

CSS3: Shadow, Text, Fonts and Transforms

CSS3 - shadow

CSS3 supported to add shadow to text or elements. Shadow property has divided as follows:

- Text shadow
- Box Shadow

Text shadow

CSS3 supported to add shadow effects to text.

CSS3 text-shadow Property Syntax

text-shadow: *h-shadow v-shadow blur-radius color|none|initial|inherit;*

Note: To add more than one shadow to the text, add a comma-separated list of shadows.

Property Values

Value	Description
<i>h-shadow</i>	Required. The position of the horizontal shadow. <i>Negative values are allowed.</i>
<i>v-shadow</i>	Required. The position of the vertical shadow. <i>Negative values are allowed.</i>
<i>blur-radius</i>	Optional. The blur radius. <i>Default value is 0</i>
<i>color</i>	Optional. The color of the shadow.
none	Default value. No shadow
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

Following is the example to add shadow effects to text:

```
<html>
<head>
<style>
  h1 {text-shadow: 2px 2px ;}
  h2 {text-shadow: 2px 2px red ;}
  h3 {text-shadow: 2px 2px 5px red;}
  h4 {color: white; text-shadow: 2px 2px 4px #000000;}
  h5 {text-shadow: 0 0 3px #FF0000; }
  h6 {text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF; }
  p {color: white;
    text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;}
</style>
</head>
<body>
  <h1>Information Technology</h1>
  <h2>Information Technology</h2>
  <h3>Information Technology</h3>
  <h4>Information Technology</h4>
  <h5>Information Technology</h5>
  <h6>Information Technology</h6>
  <p>Information Technology</p>
</body> </html>
```

It will produce the following result:

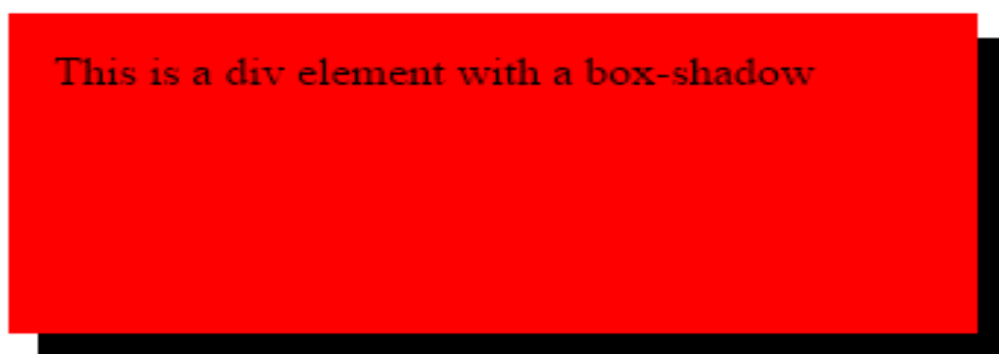


Box shadow

Used to add shadow effects to elements, following is the example to add shadow effects to element

```
<html>
  <head>
    <style>
      div {
        width:300px;
        height: 100px;
        padding:15px;
        background-color: red;
        box- shadow: 10px 10px;}
    </style>
  </head>
  <body>
    <div>This is a div element with a box-shadow</div>
  </body>
</html>
```

It will produce the following result:



CSS3 - Text

CSS3 contained several extra features, which is added later on

- [text-overflow](#)
- [word-break](#)
- [word-wrap](#)
- [text-align-last](#)

There are following most commonly used property in CSS3

Values	Description
text-overflow	Used to determines how overflowed content that is not displayed is signaled to users
word-break	Used to break the line based on word
word-wrap	Used to break the line and wrap onto next line
text-align-last	Used to align the last line of the text

CSS3 Text-Overflow Property

The text-overflow property determines how overflowed content that is not displayed is signaled to users.

CSS Syntax

Text-overflow: clip ellipsis <i>string</i> initial inherit;
--

Property Values

Value	Description
clip	Default value. Clips the text
ellipsis	Render an ellipsis ("...") to represent clipped text
string	Render the given string to represent clipped text
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

The sample example of text overflow is shown as follows:

```
<html>
<head>
  <style>
    p.text1 { white-space: nowrap;
      width: 200px;
      border: 1px solid #000000;
      overflow: hidden;
      text-overflow: clip; }
    p.text2 { white-space: nowrap;
      width: 200px;
      border: 1px solid #000000;
      overflow: hidden;
      text-overflow: ellipsis; }
  </style>
</head>
<body>
  <b>Original Text:</b>
  <p>IT Is the application of computers to store, retrieve, transmit and manipulate data,</p>
  <b>Text overflow:clip:</b>
  <p class="text1">IT Is the application of computers to store, retrieve, transmit and manipulate
    data,</p>
  <b>Text overflow:ellipsis</b>
  <p class="text2">IT Is the application of computers to store, retrieve, transmit and manipulate
    data,</p>
</body>
</html>
```

It will produce the following result:

Original Text:

IT Is the application of computers to store, retrieve, transmit and manipulate data.

Text overflow:clip:

IT Is the application of comput

Text overflow:ellipsis

IT Is the application of comp...

CSS3 word-break Property

Used to break the line,

CSS Syntax

Word-break: normal | break-all | keep-all | initial | inherit;

Property Values

Value	Description
normal	Default value. Break words according to their usual rules
break-all	Lines may break between any two letters
keep-all	Breaks are prohibited between pairs of letters
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

The following code shows the sample code of word breaking

```
<html>
<head>
  <style>
    p.text1 { width: 140px; border: 1px solid #000000; word-break: keep-all; }
    p.text2 { width: 140px; border: 1px solid #000000; word-break: break-all; }
  </style>
</head>
<body>
  <b>line break at hyphens:</b>
  <p class="text1">IT Is the application of computers to store, retrieve, transmit and manipulate
    data, often in the context of a business or other enterprise. IT is considered a subset of
    information and communications technology (ICT)..</p>
  <b>line break at any character</b>
  <p class="text2">IT Is the application of computers to store, retrieve, transmit and manipulate
    data, often in the context of a business or other enterprise. IT is considered a subset of
    information and communications technology (ICT)..</p>
</body>
```

It will produce the following result:

line break at hyphens:

is the application of computers to store, retrieve, transmit and manipulate data, often in the context of a business or other enterprise. IT is considered a subset of information and communications technology (ICT)..

line break at any character

is the application of computers to store, retrieve, transmit and manipulate data, often in the context of a business or other enterprise. IT is considered a subset of information and communications technology (ICT)..

CSS3 word-wrap Property

The CSS3 word-wrap property allows long words to be able to be broken and wrap onto the next line.

CSS Syntax

word-wrap: normal | break-word | initial | inherit;

Property Values

Value	Description
normal	Break words only at allowed break points
break-word	Allows unbreakable words to be broken
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

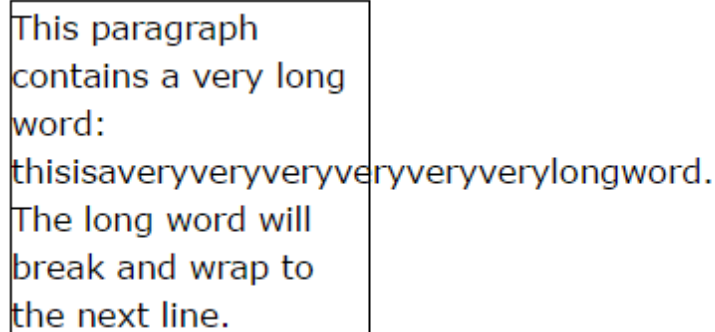
The following code will have sample syntax

```
p { word-wrap: break-word; }
```

Allow long words to be able to break and wrap onto the next line:

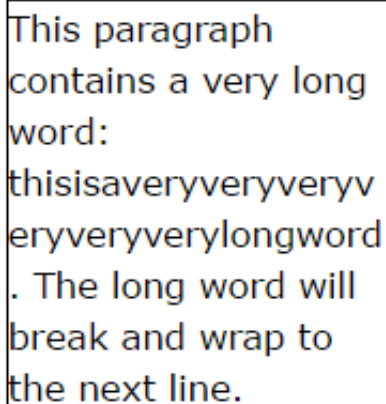
```
<html>
<head>
<style>
  p.test { width: 11em;
  border: 1px solid #000000;
  word-wrap: break-word;
  }
</style>
</head>
<body>
  <p class="test"> This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the
next line.</p>
</body>
</html>
```

If a word is too long to fit within an area, it expands outside:



This paragraph
contains a very long
word:
thisisaveryveryveryveryveryverylongword.
The long word will
break and wrap to
the next line.

The word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word:



This paragraph
contains a very long
word:
thisisaveryveryveryveryveryverylongword
. The long word will
break and wrap to
the next line.

Text-align-last

The text-align-last property specifies how to align the last line of a text. Notice that the text-align-last property sets the alignment for all last lines within the selected element. So, if you have a <div> with three paragraphs in it, text-align-last will apply to the last line of EACH of the paragraphs. To use text-align-last on only the last paragraph in the container, you can use : last child.

Note: In Edge/Internet Explorer the text-align-last property only works on text that has "text-align: justify".

CSS Syntax

text-align-last: auto|left|right|center|justify|start|end|initial|inherit;

```
<!DOCTYPE html>
<html>
<head>
<style>
div.a {
  text-align: justify; /* For Edge */
  -moz-text-align-last: right; /* For Firefox prior 58.0 */
  text-align-last: right;
}

div.b {
  text-align: justify; /* For Edge */
  -moz-text-align-last: center; /* For Firefox prior 58.0 */
  text-align-last: center;
}

div.c {
  text-align: justify; /* For Edge */
  -moz-text-align-last: justify; /* For Firefox prior 58.0 */
  text-align-last: justify;
}
</style>
</head>
<body>
<h1>The text-align-last Property</h1>
<h2>text-align-last: right:</h2>
<div class="a">
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
</div>

<h2>text-align-last: center:</h2>
<div class="b">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
</div>
<h2>text-align-last: justify:</h2>
<div class="c">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
</div></body></html>
```

It will produce the following result:

The text-align-last Property

text-align-last: right:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

text-align-last: center:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

text-align-last: justify:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

CSS3 Web Fonts - The @font-face Rule

Web fonts allow Web designers to use fonts that are not installed on the user's computer. When you have found/bought the font you wish to use, just include the font file on your web server, and it will be automatically downloaded to the user when needed. Your "own" fonts are defined within the CSS3 **@font-face** rule.

Different web fonts formats

Fonts	Description
TrueType Fonts (TTF)	TrueType is an outline font standard developed by Apple and Microsoft in the late 1980s, It became most common fonts for both windows and MAC operating systems
OpenType Fonts (OTF)	OpenType is a format for scalable computer fonts and developed by Microsoft
The Web Open Font Format (WOFF)	WOFF is used for develop web page and developed in the year of 2009. Now it is using by W3C recommendation.
SVG Fonts/Shapes	SVG allow SVG fonts within SVG documentation. We can also apply CSS to SVG with font face property
Embedded OpenType Fonts (EOT)	EOT is used to develop the web pages and it has embedded in webpages so no need to allow 3rd party fonts

Fonts description

The following list contained all the fonts description which are placed in the @font-face rule:

Values	Description
font-family	Used to defines the name of font
src	Used to defines the URL
font-style	Used to defines the fonts style
font-weight	Used to defines the font weight(boldness)

Following code shows the sample code of font face

```
<html>
  <head>
    <style>
      @font-face {
        font-family: myFirstFont;
        src: url(/css/font/SansationLight.woff);
      }
      div { font-family: myFirstFont; }
    </Style>
  </head>
  <body>
    <div>This is the example of font face with CSS3.</div>
    <p><b>Original Text :</b>This is the example of font face with CSS3.</p>
  </body>
</html>
```

It will produce the following result:

This is the example of font face with CSS3.

Original Text : This is the example of font face with CSS3.

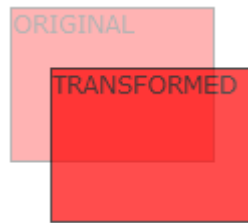
CSS3 - 2d Transforms

2D transforms are used to re-change the element structure as **translate, rotate, scale, and skew**

The following table has contained common methods which are used in 2D transforms

Values	Description
matrix(n,n,n,n,n,n)	Used to defines matrix transforms with six values
translate(x,y)	Used to transforms the element along with x-axis and y-axis
translateX(n)	Used to transforms the element along with x-axis
translateY(n)	Used to transforms the element along with y-axis
scale(x,y)	Used to change the width and height of element
scaleX(n)	Used to change the width of element
scaleY(n)	Used to change the height of element
rotate(angle)	Used to rotate the element based on an angle
skewX(angle)	Used to defines skew transforms along with x axis
skewY(angle)	Used to defines skew transforms along with y axis

The translate () Method



The **translate ()** method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

The following example moves the <div> element 50 pixels to the right, and 100 pixels down from its current position:

```
<html>
<head>
<style>
div {width: 300px; height: 100px; background-color: yellow; border: 1px solid black;
    transform: translate(50px,100px); /* Standard syntax */ }
</style>
</head>
<body>
<div>
The translate() method moves an element from its current position. This div element is
moved 50 pixels to the right, and 100 pixels down from its current position.
</div>
</body>
</html>
```

It will produce the following result :

The translate() method moves an element from its current position. This div element is moved 50 pixels to the right, and 100 pixels down from its current position.

The rotate () Method



The **rotate()** method rotates an element clockwise or counter-clockwise according to a given degree.

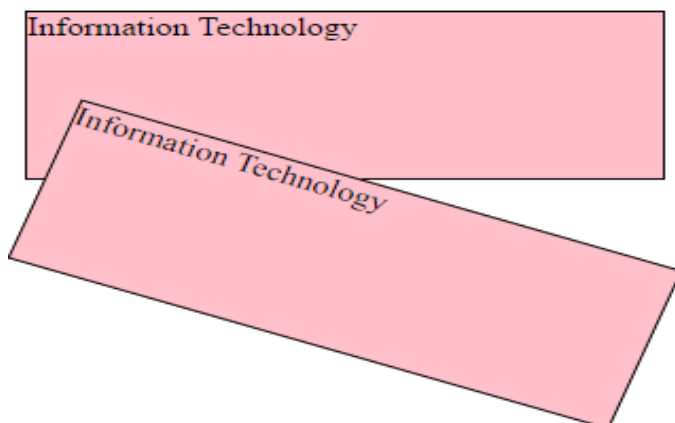
The following example rotates the <div> element clockwise with 20 degrees:

Rotate 20 degrees

Box rotation with 20 degrees angle as shown below

```
<html>
  <head>
    <style>
      div { width: 300px; height: 100px; background-color: pink; border: 1px solid
black; }
      div#myDiv { transform: rotate(20deg); }
    </style>
  </head>
  <body>
    <div> Information Technology </div>
    <div id="myDiv"> Information Technology </div>
  </body>
</html>
```

It will produce the following result –

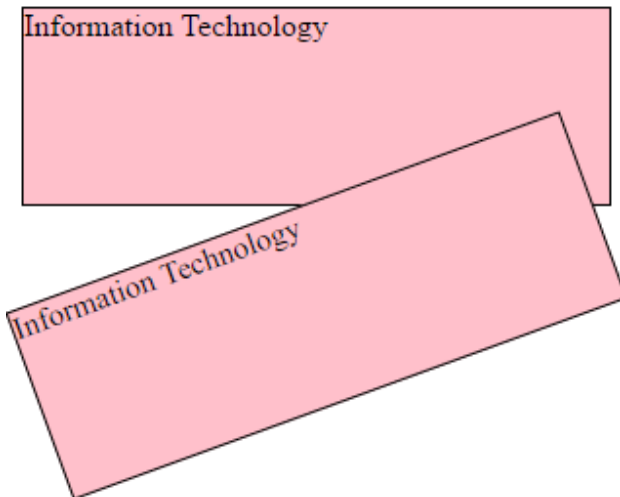


Rotate (-20) degrees

Box rotation with -20 degrees angle as shown below

```
div#myDiv { transform: rotate(-20deg); }
```

It will produce the following result –



The scale() Method



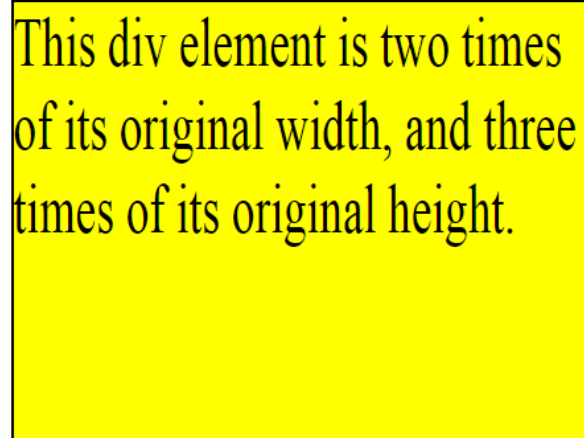
The `scale()` method increases or decreases the size of an element (according to the parameters given for the width and height).

The following example increases the `<div>` element to be two times of its original width, and three times of its original height:

```
<html>
<head>
<style>
  div { margin: 150px; width: 200px; height: 100px; background-color: yellow; border: 1px solid black;
    border: 1px solid black; transform: scale(2,3); /* Standard syntax */ }
</style>
</head>
<body>
<p>The scale() method increases or decreases the size of an element.</p>
<div>
This div element is two times of its original width, and three times of its original height.
</div>
</body>
</html>
```

It will produce the following result –

The scale() method increases or decreases the size of an element.



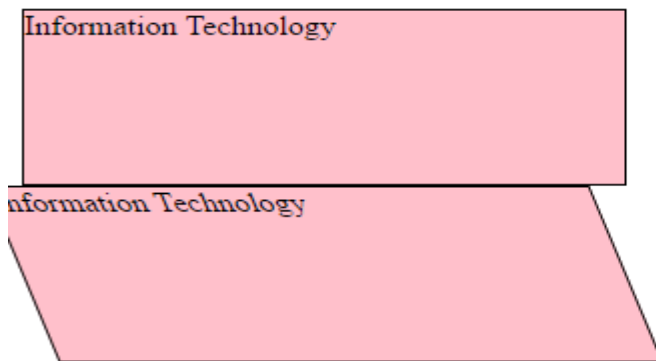
This div element is two times
of its original width, and three
times of its original height.

Skew X axis

Box rotation with skew x-axis as shown below

```
<html>
  <head>
    <style>
      div { width: 300px; height: 100px; background-color: pink; border: 1px solid black; }
      #skewDiv { transform: skewX(20deg); }
    </style>
  </head>
  <body>
    <div> Information Technology </div>
    <div id="skewDiv"> Information Technology </div>
  </body>
</html>
```

It will produce the following result –

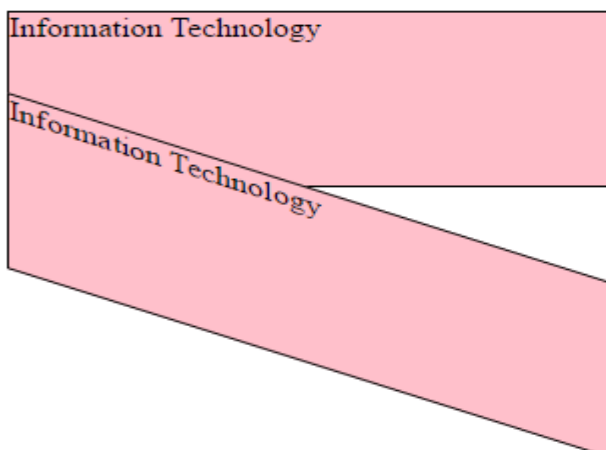


Skew Y axis

Box rotation with skew y-axis as shown below

```
#skewDiv {transform: skewY(20deg); }
```

It will produce the following result –



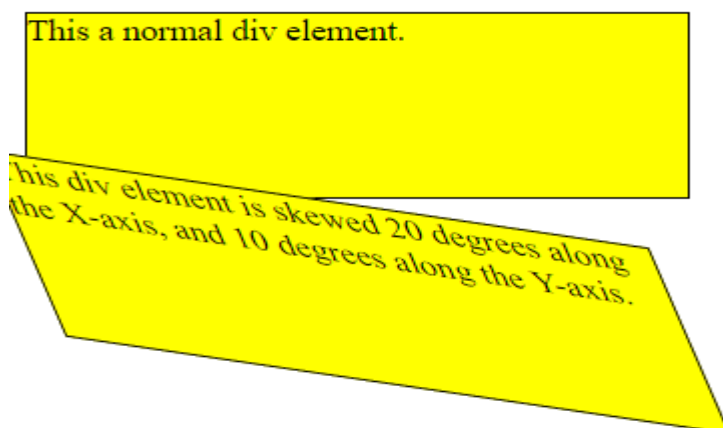
The skew() Method

The **skew()** method skews an element along the X and Y-axis by the given angles.

The following example skews the <div> element 20 degrees along the X-axis, and 10 degrees along the Y-axis:

```
<html>
<head>
<style>
div { width: 300px; height: 100px; background-color: yellow; border: 1px solid black;}
div#myDiv { transform: skew(20deg,10deg); /* Standard syntax */}
</style>
</head>
<body>
<p>The skew() method skews an element into a given angle.</p>
<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is skewed 20 degrees along the X-axis, and 10 degrees along the Y-axis.
</div>
</body>
</html>
```

The **skew()** method skews an element into a given angle.



The matrix() Method



The matrix() method combines all the 2D transform methods into one.

The matrix() method takes **six parameters**, containing mathematic functions, which allows you to **rotate**, **scale**, **move (translate)**, and **skew** elements.

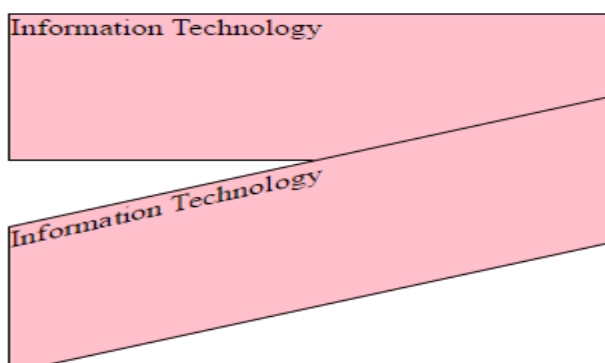
The parameters are as follow:

matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY()):

Box rotation with Matrix transforms as shown below

```
<html>
  <head>
    <style>
      div { width: 300px; height: 100px; background-color: pink; border: 1px solid black; }
      div#myDiv1 { transform: matrix(1, -0.3, 0, 1, 0, 0); }
    </style>
  </head>
  <body>
    <div> Information Technology </div>
    <div id="myDiv1"> Information Technology </div>
  </body>
</html>
```

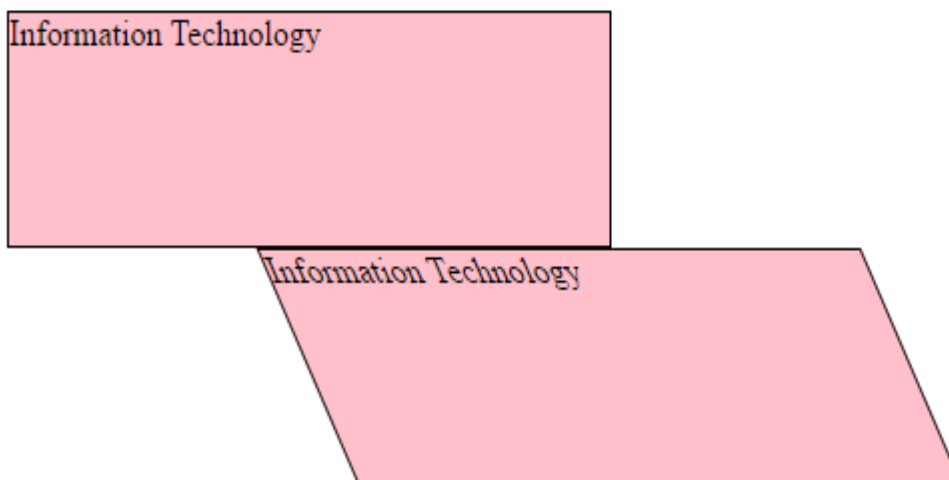
It will produce the following result:



Matrix transforms with another direction

```
<html>
  <head>
    <style>
      div {width: 300px; height: 100px; background-color: pink; border: 1px solid
        black;}
      div#myDiv2 {transform: matrix(1, 0, 0.5, 1, 150, 0); }
    </style>
  </head>
  <body>
    <div>Information Technology </div>
    <div id="myDiv2">Information Technology </div>
  </body>
</html>
```

It will produce the following result –



CSS3 - 3D transforms

Using with 3d transforms, we can move element to x-axis, y-axis and z-axis,

The following 3D transformation methods:

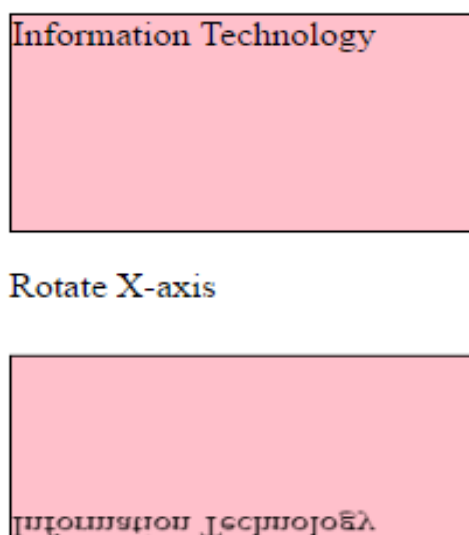
- rotateX()
- rotateY()

X-axis 3D transforms

The following an example shows the x-axis 3D transforms

```
<html>
<head>
  <style>
    div { width: 200px; height: 100px; background-color: pink; border: 1px solid black; }
    div#myDiv { transform: rotateX(150deg); }
  </style>
</head>
<body>
  <div> Information Technology </div>
  <p>Rotate X-axis</p>
  <div id="myDiv"> Information Technology </div>
</body>
</html>
```

It will produce the following result –

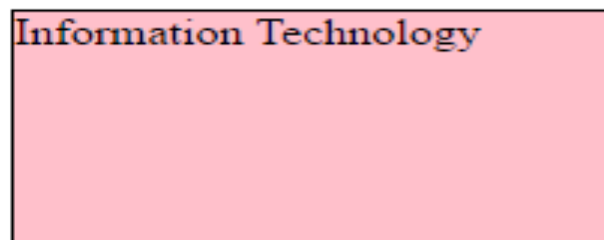


Y-axis 3D transforms

The following an example shows the y-axis 3D transforms

```
<html>
  <head>
    <style>
      div { width: 200px; height: 100px; background-color: pink; border: 1px solid black; }
      div#yDiv { transform: rotateY(150deg); }
    </style>
  </head>
  <body>
    <div>Information Technology </div>
    <p>Rotate Y axis</p>
    <div id="yDiv"> Information Technology </div>
  </body>
</html>
```

It will produce the following result:



Rotate Y-axis

