and a width of 600px.
</div>
</body>
</html>
```

## Output:



## MAX-HEIGHT

The *max-height* property allows you to specify the maximum height of a box. The value of the max-height property can be a number, a length, or a percentage.

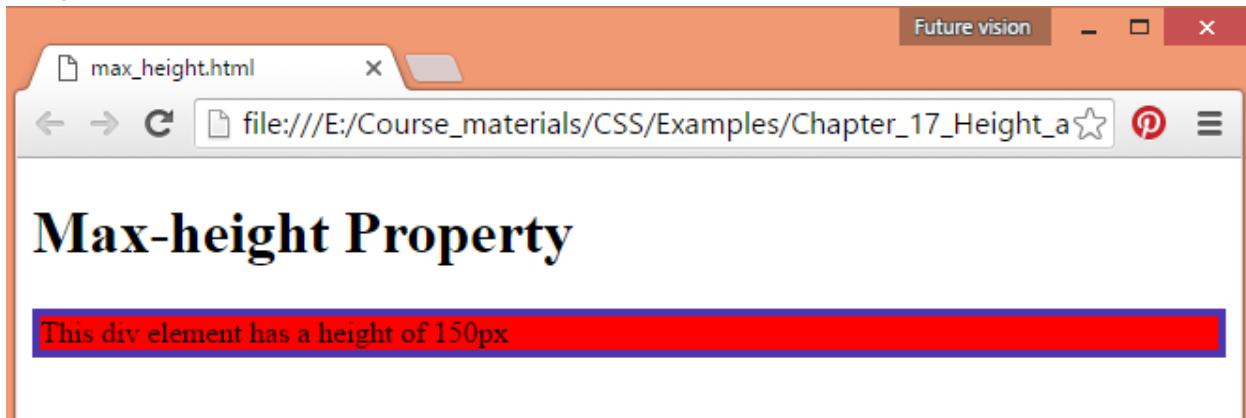
```
div.b{
 max-height: 150px;
 border: 4px solid #4c35af;
}
```

## Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 div.b{
 max-height: 150px;
 border: 4px solid #4c35af;
 }
</style>
</head>
<body>
<h1>
 max-height Property
</h1>
<div class="b">
```

```
This div element has a height of 150px
</div>
</body>
</html>
```

#### Output:



#### SETTING MAX-WIDTH

The **max-width** property is used to set the maximum width of an element.

The **max-width** can be specified in *length values*, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

The problem with the `<div>` above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.

Using **max-width** instead, in this situation, will improve the browser's handling of small windows.

```
div.c {
 max-width: 500px;
 height: 100px;
 border: 4px solid #4c35af;
}
```

#### MIN-HEIGHT PROPERTY

The *min-height* property allows you to specify the minimum height of a box. The value of the min-height property can be a number, a length, or a percentage.

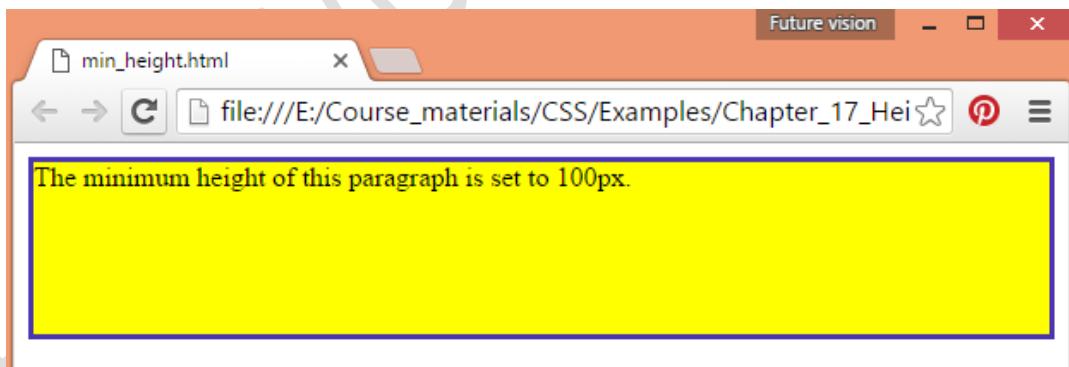
```
div .d{
```

```
min-height: 100px;
background-color: yellow;
border: 3px solid #4c35af;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 div.d {
 min-height: 100px;
 background-color: yellow;
 border: 3px solid #4c35af;
 }
</style>
</head>
<body>
<div class="d">The minimum height of this paragraph is set to 100px.</div>
</body>
</html>
```

### Output:



## MIN-WIDTH PROPERTY

The *min-width* property allows you to specify the minimum width of a box. The value of the min-width property can be a number, a length, or a percentage.

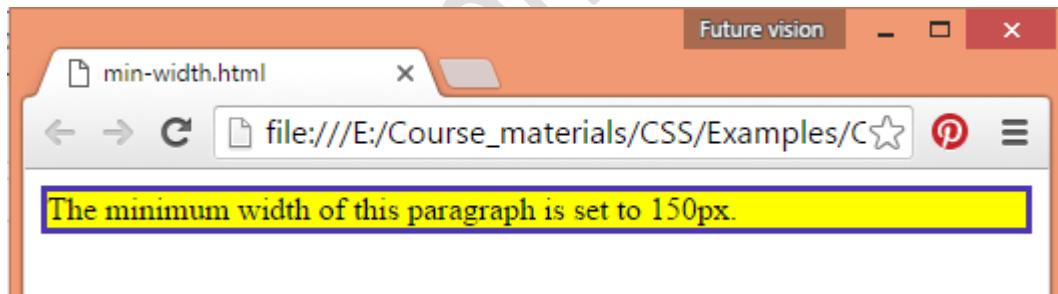
```
div.e {
 min-width: 150px;
 background-color: yellow;
```

```
border: 3px solid #4c35af;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 div.e {
 min-width: 150px;
 background-color: yellow;
 border: 3px solid #4c35af;
 }
</style>
</head>
<body>
<div class="e">The minimum width of this paragraph is set to 150px.</div>
</body>
</html>
```

### Output:



## CHAPTER 18 - CSS LAYOUTS FLOAT AND CLEAR

CSS also provides *table-layout* property to make your tables load much faster.

- ✓ The **float** property specifies whether or not an element should float.
- ✓ The **clear** property is used to control the behavior of floating elements.

### THE FLOAT PROPERTY

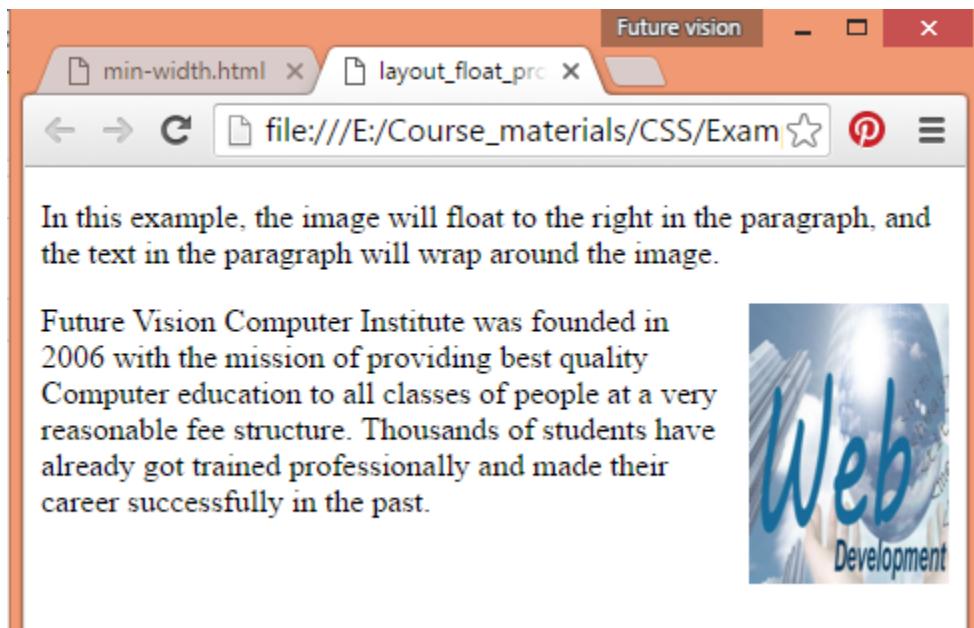
In its simplest use, the **float** property can be used to wrap text around images.

```
img.float {
 float: right;
 margin: 0 0 15px 15px;
}
```

#### **Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
img.float {
 float: right;
 margin: 0 0 15px 15px;
}
</style>
</head>
<body>
<p>In this example, the image will float to the right in the paragraph, and the text in the
paragraph will wrap around the image.</p>
<p>
Future Vision Computer Institute was founded in 2006 with the mission of providing best
quality Computer education to all classes of people at a very reasonable fee structure.
Thousands of students have already got trained professionally and made their career
successfully in the past.</p>
</body>
</html>
```

## Output:



## THE CLEAR PROPERTY

The **clear** property is used to control the behavior of floating elements.

Elements after a floating element will flow around it. To avoid this, use the **clear** property.

The **clear** property specifies on which sides of an element floating elements are not allowed to float:

```
div {
 clear: left;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
.div1 {
 float: left;
 width: 100px;
 height: 40px;
 margin: 15px;
 border: 4px solid #4c35af;
}
.div2 {
```

```

 border: 2px solid green;
 }
 .div3 {
 float: left;
 width: 100px;
 height: 40px;
 margin: 15px;
 border: 4px solid #4c35af;
 }
 .div4 {
 border: 2px solid green;
 clear: left;
 }

```

</style>

</head>

<body>

<h2>Without clear</h2>

<div class="div1">div1</div>

<div class="div2">div2 - Notice that the div2 element is after div1, in the HTML code. However, since div1 is floated to the left, this happens: the text in div2 is floated around div1, and div2 surrounds the whole thing.</div><br>

<h2>Using clear</h2>

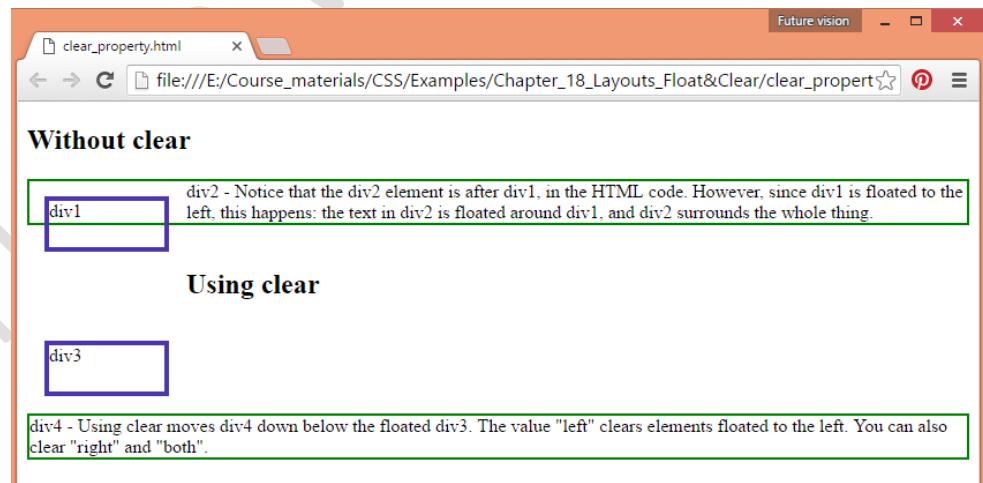
<div class="div3">div3</div>

<div class="div4">div4 - Using clear moves div4 down below the floated div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".</div>

</body>

</html>

## Output:



## THE CLEARFIX HACK - OVERFLOW: AUTO

If an element is taller than the element containing it, and it is floated, it will overflow outside of its container.

Then we can add `overflow: auto;` to the containing element to fix this problem:

```
.clearfix {
 overflow: auto;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 div#border {
 border: 4px solid #4c35af;
 }
 .img1 {
 float: right;
 }
 .clearfix {
 overflow: auto;
 }
 .img2 {
 float: right;
 }
</style>
</head>
<body>
 <p>In this example, the image is taller than the element containing it, and it is floated, so it overflows outside of its container:</p>
 <div id="border">
 Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past</div>
 <p style="clear:right">About Future Vision Computer Institute</p>
 <div id="border" class="clearfix">
```

Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past

</body>  
</html>

## Output:

The screenshot shows a web browser window titled "Future vision" displaying two examples of CSS layout issues. The top example, titled "clear\_property.html", contains text about Future Vision Computer Institute and an image of a globe labeled "Web Development". A blue box highlights the text area. Below it, another blue box highlights the image area. The bottom example, titled "clearfix\_hack\_overflow\_auto.html", also contains text about the institute and the same "Web Development" image, with a blue box highlighting the text area.

In this example, the image is taller than the element containing it, and it is floated, so it overflows outside of its container:

Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past

About Future Vision Computer Institute

Future Vision Computer Institute was founded in 2006 with the mission of providing best quality Computer education to all classes of people at a very reasonable fee structure. Thousands of students have already got trained professionally and made their career successfully in the past

## CHAPTER 19- CSS ANIMATABLE

Some CSS properties are animatable, meaning that they can be used in animations and transitions.

Animatable properties can change gradually from one value to another, like size, numbers, percentage and color.

```
/* Code for Chrome, Safari and Opera */
@-webkit-keyframes mymove {
 from {background-color: yellow;}
 to {background-color: blue;}
}

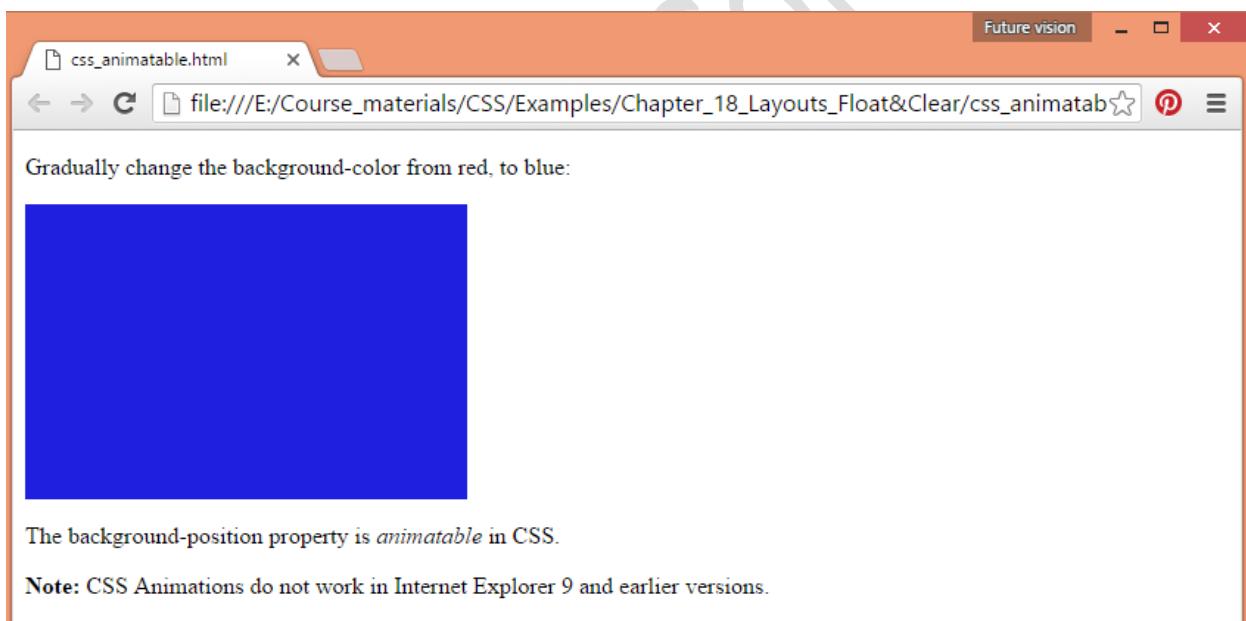
/* Standard syntax */
@keyframes mymove {
 from {background-color: yellow;}
 to {background-color: blue;}
}
```

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
#myDIV {
 width: 300px;
 height: 200px;
 background: yellow;
 -webkit-animation: mymove 6s infinite; /* Chrome, Safari, Opera */
 animation: mymove 6s infinite;
}
/* Chrome, Safari, Opera */
@-webkit-keyframes mymove {
 from {background-color: yellow;}
 to {background-color: blue;}
}
/* Standard syntax */
```

```
@keyframes mymove {
 from {background-color: yellow;}
 to {background-color: blue;}
}
</style>
</head>
<body>
<p>Gradually change the background-color from red, to blue:</p>
<div id="myDIV"></div>
<p>The background-position property is animatable in CSS.</p>
<p>Note: CSS Animations do not work in Internet Explorer 9 and earlier
versions.</p>
</body>
</html>
```

### Output:



## CHAPTER 20-CSS ATTRIBUTE SELECTORS

The [attribute] selector is used to select elements with a specified attribute.

The following example selects all  elements with a target attribute:

```
a[target] {
 background-color: lightblue;
}
```

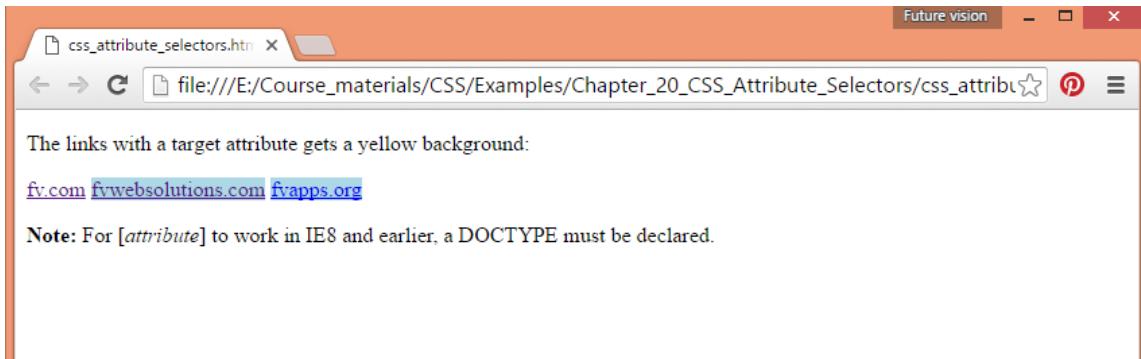
### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
a[target] {
 background-color: lightblue;
}
</style>
</head>
<body>

<p>The links with a target attribute gets a yellow background:</p>
fv.com
fvwebsolutions.com
<a
href="https://play.google.com/store/apps/developer?id=Future+Vision+computers,+Sur
at" target="_top">fvapps.org
<p style="color:#000;">Note: For [<i>attribute</i>] to work in IE8 and earlier, a
DOCTYPE must be declared.</p>

</body>
</html>
```

## Output:



## CSS [ATTRIBUTE~="VALUE"] SELECTOR

The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.

```
[title~="css"] {
 border: 4px solid red;
}
```

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
 [title~=css] {
 border: 4px solid red;
 }
</style>
</head>
<body>
 <p>All images with the title attribute containing the word "css" get a yellow border.</p>

 <p style="color:#000">Note: For [<i>attribute</i>~=<i>value</i>] to work in IE8
 and earlier, a DOCTYPE must be declared.</p>
</body>
</html>
```

## Output:



## STYLING FORMS

The attribute selectors can be useful for styling forms without class or ID:

```
input[type="text"] {
 width: 160px;
 display: block;
 margin-bottom: 10px;
 background-color: yellow;
}

input[type="button"] {
 width: 130px;
 margin-left: 35px;
 display: block;
}
```

### Example:

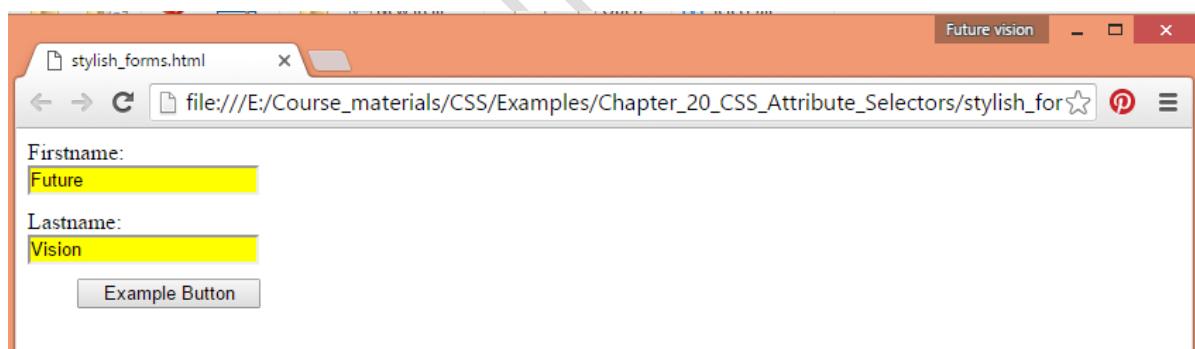
```
<!DOCTYPE html>
<html>
<head>
<style>
 input[type=text] {
 width: 160px;
 display: block;
```

```

 margin-bottom: 10px;
 background-color: yellow;
 }
 input[type=button] {
 width: 130px;
 margin-left: 35px;
 display: block;
 }
</style>
</head>
<body>
<form name="input" action="" method="get">
 Firstname:<input type="text" name="Name" value="Future" size="20">
 Lastname:<input type="text" name="Name" value="Vision" size="20">
 <input type="button" value="Example Button">
</form>
</body>
</html>

```

### Output:



## CHAPTER 21-The Z-index property

When elements are positioned, they can overlap other elements.

The **z-index** property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order:

**Example:**

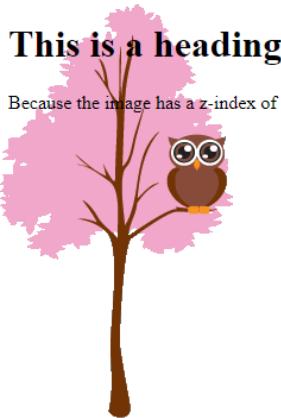
```
<!DOCTYPE html>
<html>
<head>
<style>
img {
 position: absolute;
 left: 0px;
 top: 0px;
 z-index: -1;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

<p>Because the image has a z-index of -1, it will be placed behind the text.</p>

</body>
</html>
```

## Output:



Because the image has a z-index of -1, it will be placed behind the text.

## Another Z-index Example

Here we see that an element with greater stack order is always above an element with a lower stack order:

### Example:

```
<!DOCTYPE html>

<html>
<head>
<style>

.container {
 position: relative;
}

.black-box {
 position: relative;
 z-index: 1;
 border: 2px solid black;
 height: 100px;
 margin: 30px;
}
```

```
.gray-box {
 position: absolute;
 z-index: 3; /* gray box will be above both green and black box */
 background: lightgray;
 height: 60px;
 width: 70%;
 left: 50px;
```

FutureVisionComputers



```
top: 50px;
}

.green-box {
position: absolute;
z-index: 2; /* green box will be above black box */
background: lightgreen;
width: 35%;
left: 270px;
top: -15px;
height: 100px;
}

</style>
</head>

<body>

<h1>Z-index Example</h1>

<p>An element with greater stack order is always above an element with a lower stack order.</p>

<div class="container">
 <div class="black-box">Black box (z-index: 1)</div>
 <div class="gray-box">Gray box (z-index: 3)</div>
 <div class="green-box">Green box (z-index: 2)</div>
</div>

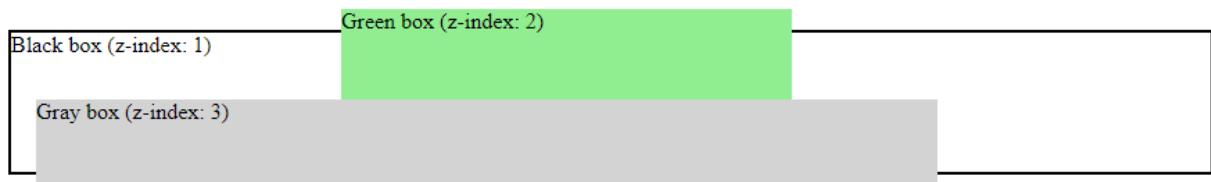
</body>

</html>
```

## Output:

### Z-index Example

An element with greater stack order is always above an element with a lower stack order.



### Without Z-index

If two positioned elements overlap each other without a **z-index** specified, the element defined **last in the HTML code** will be shown on top.

#### Example:

```
<!DOCTYPE html>

<html>
<head>
<style>
.container {
 position: relative;
}

.black-box {
 position: relative;
 border: 2px solid black;
 height: 100px;
 margin: 30px;
}

.gray-box {
 position: absolute;
```

```
background: lightgray;
height: 60px;
width: 70%;
left: 50px;
top: 50px;
}
```

```
.green-box {
position: absolute;
background: lightgreen;
width: 35%;
left: 270px;
top: -15px;
height: 100px;
}
</style>
</head>
<body>
```

```
<h1>Overlapping elements</h1>
```

```
<p>If two positioned elements overlap each other without a z-index specified,
the element defined last in the HTML code will be shown on top:</p>
```

```
<div class="container">
<div class="black-box">Black box</div>
<div class="gray-box">Gray box</div>
<div class="green-box">Green box</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

**Output:**

## Overlapping elements

If two positioned elements overlap each other without a z-index specified, the element defined last in the HTML code will be shown on top:



## CHAPTER 22-CSS Combinators

A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

### Descendant Selector (space)

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all `<p>` elements inside `<div>` elements:

#### **Example:**

```
<!DOCTYPE html>

<html>
<head>
<style>
div p {
 background-color: yellow;
}
</style>
</head>
<body>
<h2>Descendant Selector</h2>
<p>The descendant selector matches all elements that are descendants of a specified element.</p>
```

```
<div>

 <p>Paragraph 1 in the div.</p>
 <p>Paragraph 2 in the div.</p>
 <section><p>Paragraph 3 in the div.</p></section>

</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
</body>
</html>
```

**Output:**

### Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

## Child Selector (>)

The child selector selects all elements that are the children of a specified element.

The following example selects all `<p>` elements that are children of a `<div>` element:

### **Example:**

```
<!DOCTYPE html>

<html>
<head>
<style>
div > p {
 background-color: yellow;
}
</style>
</head>
<body>
<h2>Child Selector</h2>
<p>The child selector (>) selects all elements that are the children of a specified element.</p>
<div>
<p>Paragraph 1 in the div.</p>
<p>Paragraph 2 in the div.</p>
<section>
<!-- not Child but Descendant -->
<p>Paragraph 3 in the div (inside a section element).</p>
</section>
<p>Paragraph 4 in the div.</p>
</div>

<p>Paragraph 5. Not in a div.</p>
<p>Paragraph 6. Not in a div.</p>
```

```
</body>
</html>
```

## Output:

### Child Selector

The child selector (>) selects all elements that are the children of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div (inside a section element).

Paragraph 4 in the div.

Paragraph 5. Not in a div.

Paragraph 6. Not in a div.

### Adjacent Sibling Selector (+)

The adjacent sibling selector is used to select an element that is directly after another specific element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects the first `<p>` element that are placed immediately after `<div>` elements:

#### Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

div + p {

 background-color: yellow;

}

</style>

</head>
```

```
<body>

<h2>Adjacent Sibling Selector</h2>

<p>The + selector is used to select an element that is directly after another specific element.</p>

<p>The following example selects the first p element that are placed immediately after div elements:</p>

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

</div>

<p>Paragraph 3. After a div.</p>

<p>Paragraph 4. After a div.</p>

<div>

<p>Paragraph 5 in the div.</p>

<p>Paragraph 6 in the div.</p>

</div>

<p>Paragraph 7. After a div.</p>

<p>Paragraph 8. After a div.</p>

</body>

</html>
```

**Output:**

**Adjacent Sibling Selector**

The + selector is used to select an element that is directly after another specific element.

The following example selects the first p element that are placed immediately after div elements:

Paragraph 1 in the div.

Paragraph 2 in the div.

**Paragraph 3. After a div.**

Paragraph 4. After a div.

Paragraph 5 in the div.

Paragraph 6 in the div.

**Paragraph 7. After a div.**

Paragraph 8. After a div.

## General Sibling Selector (~)

The general sibling selector selects all elements that are next siblings of a specified element.

The following example selects all `<p>` elements that are next siblings of `<div>` elements:

### **Example:**

```
<!DOCTYPE html>

<html>
<head>
<style>
div ~ p {
 background-color: yellow;
}
</style>
</head>
<body>
<h2>General Sibling Selector</h2>
<p>The general sibling selector (~) selects all elements that are next siblings of a specified element.</p>
<p>Paragraph 1.</p>
<div>
<p>Paragraph 2.</p>
</div>
<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>
</body>
</html>
```

## Output:

### General Sibling Selector

The general sibling selector (~) selects all elements that are next siblings of a specified element.

Paragraph 1.

Paragraph 2.

Paragraph 3.

Some code.

Paragraph 4.

## CHAPTER 23-Pseudo classes

# What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

### Syntax:

The syntax of pseudo-classes:

```
selector:pseudo-class {
 property: value;
}
```

### Anchor Pseudo-classes

Links can be displayed in different ways:

#### Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

/* unvisited link */

a:link {

 color: red;

}
```

```
/* visited link */

a:visited {
 color: green;
}

/* mouse over link */

a:hover {
 color: hotpink;
}

/* selected link */

a:active {
 color: blue;
}

</style>

</head>

<body>

<h2>Styling a link depending on state</h2>

<p>This is a link</p>

<p>Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>

<p>Note: a:active MUST come after a:hover in the CSS definition in order to be effective.</p>

</body>

</html>
```

## Output:

### Styling a link depending on state

[This is a link](#)

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

## Pseudo-classes & HTML classes

Pseudo-classes can be combined with HTML classes:

When you hover over the link in the example, it will change color:

### Example:

```
<!DOCTYPE html>

<html>
 <head>
 <style>
 a.highlight:hover {
 color: #ff0000;
 font-size: 22px;
 }
 </style>
 </head>
 <body>
 <h2>Pseudo-classes and HTML Classes</h2>
 <p>When you hover over the first link below, it will change color and font size:</p>
 <p>CSS Syntax</p>
 <p>CSS Tutorial</p>
 </body>
</html>
```

**Output:**

## Pseudo-classes and HTML Classes

When you hover over the first link below, it will change color and font size:

[CSS Syntax](#)

[CSS Tutorial](#)

### Hover on <div>

An example of using the :hover pseudo-class on a <div> element:

#### **Example:**

```
<!DOCTYPE html>

<html>
<head>
<style>
div {
 background-color: green;
 color: white;
 padding: 25px;
 text-align: center;
}
div:hover {
 background-color: blue;
}
</style>
```

```
</head>

<body>

<p>Mouse over the div element below to change its background color:</p>

<div>Mouse Over Me</div>

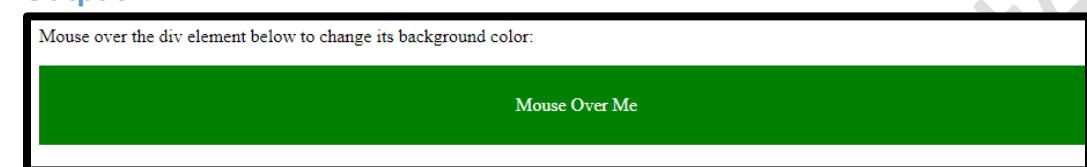
</body>

</html>
```

**Output:**

Mouse over the div element below to change its background color:

Mouse Over Me



## CHAPTER 24- CSS Pseudo elements

### What are Pseudo-Elements?

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

### Syntax:

```
selector::pseudo-element {
 property: value;
}
```

### The ::first-line Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all <p> elements:

### Example:

```
<!DOCTYPE html>

<!DOCTYPE html>

<html>

<head>

<style>

p::first-line {

 color: #ff0000;

 font-variant: small-caps;

}

</style>

</head>

<body>
```

```
<p>You can use the ::first-line pseudo-element to add a special effect to the first line of a text.
Some more text. And even more, and more, and more, and more, and more, and more, and more,
and more, and more, and more, and more, and more.</p>
```

</body>

</html>

**Note:** The `::first-line` pseudo-element can only be applied to block-level elements.

The following properties apply to the `::first-line` pseudo-element:

- font properties
  - color properties
  - background properties
  - word-spacing
  - letter-spacing
  - text-decoration
  - vertical-align
  - text-transform
  - line-height
  - clear

## Output:

# The ::first-letter Pseudo-element

The ::first-letter pseudo-element is used to add a special style to the first letter of a text.

The following example formats the first letter of the text in all `<p>` elements:

### Example:

```
<!DOCTYPE html>

<html>
<head>
<style>

p::first-letter {
 color: #ff0000;
 font-size: xx-large;
}

</style>
</head>
<body>

<p>You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!</p>

</body>
</html>
```

**Note:** The `::first-letter` pseudo-element can only be applied to block-level elements.

The following properties apply to the `::first-letter` pseudo-element:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

## Output:

You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!

## Pseudo-elements & HTML classes

Pseudo-elements can be combined with HTML classes:

The example above will display the first letter of paragraphs with class="intro", in red and in a larger size.

### Example:

```
<!DOCTYPE html>

<html>
 <head>
 <style>
 p.intro::first-letter {
 color: #ff0000;
 font-size: 200%;
 }
 </style>
 </head>
 <body>
 <p class="intro">This is an introduction.</p>
 <p>This is a paragraph with some text. A bit more text even.</p>
 </body>
</html>
```

## Output:

```
This is an introduction.
This is a paragraph with some text. A bit more text even.
```

## Multiple Pseudo-elements

Several pseudo-elements can also be combined.

In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color:

### Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

p::first-letter {

 color: #ff0000;

 font-size: xx-large;

}

p::first-line {

 color: #0000ff;

 font-variant: small-caps;

}

</style>

</head>
```

```
<body>
<p>You can combine the ::first-letter and ::first-line pseudo-elements to add a special effect to the first letter and the first line of a text!</p>
</body>
</html>
```

#### Output:

You can combine the ::first-letter and ::first-line pseudo-elements to add a special effect to the first letter and the first line of a text!

### CSS- The ::before Pseudo-element

The ::before pseudo-element can be used to insert some content before the content of an element.

The following example inserts an image before the content of each `<h1>` element:

#### Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

h1::before {

 content: url(smiley.gif);

}

</style>

</head>
```

```
<body>
<h1>This is a heading</h1>
<p>The ::before pseudo-element inserts content before the content of an element.</p>
<h1>This is a heading</h1>
</body>
</html>
```

**Output:**

 **This is a heading**

The ::before pseudo-element inserts content before the content of an element.

 **This is a heading**

## CSS- The ::after Pseudo-element

The ::after pseudo-element can be used to insert some content after the content of an element.

The following example inserts an image after the content of each `<h1>` element:

**Example:**

```
<!DOCTYPE html>

<html>
<head>
<style>
h1::after {
 content: url(smiley.gif);
}
</style>
</head>
```

```
<body>

<h1>This is a heading</h1>

<p>The ::after pseudo-element inserts content after the content of an element.</p>

<h1>This is a heading</h1>

</body>

</html>
```

**Output:**

**This is a heading** 😊

The ::after pseudo-element inserts content after the content of an element.

**This is a heading** 😊

## CSS- The ::marker Pseudo-element

The ::marker pseudo-element selects the markers of list items.

The following example styles the markers of list items:

**Example:**

```
<!DOCTYPE html>

<html>

<head>

<style>

::marker {

color: red;

font-size: 23px;

}

</style>

</head>
```

```
<body>

 Coffee
 Tea
 Milk

 First
 Second
 Third

</body>
</html>
```

#### Output:

- Coffee
  - Tea
  - Milk
- 1. First
  - 2. Second
  - 3. Third

## CSS- The ::selection Pseudo-element

The ::selection pseudo-element matches the portion of an element that is selected by a user.

The following CSS properties can be applied to ::selection: color, background, cursor, and outline.

The following example makes the selected text red on a yellow background:

**Example:**

```
<!DOCTYPE html>

<html>

<head>

<style>

::selection {

color: red;

background: yellow;

}

</style>

</head>

<body>

<h1>Select some text on this page:</h1>

<p>This is a paragraph.</p>

<div>This is some text in a div element.</div>

</body>

</html>
```

**Output:**

**Select some text on this page:**

This is a paragraph.

This is some text in a div element.

## CHAPTER 25- CSS Navigation Bars

Having easy-to-use navigation is important for any web site.

With CSS you can transform boring HTML menus into good-looking navigation bars.

### Navigation Bar = List of Links

A navigation bar needs standard HTML as a base.

In our examples we will build the navigation bar from a standard HTML list.

A navigation bar is basically a list of links, so using the `<ul>` and `<li>` elements makes perfect sense:

#### **Example:**

```
<!DOCTYPE html>

<html>

<body>

Home

News

Contact

About

<p>Note: We use href="#" for test links. In a real web site this would be URLs.</p>

</body>

</html>
```

#### **Output:**

- [Home](#)
- [News](#)
- [Contact](#)
- [About](#)

Note: We use href="#" for test links. In a real web site this would be URLs.

FutureVisionComputers



## Vertical Navbar

To build a vertical navigation bar, you can style the `<a>` elements inside the list, in addition to the code from the previous page:

### **Example:**

```
<!DOCTYPE html>

<html>
<head>
<style>
ul {
 list-style-type: none;
 margin: 0;
 padding: 0;
}

li a {
 display: block;
 width: 60px;
 background-color: #dddddd;
}
</style>
</head>
<body>

Home
News
Contact
About

```

```
<p>A background color is added to the links to show the link area.</p>
```

```
<p>Notice that the whole link area is clickable, not just the text.</p>
```

```
</body>
```

```
</html>
```

#### Output:

```
Home
News
Contact
About
```

A background color is added to the links to show the link area.

Notice that the whole link area is clickable, not just the text.

## Horizontal Navigation Bar Examples

Create a basic horizontal navigation bar with a dark background color and change the background color of the links when the user moves the mouse over them:

#### Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

ul {
list-style-type: none;
margin: 0;
padding: 0;
overflow: hidden;
background-color: #333;
```

```
}

li {
 float: left;
}

li a {
 display: block;
 color: white;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
}

li a:hover {
 background-color: #111;
}

</style>

</head>

<body>

Home

News

Contact

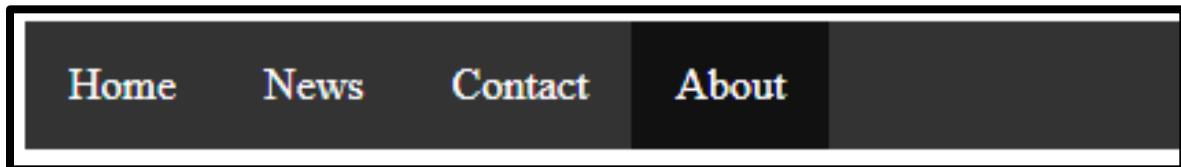
About

</body>
```



</html>

Output:



FutureVisionComputers



## CHAPTER 26- CSS Website Layout

A website is often divided into headers, menus, content and a footer, so let's build a website layout in this chapter

### Header

A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:

#### **Example:**

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>CSS Website Layout</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {

margin: 0;

}

/* Style the header */

.header {

background-color: #f1f1f1;

padding: 20px;

text-align: center;

}

</style>

</head>

<body>
```

```
<div class="header">
 <h1>Future Vision Computers</h1>
</div>
</body>
</html>
```

**Output:**



Future Vision Computers

## Navigation bar

A navigation bar contains a list of links to help visitors navigating through your website:

**Example:**

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>CSS Website Layout</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
* {
 box-sizing: border-box;
}

body {
 margin: 0;
}

/* Style the header */
```



```
.header {
background-color: #f1f1f1;
padding: 20px;
text-align: center;
}
/* Style the top navigation bar */
.topnav {
overflow: hidden;
background-color: #333;
}
/* Style the topnav links */
.topnav a {
float: left;
display: block;
color: #f2f2f2;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}
/* Change color on hover */
.topnav a:hover {
background-color: #ddd;
color: black;
}
</style>
</head>
```



```
<body>
<div class="header">
 <h1>Future Vision Computers</h1>
</div>
<div class="topnav">
 Home
 Our courses
 Contact Us
</div>
</body>
</html>
```

**Output:**



**Content**

The layout in this section, often depends on the target users. The most common layout is one (or combining them) of the following:

- 1-column (often used for mobile browsers)
- 2-column (often used for tablets and laptops)
- 3-column layout (only used for desktops)

### Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>CSS Website Layout</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

* {

 box-sizing: border-box;
}

body {

 margin: 0;
}

/* Style the header */

.header {

 background-color: #f1f1f1;
 padding: 20px;
 text-align: center;
}

/* Style the top navigation bar */

.topnav {

 overflow: hidden;
 background-color: #333;
}


```



```
/* Style the topnav links */

.topnav a {

float: left;

display: block;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

/* Change color on hover */

.topnav a:hover {

background-color: #ddd;

color: black;

}

/* Create three equal columns that floats next to each other */

.column {

float: left;

width: 33.33%;

padding: 15px;

}

/* Clear floats after the columns */

.row:after {

content: "";

display: table;

clear: both;
```



```
}

/* Responsive layout - makes the three columns stack on top of each other instead of next to
each other */

@media screen and (max-width:600px) {

.column {

width: 100%;

}

}

</style>

</head>

<body>

<div class="header">

<h1>Future Vision Computers</h1>

<p>Resize the browser window to see the responsive effect.</p>

</div>

<div class="topnav">

Home

Our courses

Contact us

</div>

<div class="row">

<div class="column">

<h2>Graphic Designing</h2>

<p>Graphic design is a craft where professionals create visual content to communicate
messages. By applying visual hierarchy and page layout techniques, designers use typography


```



and pictures to meet users' specific needs and focus on the logic of displaying elements in interactive designs, to optimize the user experience.</p>

</div>

<div class="column">

<h2>Web Designing</h2>

<p>Web design refers to the design of websites that are displayed on the internet. It usually refers to the user experience aspects of website development rather than software development.</p>

</div>

<div class="column">

<h2>App Development</h2>

<p>Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.</p>

</div>

</div>

</body>

</html>

### Output:

# Future Vision Computers

Resize the browser window to see the responsive effect.

[Home](#) [Our courses](#) [Contact us](#)

### Graphic Designing

Graphic design is a craft where professionals create visual content to communicate messages. By applying visual hierarchy and page layout techniques, designers use typography and pictures to meet users' specific needs and focus on the logic of displaying elements in interactive designs, to optimize the user experience.

### Web Designing

Web design refers to the design of websites that are displayed on the internet. It usually refers to the user experience aspects of website development rather than software development.

### App Development

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.



## Footer

The footer is placed at the bottom of your page. It often contains information like copyright and contact info:

### Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>CSS Website Layout</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

* {

 box-sizing: border-box;
}

body {

 margin: 0;
}

/* Style the header */

.header {

 background-color: #f1f1f1;
 padding: 20px;
 text-align: center;
}

/* Style the top navigation bar */

.topnav {

 overflow: hidden;
```



```
background-color: #333;
}

/* Style the topnav links */

.topnav a {

float: left;

display: block;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;
}

/* Change color on hover */

.topnav a:hover {

background-color: #ddd;

color: black;
}

/* Create three equal columns that floats next to each other */

.column {

float: left;

width: 33.33%;

padding: 15px;
}

/* Clear floats after the columns */

.row:after {

content: "";
```



```
display: table;

clear: both;

}

/* Responsive layout - makes the three columns stack on top of each other instead of next to
each other */

@media screen and (max-width:600px) {

.column {

width: 100%;

}

}

/* Style the footer */

.footer {

background-color: #f1f1f1;

padding: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class="header">

<h1>Future Vision Computers</h1>

<p>Resize the browser window to see the responsive effect.</p>

</div>

<div class="topnav">

Home
```



```
Our courses

Contact us

</div>

<div class="row">

<div class="column">

<h2>Graphic Designing</h2>

<p>Graphic design is a craft where professionals create visual content to communicate messages. By applying visual hierarchy and page layout techniques, designers use typography and pictures to meet users' specific needs and focus on the logic of displaying elements in interactive designs, to optimize the user experience.</p>

</div>

<div class="column">

<h2>Web Designing</h2>

<p>Web design refers to the design of websites that are displayed on the internet. It usually refers to the user experience aspects of website development rather than software development.</p>

</div>

<div class="column">

<h2>App Development</h2>

<p>Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.</p>

</div>

</div>

<div class="footer">

<p>Footer</p>

</div>
```

```
</body>
```

```
</html>
```

**Output:**

# Future Vision Computers

Resize the browser window to see the responsive effect.

Home   Our courses   Contact us

<b>Graphic Designing</b> <p>Graphic design is a craft where professionals create visual content to communicate messages. By applying visual hierarchy and page layout techniques, designers use typography and pictures to meet users' specific needs and focus on the logic of displaying elements in interactive designs, to optimize the user experience.</p>	<b>Web Designing</b> <p>Web design refers to the design of websites that are displayed on the internet. It usually refers to the user experience aspects of website development rather than software development.</p>	<b>App Development</b> <p>Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Footer