**CSS Animation**

**CSS Animation property** is used *to create animation on the webpage*. It can be used as a replacement of animation created by Flash and JavaScript.

**CSS3 @keyframes Rule**

The animation is created in the @keyframe rule. It is used to control the intermediate steps in a CSS animation sequence.

**What animation does**

An animation *makes an element change gradually* from one style to another. You can add as many as properties you want to add. You can also specify the changes in percentage.0% specify the start of the animation and 100% specify its completion.

**How CSS animation works**

When the animation is created in the @keyframe rule, it must be bound with selector; otherwise the animation will have no effect.

The animation could be bound to the selector by specifying at least these two properties:

* The name of the animation
* The duration of the animation

**CSS animation properties**

|  |  |
| --- | --- |
| **Property** | **Description** |
| @keyframes | It is used to specify the animation. |
| animation | This is a shorthand property, used for setting all the properties, except the animation-play-state and the animation-fill- mode property. |
| animation-delay | It specifies when the animation will start. |
| animation-direction | It specifies if or not the animation should play in reserve on alternate cycle. |
| animation-duration | It specifies the time duration taken by the animation to complete one cycle. |
| animation-fill-mode | it specifies the static style of the element. (when the animation is not playing) |
| animation-iteration-count | It specifies the number of times the animation should be played. |
| animation-play-state | It specifies if the animation is running or paused. |
| animation-name | It specifies the name of @keyframes animation. |
| animation-timing-function | It specifies the speed curve of the animation. |

**CSS animation example: changing background color**

Let's see a simple CSS animation example that changes background color of rectangle from RED to BLACK.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 100px;

    height: 100px;

    background: red;

    -webkit-animation: myfirst 6s; /\* Chrome, Safari, Opera \*/

    animation: myfirst 5s;

}

/\* Chrome, Safari, Opera \*/

@-webkit-keyframes myfirst {

    from {background: red;}

    to {background: green;}

}

/\* Standard syntax \*/

@keyframes myfirst {

    from {background: red;}

    to {background: green;}

}

**</style>**

**</head>**

**<body>**

**<p><b>**Note:**</b>** The IE 9 and earlier versions don't support CSS3 animation property. **</p>**

**<div></div>**

**</body>**

**</html>**

**CSS animation example: Moving Rectangle**

Let's take another example to display animation with percentage value.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 100px;

    height: 100px;

    background: red;

    position: relative;

    -webkit-animation: myfirst 5s; /\* Chrome, Safari, Opera \*/

    animation: myfirst 5s;

}

/\* Chrome, Safari, Opera \*/

@-webkit-keyframes myfirst {

    0%   {background:red; left:0px; top:0px;}

    25%  {background:yellow; left:300px; top:0px;}

    50%  {background:blue; left:200px; top:300px;}

    75%  {background:green; left:0px; top:200px;}

    100% {background:red; left:0px; top:0px;}

}

/\* Standard syntax \*/

@keyframes myfirst {

    0%   {background:red; left:0px; top:0px;}

    25%  {background:yellow; left:300px; top:0px;}

    50%  {background:blue; left:300px; top:200px;}

    75%  {background:green; left:0px; top:200px;}

    100% {background:red; left:0px; top:0px;}

}

**</style>**

**</head>**

**<body>**

**<p><b>**Note:**</b>** The Internet Explorer 9 and its earlier versions don't support this example.**</p>**

**<div></div>**

**</body>**

**</html>**

# CSS Gradient

CSS gradient is used to display smooth transition within two or more specified colors.

## Why CSS Gradient

These are the following reasons to use CSS gradient.

* You don't have to use images to display transition effects.
* The download time and bandwidth usage can also be reduced.
* It provides better look to the element when zoomed, because the gradient is generated by the browser.

There are two types of gradient in CSS3.

1. Linear gradients
2. Radial gradients

## 1) CSS Linear Gradient

The CSS3 linear gradient goes up/down/left/right and diagonally. To create a CSS3 linear gradient, you must have to define two or more color stops. The color stops are the colors which are used to create a smooth transition. Starting point and direction can also be added along with the gradient effect.

background: linear-gradient (direction, color-stop1, color-stop2.....);

## (i) CSS Linear Gradient: (Top to Bottom)

Top to Bottom Linear Gradient is the default linear gradient. Let's take an example of linear gradient that starts from top. It starts red and transitions to green.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

#grad1 {

    height: 100px;

    background: -webkit-linear-gradient(red, green); /\* For Safari 5.1 to 6.0 \*/

    background: -o-linear-gradient(red, green); /\* For Opera 11.1 to 12.0 \*/

    background: -moz-linear-gradient(red, green); /\* For Firefox 3.6 to 15 \*/

    background: linear-gradient(red, green); /\* Standard syntax (must be last) \*/

}

**</style>**

**</head>**

**<body>**

**<h3>**Linear Gradient - Top to Bottom**</h3>**

**<p>**This linear gradient starts at the top. It starts red, transitioning to green:**</p>**

**<div** id="grad1"**></div>**

**</body>**

**</html>**

Output:

### Linear Gradient - Top to Bottom

This linear gradient starts at the top. It starts red, transitioning to green:

## (ii) CSS Linear Gradient: (Left to Right)

The following example shows the linear gradient that starts from left and goes to right. It starts red from left side and transitioning to green.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

#grad1 {

    height: 100px;

    background: -webkit-linear-gradient(left, red, green); /\* For Safari 5.1 to 6.0 \*/

    background: -o-linear-gradient(right, red, green); /\* For Opera 11.1 to 12.0 \*/

    background: -moz-linear-gradient(right, red, green); /\* For Firefox 3.6 to 15 \*/

    background: linear-gradient(to right, red , green); /\* Standard syntax (must be last) \*/

}

**</style>**

**</head>**

**<body>**

**<h3>**Linear Gradient - Left to Right**</h3>**

**<p>**This linear gradient starts at the left. It starts red, transitioning to green:**</p>**

**<div** id="grad1"**></div>**

**</body>**

**</html>**

**Output:**

### Linear Gradient - Left to Right

This linear gradient starts at the left. It starts red, transitioning to green:

## (iii) CSS Linear Gradient: (Diagonal)

If you specify both the horizontal and vertical starting positions, you can make a linear gradient diagonal.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

#grad1 {

    height: 100px;

    background: -webkit-linear-gradient(left top, red , green); /\* For Safari 5.1 to 6.0 \*/

    background: -o-linear-gradient(bottom right, red, green); /\* For Opera 11.1 to 12.0 \*/

    background: -moz-linear-gradient(bottom right, red, green); /\* For Firefox 3.6 to 15 \*/

    background: linear-gradient(to bottom right, red , green); /\* Standard syntax (must be last) \*/

}

**</style>**

**</head>**

**<body>**

**<h3>**Linear Gradient - Diagonal**</h3>**

**<p>**This linear gradient starts at top left. It starts red, transitioning to green:**</p>**

**<div** id="grad1"**></div>**

**</body>**

**</html>**

**Output:**

### Linear Gradient - Diagonal

This linear gradient starts at top left. It starts red, transitioning to green:

## 2) CSS Radial Gradient

You must have to define at least two color stops to create a radial gradient. It is defined by its center.

background: radial-gradient(shape size at position, start-color, ..., last-color);

## (i) CSS Radial Gradient: (Evenly Spaced Color Stops)

Evenly spaced color stops is a by default radial gradient. Its by default shape is eclipse, size is farthest- carner, and position is center.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

#grad1 {

    height: 150px;

    width: 200px;

    background: -webkit-radial-gradient(blue, green, red); /\* Safari 5.1 to 6.0 \*/

    background: -o-radial-gradient(blue, green, red); /\* For Opera 11.6 to 12.0 \*/

    background: -moz-radial-gradient(blue, green, red); /\* For Firefox 3.6 to 15 \*/

    background: radial-gradient(blue, green, red); /\* Standard syntax (must be last) \*/

}

**</style>**

**</head>**

**<body>**

**<h3>**Radial Gradient - Evenly Spaced Color Stops**</h3>**

**<div** id="grad1"**></div>**

**</body>**

**</html>**

**Output:**

### Radial Gradient - Evenly Spaced Color Stops

## (ii) Radial Gradient: (Differently Spaced Color Stops)

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

#grad1 {

    height: 150px;

    width: 200px;

    background: -webkit-radial-gradient(blue 5%, green 15%, red 60%); /\* Safari 5.1 to 6.0 \*/

    background: -o-radial-gradient(blue 5%, green 15%, red 60%); /\* For Opera 11.6 to 12.0 \*/

    background: -moz-radial-gradient(blue 5%, green 15%, red 60%); /\* For Firefox 3.6 to 15 \*/

    background: radial-gradient(blue 5%, green 15%, red 60%); /\* Standard syntax (must be last) \*/

}

**</style>**

**</head>**

**<body>**

**<h3>**Radial Gradient - Differently Spaced Color Stops**</h3>**

**<div** id="grad1"**></div>**

**</body>**

**</html>**

Output:

### Radial Gradient - Differently Spaced Color Stops

## Supporting Browsers

This table specify the first browser version that fully supports the gradient property.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **property** | **chrome** | **firefox** | **ie** | **opera** | **safari** |
| linear-gradient | 26.0  10.0-webkit- | 16.0  3.6-moz- | 10.0 | 12.1  11.1-o- | 6.1  5.1-webkit- |
| radial-gradient | 26.0  10.0-webkit- | 16.0 3.6-moz- | 10.0 | 12.1 11.1-o- | 6.1 5.1-webkit- |
| repeating-linear-gradient | 26.0  10.0-webkit- | 16.0 3.6-moz- | 10.0 | 12.1 11.1-o- | 6.1 5.1-webkit- |
| repeating-radial-gradient | 26.0 10.0-webkit- | 16.0 3.6-moz- | 10.0 | 12.1 11.1-o- | 6.1 5.1-webkit- |

# CSS Transition

The CSS transitions are effects that are added to change the element gradually from one style to another, without using flash or JavaScript.

You should specify two things to create CSS transition.

* The CSS property on which you want to add an effect.
* The time duration of the effect.

Let's take an example which defines transition effect on width property and duration of 3 seconds.

#### Note: If you don't specify the duration part, the transition will have no effect because its default value is 0. The transition effect is started when you move cursor over an element.

#### Note: The transition property is not supported by IE9 and earlier version.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div{

    width: 100px;

    height: 100px;

    background: orange;

    -webkit-transition: width 1s; /\* For Safari 3.1 to 6.0 \*/

    transition: width 1s;

}

div:hover {

    width: 300px;

}

**</style>**

**</head>**

**<body>**

**<div></div>**

**<p>**Move the cursor over the div element above, to see the transition effect.**</p>**

**</body>**

**</html>**

#### Note: When you take mouse cursor out of the element, it gains its original style gradually.

## CSS Multiple Transition Effect

It is used to add transition effect for more than one CSS property. If you want to add transition effect on more than one property, separate those properties with a comma.

Let's take an example. Here, the transition effects on width, height and transformation.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    padding:15px;

    width: 150px;

    height: 100px;

    background: violet;

    -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /\* For Safari 3.1 to 6.0 \*/

    transition: width 2s, height 2s, transform 2s;

}

div:hover {

    width: 300px;

    height: 200px;

    -webkit-transform: rotate(360deg); /\* Chrome, Safari, Opera \*/

    transform: rotate(360deg);

}

**</style>**

**</head>**

**<body>**

**<div><b>**Be-practical.com**</b><p>** (Move your cursor on me to see the effect)**</p></div>**

**</body>**

**</html>**

**CSS Tooltips**

CSS Tooltips are a great way to display extra information about something when the user moves the mouse cursor over an element.

**Basic Tooltip Example**

Let's create basic tooltip that appears when the cursor of the mouse moves over an element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: red;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<p>**Move the mouse over the text below:**</p>**

**<div** class="tooltip"**>**Hover over me

**<span** class="tooltiptext"**>**This is tooltip text**</span>**

**</div>**

**</body>**

**</html>**

By using tooltips, you can display the position of the tooltip information in four ways:

* Top of the element
* Left side of the element
* Right side of the element
* Bottom of the element

**Top Tooltip**

The top tooltip specifies that if you move your mouse cursor over the element, the tooltip information will be displayed on the top of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: red;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    bottom: 100%;

    left: 50%;

    margin-left: -60px;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Top Tooltip Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>** Welcome to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology.**</span>**

**</div>**

**</body>**

**</html>**

**Bottom Tooltip**

The bottom tooltip specifies that if you move your mouse cursor over the element, the tooltip information will be displayed on the bottom of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}



.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: red;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    top: 100%;

    left: 50%;

    margin-left: -60px;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Bottom Tooltip Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>** Welcome to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology.**</span>**

**</div>**

**</body>**

**</html>**

**Left Tooltip**

The left tooltip specifies that if you move your mouse cursor over the element, the tooltip information will be displayed on the left side of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: red;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    top: -5px;

    right: 105%;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Left Tooltip Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>** Welcome to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology.**</span>**

**</div>**

**</body>**

**</html>**

**Right Tooltip**

The right tooltip specifies that if you move your mouse cursor over the element, the tooltip information will be displayed on the right side of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: red;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    top: -5px;

    left: 105%;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Right Tooltip Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>** Welcome to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology.**</span>**

**</div>**

**</body>**

**</html>**

**CSS Tooltip Animation/ Fade In Tooltips**

The CSS fade in tooltip or tooltip animation is used to fade in the tooltip text when it is about to visible. The CSS3 "transition" property along with "opacity" property is used o attain fade in tooltip or tooltip animation. The animation time from being completely invisible to 100% visible is specified in second.

Let's take an example to demonstrate the CSS tooltip animation. In this example, the fade in time is 5 second.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: red;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    bottom: 100%;

    left: 50%;

    margin-left: -60px;

    /\* Fade in tooltip - takes 1 second to go from 0% to 100% opac: \*/

    opacity: 0;

    transition: opacity 5s;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

    opacity: 1;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Fade In Tooltip Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>** Welcome to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology.**</span>**

**</div>**

**</body>**

**</html>**

**CSS Arrow**

The CSS arrow is used to add an array along with tooltip. It makes the tooltip like a speech bubble. It can also be represented in four ways:

* Top arrow
* Bottom arrow
* Left arrow
* Right arrow

**CSS Top Arrow**

The top arrow is used to add an arrow like structure on the top of the tooltip when you move your mouse cursor over the element. It will display the tooltip on the bottom of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: black;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    top: 150%;

    left: 50%;

    margin-left: -60px;

}

.tooltip .tooltiptext::after {

    content: "";

    position: absolute;

    bottom: 100%;

    left: 50%;

    margin-left: -5px;

    border-width: 5px;

    border-style: solid;

    border-color: transparent transparent black transparent;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Top Arrow Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>**Welcom to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology**</span>**

**</div>**

**</body>**

**</html>**

**CSS Bottom Arrow**

The bottom arrow is used to add an arrow like structure on the bottom of the tooltip when you move your mouse cursor over the element. It will display the tooltip on the top of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: black;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    bottom: 150%;

    left: 50%;

    margin-left: -60px;

}

.tooltip .tooltiptext::after {

    content: "";

    position: absolute;

    top: 100%;

    left: 50%;

    margin-left: -5px;

    border-width: 5px;

    border-style: solid;

    border-color: black transparent transparent transparent;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Bottom Arrow Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>**Welcom to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology**</span>**

**</div>**

**</body>**

**</html>**

**CSS Left Arrow**

The left arrow is used to add an arrow like structure on the left of the tooltip when you move your mouse cursor over the element. It will display the tooltip on the right of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: black;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    top: -5px;

    left: 110%;

}

.tooltip .tooltiptext::after {

    content: "";

    position: absolute;

    top: 50%;

    right: 100%;

    margin-top: -5px;

    border-width: 5px;

    border-style: solid;

    border-color: transparent black transparent transparent;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Left Arrow Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>**Welcom to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology**</span>**

**</div>**

**</body>**

**</html>**

**Right Arrow**

The right arrow is used to add an arrow like structure on the right of the tooltip when you move your mouse cursor over the element. It will display the tooltip on the left of the element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style>**

.tooltip {

    position: relative;

    display: inline-block;

    border-bottom: 1px dotted black;

}

.tooltip .tooltiptext {

    visibility: hidden;

    width: 120px;

    background-color: black;

    color: #fff;

    text-align: center;

    border-radius: 6px;

    padding: 5px 0;

    position: absolute;

    z-index: 1;

    top: -5px;

    right: 110%;

}

.tooltip .tooltiptext::after {

    content: "";

    position: absolute;

    top: 50%;

    left: 100%;

    margin-top: -5px;

    border-width: 5px;

    border-style: solid;

    border-color: transparent transparent transparent black;

}

.tooltip:hover .tooltiptext {

    visibility: visible;

}

**</style>**

**<body** style="text-align:center;"**>**

**<h2>**Right Arrow Example**</h2>**

**<p>**Move your mouse cursor over the below heading**</p>**

**<div** class="tooltip"**><strong>**Welcom to Be-practical**</strong>**

**<span** class="tooltiptext"**>**A solution of all technology**</span>**

**</div>**

**</body>**

**</html>**

**CSS Flexbox**

CSS3 Flexible boxes also known as CSS Flexbox, is a new layout mode in CSS3.

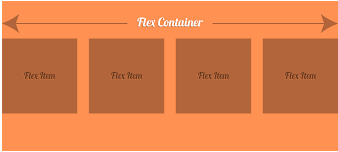
The CSS3 flexbox is used to make the elements behave predictably when they are used with different screen sizes and different display devices. It provides a more efficient way to layout, align and distribute space among items in the container.

It is mainly used to make CSS3 capable to change its item?s width and height to best fit for all available spaces. It is preferred over block model.

The CSS3 flexbox contains flex containers and flex items.

**Flex container:**The flex container specifies the properties of the parent. It is declared by setting the display property of an element to either flex or inline-flex.

**Flex items:**The flex items specify properties of the children. There may be one or more flex items inside a flex container.



**Note:**Flexbox specifies how flex items are set inside a flex container. It sets the flex items inside a flex container along a flex line. By default, there is only one flex line per flex container. Everything outside a flex container and inside a flex item is considered as usual.

Let's take an example to show three flex items within a flex container. By default, they are set along the horizontal flex line, from left to right:

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<meta** http-equiv="Content-Type"  content="text/html; charset=windows-1252"**>**

**<style>**

.flex-container {

    display: -webkit-flex;

    display: flex;

    width: 400px;

    height: 200px;

    background-color: lightpink;

}

.flex-item {

    background-color: brown;

    width: 100px;

    height: 100px;

    margin: 10px;

}

**</style>**

**</head>**

**<body>**

**<div** class="flex-container"**>**

**<div** class="flex-item"**>**flex item 1**</div>**

**<div** class="flex-item"**>**flex item 2**</div>**

**<div** class="flex-item"**>**flex item 3**</div>**

**</div>**

**</body>**

**</html>**

You can also change the direction of the flex line by using direction property. If you want to set the direction line right-to-left then set direction property to rtl.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

    direction: rtl;

}

.flex-container {

    display: -webkit-flex;

    display: flex;

    width: 400px;

    height: 200px;

    background-color: lightpink;

}

.flex-item {

    background-color: brown;

    width: 100px;

    height: 100px;

    margin: 10px;

}

**</style>**

**</head>**

**<body>**

**<div** class="flex-container"**>**

**<div** class="flex-item"**>**flex item 1**</div>**

**<div** class="flex-item"**>**flex item 2**</div>**

**<div** class="flex-item"**>**flex item 3**</div>**

**</div>**

**</body>**

**</html>**

**CSS3 Flexbox Properties**

Following is a list of CSS3 Flexbox properties:

|  |  |
| --- | --- |
| **property** | **description** |
| display | it is used to specify the type of box used for an html element. |
| flex-direction | it is used to specify the direction of the flexible items inside a flex container. |
| justify-content | it is used to align the flex items horizontally when the items do not use all available space on the main-axis. |
| align-items | it is used to align the flex items vertically when the items do not use all available space on the cross-axis. |
| flex-wrap | it specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line. |
| align-content | it is used to modify the behavior of the flex-wrap property. it is similar to align-items, but instead of aligning flex items, it aligns flex lines. |
| flex-flow | it specifies a shorthand property for flex-direction and flex-wrap. |
| order | it specifies the order of a flexible item relative to the rest of the flex items inside the same container. |
| align-self | it is used on flex items. it overrides the container's align-items property. |
| flex | it specifies the length of a flex item, relative to the rest of the flex items inside the same container. |

Supporting Browsers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **chrome browser Chrome** | **ie browser IE** | **firefox browser Firefox** | **opera browser Opera** | **safari browser Safari** |
| basic support (single-line flexbox) | 29.0 21.0 -webkit- | 11.0 | 22.0 18.0 -moz- | 12.1 -webkit- | 6.1 -webkit- |
| multi-line flexbox | 29.021.0 -webkit | 11.0 | 28.0 | 17.0 15.0 -webkit- 12.1 | 6.1 -webkit- |

**CSS Media Queries**

**What is Media Query?**

CSS Media query is a W3C recommendation and a CSS3 module which is used to adapt to conditions such as screen resolution (e.g. Smartphone screen vs. computer screen).

* The media query technique first used in CSS3.
* It became a W3C recommendation in June 2012.
* It is an extension of media dependent stylesheets used in different media types (i.e. screen and print) found in CSS2.
* The most commonly used media feature is "width".
* It uses the @media rule to include a block of CSS properties only if a certain condition is true.

**Recommended Media features for Media queries:**

Following is a list of media features recommended for media queries by W3C.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Value** | **Min/Max** | **Description** |
| color | integer | yes | It specifies the number of bits per color component. |
| color-index | integer | yes | It specifies the number of entries in the color lookup table. |
| device-aspect-ratio | integer/integer | yes | It specifies the aspect ratio of the device. |
| device-height | length | yes | It specifies the height of the output device. |
| device-width | length | yes | It specifies the width of the output device. |
| grid | integer | no | It is true for a grid-based device. |
| height | length | yes | It specifies the height of the rendering surface. |
| monochrome | integer | yes | It specifies the number of bits per pixel in a monochrome frame buffer. |
| resolution | resolution ("dpi" or "dpcm") | yes | It specifies the resolution of the display screen. |
| scan | "progressive" or "interlaced" | no | It specifies scanning process of "tv" media types. |
| width | length | yes | It specifies the width of the rendering surface. |

**What is Responsive Web Design?**

The term Responsive Web Design was given by Ethan Marcotte. It facilitates you to use fluid grids, flexible images, and media queries to progressively enhance a web page for different viewing contexts i.e. Desktop, Smartphone, Tablet etc.

**What screen resolutions do you use while taking screenshots?**

Smartphone: 320px

Tablet: 768px

Netbook: 1024px

Desktop: 1600px

Let's take an example to see the simple use of media query:

This example specifies that if you resize your window less than 500px, the background color will be changed.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<meta** name="viewport" content="width=device-width, initial-scale=1.0"**/>**

**<style>**

body {

    background-color:yellow;

}

@media only screen and (max-width: 500px) {

    body {

        background-color:green;

    }

}

**</style>**

**</head>**

**<body>**

**<p>**If you resize the browser window and the width of this document is

less than 500 pixels, the background-

color is "green", otherwise it is "yellow"**</p>**

**</body>**

**</html>**

**How to add a breakpoint?**

Media query can be used to create a responsive webpage. The breakpoint is used on web pages where you want that certain parts of the design will behave differently on each side of the breakpoint.

**Let's take an example:**

Here we use a media query to add a breakpoint at 768px.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<meta** name="viewport" content="width=device-width, initial-scale=1.0"**>**

**<style>**

\* {

    box-sizing: border-box;

}

.row:after {

    content: "";

    clear: both;

    display: block;

}

[class\*="col-"] {

    float: left;

    padding: 15px;

}

html {

    font-family: "Lucida Sans", sans-serif;

}

.header {

    background-color: purple;

    color: pink;

    padding: 15px;

}

.menu ul {

    list-style-type: none;

    margin: 0;

    padding: 0;

}

.menu li {

    padding: 8px;

    margin-bottom: 7px;

    background-color :green;

    color: #ffffff;

    box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24);

}

.menu li:hover {

    background-color: #0099cc;

}

.aside {

    background-color: green;

    padding: 15px;

    color: #ffffff;

    text-align: center;

    font-size: 14px;

    box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24);

}

.footer {

    background-color: #0099cc;

    color: #ffffff;

    text-align: center;

    font-size: 12px;

    padding: 15px;

}

/\* For mobile phones: \*/

[class\*="col-"] {

    width: 100%;

}

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

    /\* For tablets: \*/

    .col-m-1 {width: 8.33%;}

    .col-m-2 {width: 16.66%;}

    .col-m-3 {width: 25%;}

    .col-m-4 {width: 33.33%;}

    .col-m-5 {width: 41.66%;}

    .col-m-6 {width: 50%;}

    .col-m-7 {width: 58.33%;}

    .col-m-8 {width: 66.66%;}

    .col-m-9 {width: 75%;}

    .col-m-10 {width: 83.33%;}

    .col-m-11 {width: 91.66%;}

    .col-m-12 {width: 100%;}

}

@media only screen and (min-width: 768px) {

    /\* For desktop: \*/

    .col-1 {width: 8.33%;}

    .col-2 {width: 16.66%;}

    .col-3 {width: 25%;}

    .col-4 {width: 33.33%;}

    .col-5 {width: 41.66%;}

    .col-6 {width: 50%;}

    .col-7 {width: 58.33%;}

    .col-8 {width: 66.66%;}

    .col-9 {width: 75%;}

    .col-10 {width: 83.33%;}

    .col-11 {width: 91.66%;}

    .col-12 {width: 100%;}

}

**</style>**

**</head>**

**<body>**

**<div** class="header"**>**

**<h1>**Be-practical**</h1>**

**</div>**

**<div** class="row"**>**

**<div** class="col-3 col-m-3 menu"**>**

**<ul>**

**<li>**C/C++**</li>**

**<li>**Java**</li>**

**<li>**SQL**</li>**

**<li>**Android**</li>**

**<li>**HTML**</li>**

**<li>**CSS**</li>**

**<li>**Cloud Computing**</li>**

**<li>**PHP**</li>**

**<li>**JSON**</li>**

**<li>**JQuery**</li>**

**<li>**MongoDB**</li>**

**<li>**Oracle**</li>**

**</ul>**

**</div>**

**<div** class="col-6 col-m-9"**>**

**<h1>**About Be-practical**</h1>**

**<p>**Be-practical is written and developed that students may learn

computer science related technologies easily.**</p>**

**</div>**

**<div** class="col-3 col-m-12"**>**

**<div** class="aside"**>**

**<h2>**What is Be-practical?**</h2>**

**<p>**Be-practical is the No.1 Java training institute in Noida, Delhi, Gurgaon, Ghaziabad and Faridabad.  You will get practical training on Java by our Java expert who have 7+ year industrial experience.**</p>**

**<h2>**Why Be-practical?**</h2>**

**<p>**Life Time Validity, Training by Java Professionals, Problem Solving Team, Project Development, Small Batches to focus on each student  **</p>**

**<h2>**How to reach?**</h2>**

**<p>**Be-practical is located in Noida (Gautam Budhh Nagar). The full address is G-13 Second  Floor Sector 3 (Near Sector-16 Metro Station)  Noida(U.P)**</p>**

**</div>**

**</div>**

**</div>**

**<div** class="footer"**>**

**<p>**Resize the browser window to see how the content respond to the resizing. **</p>**

**</div>**

**</body>**

**</html>**

**CSS Transforms**

CSS3 supports transform property. This transform property facilitates you to translate, rotate, scale, and skews elements.

Transformation is an effect that is used to change shape, size and position.

There are two type of transformation i.e. 2D and 3D transformation supported in CSS3.

**CSS 2D Transforms**

The CSS 2D transforms are used to re-change the structure of the element as translate, rotate, scale and skew etc.

Following is a list of 2D transforms methods:

* **translate(x,y):** It is used to transform the element along X-axis and Y-axis.
* **translateX(n):** It is used to transform the element along X-axis.
* **translateY(n):** It is used to transform the element along Y-axis.
* **rotate():** It is used to rotate the element on the basis of an angle.
* **scale(x,y):** It is used to change the width and height of an element.
* **scaleX(n):** It is used to change the width of an element.
* **scaleY(n):** It is used to change the height of an element.
* **skewX():** It specifies the skew transforms along with X-axis.
* **skewY():**It specifies the skew transforms along with Y-axis.
* **matrix():** It specifies matrix transforms.

**Supporting browsers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **chrome browser Chrome** | **ie browser IE** | **firefox browser Firefox** | **opera browser Opera** | **safari browser Safari** |
| transform | 36.04.0 -webkit- | 10.09.0 -ms- | 16.03.5 -moz- | 23.0  15.0 -webkit-  12.1  10.5 -o- | 9.03.2 -webkit- |
| transform-origin  (two-value syntax) | 36.0  4.0 -webkit- | 10.0  9.0 -ms- | 16.03.5 -moz- | 9.03.2 -webkit- | 23.0  15.0 -webkit-  12.110.5 -o- |

**The translate() method**

The CSS translate() method is used to move an element from its current position according to the given parameters i.e. X-axis and Y-axis.

Let's take an example to translate a

element 50 pixels right, and 100 pixels down from its current position.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<meta** http-equiv="Content-Type" content="text/html; charset=windows-1252"**>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: lightgreen;

    border: 1px solid black;

    -ms-transform: translate(100px,80px); /\* IE 9 \*/

    -webkit-transform: translate(100px,80px); /\* Safari \*/

    transform: translate(100px,80px); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

This div element is translated 50 pixels right, and 100 pixels down from its current position by using translate () method.

**</div>**

**</body>**

**</html>**

**The rotate() method**

The CSS rotate() method is used to rotate an element clockwise or anti-clockwise according to the given degree.

Let's take an example to rotate a

element by 300.

See this example:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: lightpink;

    border: 1px solid black;

}

div#myDiv {

    -ms-transform: rotate(30deg); /\* IE 9 \*/

    -webkit-transform: rotate(30deg); /\* Safari \*/

    transform: rotate(30deg); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

This a normal div element.

**</div>**

**<div** id="myDiv"**>**

This is the 30 degree clockwise rotated div element by using rotate() method.

**</div>**

**</body>**

**</html>**

**The scale() method**

The CSS scale() method is used to increase or decrease the size of an element according to the given width and height.

Let's take an example to increase the size of an

element by increasing its width and height two times.

strong>See this example:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    margin: 150px;

    width: 200px;

    height: 100px;

    background-color: lightpink;

    border: 1px solid black;

    border: 1px solid black;

    -ms-transform: scale(2.5,2); /\* IE 9 \*/

    -webkit-transform: scale(2.5,2); /\* Safari \*/

    transform: scale(2.5,2); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

This div element is scaled 2.5 times of its original width, and 2 times of its original height.

**</div>**

**</body>**

**</html>**

**The skewX() method**

The CSS skewX() method is used to skew an element along the X axis according to the given angle. Let?s take an example to skew an

element 30 degrees along the X-axis.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: lightpink;

    border: 1px solid black;

}

div#myDiv {

    -ms-transform: skewX(30deg); /\* IE 9 \*/

    -webkit-transform: skewX(30deg); /\* Safari \*/

    transform: skewX(30deg); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

This a normal div element.

**</div>**

**<div** id="myDiv"**>**

This div element is skewed 30 degrees along the X-axis.

**</div>**

**</body>**

**</html>**

**The skewY() method**

The CSS skewY() method is used to skew an element along the Y axis according to the given angle. Let's take an example to skew an

element 30 degrees along the Y-axis.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: lightpink;

    border: 1px solid black;

}

div#myDiv {

    -ms-transform: skewY(30deg); /\* IE 9 \*/

    -webkit-transform: skewY(30deg); /\* Safari \*/

    transform: skewY(30deg); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

This a normal div element.

**</div>**

**<div** id="myDiv"**>**

This div element is skewed 30 degrees along the Y-axis.

**</div>**

**</body>**

**</html>**

**The skew() method**

The CSS skew() method is used to skew an element along with X-axis and Y-axis according to the given angle.

Let's take a <div> element and skew it 30 degree along the X-axis and 20 degree along the Y-axis.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: lightpink;

    border: 1px solid black;

}

div#myDiv {

    -ms-transform: skew(30deg,20deg); /\* IE 9 \*/

    -webkit-transform: skew(30deg,20deg); /\* Safari \*/

    transform: skew(30deg,20deg); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

This a normal div element.

**</div>**

**<div** id="myDiv"**>**

This div element is skewed 30 degrees along the X-axis, and 20 degrees along the Y-axis.

**</div>**

**</body>**

**</html>**

**The matrix() method**

The CSS matrix() method combines all the six 2D transform methods altogether. It allows you to rotate, scale, translate, and skew elements.

**Syntax:**

The parameters of matrix method:

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

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: lightpink;

    border: 1px solid black;

}

div#myDiv1 {

    -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /\* IE 9 \*/

    -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /\* Safari \*/

    transform: matrix(1, -0.3, 0, 1, 0, 0); /\* Standard syntax \*/

}

div#myDiv2 {

    -ms-transform: matrix(1, 0, 0.5, 1, 150, 0); /\* IE 9 \*/

    -webkit-transform: matrix(1, 0, 0.5, 1, 150, 0); /\* Safari \*/

    transform: matrix(1, 0, 0.5, 1, 150, 0); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<p>**The matrix() method combines all the 2D transform methods into one.**</p>**

**<div>**

This  is a normal div element.

**</div>**

**<div** id="myDiv1"**>**

Using the matrix() method.

**</div>**

**<div** id="myDiv2"**>**

Another use of the matrix() method.

**</div>**

**</body>**

**</html>**

**CSS 3D Transforms**

The CSS 3D transforms facilitates you to move an element to X-axis, Y-axis and Z-axis. Following is a list of 3D transforms methods:

|  |  |
| --- | --- |
| **Function** | **Description** |
| matrix3D(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n) | It specifies a 3D transformation, using a 4x4 matrix of 16 values. |
| translate3D(x,y,z) | It specifies a 3D translation. |
| translateX(x) | It specifies 3D translation, using only the value for the X-axis. |
| translateY(y) | It specifies 3D translation, using only the value for the Y-axis. |
| translateZ(z) | It specifies 3D translation, using only the value for the Z-axis. |
| scale3D(x,y,z) | It specifies 3D scale transformation |
| scaleX(x) | It specifies 3D scale transformation by giving a value for the X-axis. |
| scaley(y) | It specifies 3D scale transformation by giving a value for the Y-axis. |
| scaleZ(z) | It specifies 3D scale transformation by giving a value for the Z-axis. |
| rotate3D(X,Y,Z,angle) | It specifies 3D rotation along with X-axis, Y-axis and Z-axis. |
| rotateX(angle) | It specifies 3D rotation along with X-axis. |
| rotateY(angle) | It specifies 3D rotation along with Y-axis. |
| rotateZ(angle) | It specifies 3D rotation along with Z-axis. |
| perspective(n) | It specifies a perspective view for a 3D transformed element. |

**Supporting Browsers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **chrome browser Chrome** | **ie browser IE** | **firefox browser Firefox** | **opera browser Opera** | **safari browser Safari** |
| transform | 36.012.0 -webkit- | 10.0 | 16.0  10.0 -moz- | 23.015.0 -webkit- | 9.04.0 -webkit- |
| transform-origin  (three-value syntax) | 36.012.0 -webkit- | 10.0 | 16.010.0 -moz- | 23.015.0 -webkit- | 9.04.0 -webkit- |
| transform-style | 36.012.0 -webkit- | 11.0 | 16.010.0 -moz- | 23.015.0 -webkit- | 9.04.0 -webkit- |
| perspective | 36.012.0 -webkit- | 10.0 | 16.010.0 -moz- | 23.015.0 -webkit- | 9.04.0 -webkit- |
| perspective-origin | 36.012.0 -webkit- | 10.0 | 16.010.0 -moz- | 23.015.0 -webkit- | 9.04.0 -webkit- |
| backface-visibility | 36.0  12.0 -webkit- | 10.0 | 16.010.0 -moz- | 23.015.0 -webkit- | 9.04.0 -webkit- |

**The 3D rotateX() method (X-axis rotation)**

The CSS 3D rotateX() method is used to rotate an element around its X-axis according to the given degree.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: lightpink;

    border: 1px solid black;

}

div#myDiv {

    -webkit-transform: rotateX(150deg); /\* Safari \*/

    transform: rotateX(150deg); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

This is Be-practical!

**</div>**

**<div** id="myDiv"**>**

This is Be-practical!

**</div>**

**<p><b>**Note:**</b>** Internet Explorer 9 (and earlier versions) does not support the rotateX() method.**</p>**

**</body>**

**</html>**

**The 3D rotateY() method (Y-axis rotation)**

The CSS 3D rotateY() method is used to rotate an element around its Y-axis according to the given degree.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color:lightpink;

    border: 1px solid black;

}

div#myDiv {

    -webkit-transform: rotateY(150deg); /\* Safari \*/

    transform: rotateY(150deg); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

Welcome to Be-practical!.

**</div>**

**<div** id="myDiv"**>**

Welcome to Be-practical!.

**</div>**

**<p><b>**Note:**</b>** Internet Explorer 9 (and earlier versions) does not support the rotateY() method.**</p>**

**</body>**

**</html>**

**The 3D rotateZ() method (Z-axis rotation)**

The CSS 3D rotateZ() method is used to rotate an element around its Z-axis according to the given degree.

**See this example:**>

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    width: 300px;

    height: 100px;

    background-color: yellow;

    border: 1px solid black;

}

div#myDiv {

    -webkit-transform: rotateZ(90deg); /\* Safari \*/

    transform: rotateZ(90deg); /\* Standard syntax \*/

}

**</style>**

**</head>**

**<body>**

**<div>**

Welcome to Be-practical!

**</div>**

**<div** id="myDiv"**>**

Welcome to Be-practical!

**</div>**

**<p><b>**Note:**</b>** Internet Explorer 9 (and earlier versions) does not support the rotateZ() method.**</p>**

**</body>**

**</html>**

**CSS Aural Media/ Style Sheets**

CSS Aural Media or CSS Style Sheets are used to attach specific sound style features to specific document elements. It uses speech synthesis and sound effect to facilitate users to listen to information instead of reading them. So, it is very useful for visually impaired people.

Aural media can be used in following:

* Used by blind or visually impaired people
* Help users learning to read and right pronunciation
* Training
* Help users who have reading problems
* Facilitates web access in vehicles
* Home entertainment
* Used by print-impaired communities
* Medical documentation
* Industrial documentation

**Aural Media Properties**

Following is a list of aural media properties:

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **property** | **explanation** | **value** |
| azimuth | it is set where the sound should come from. | angle  left-side  far-left  left  center-left  center  center-right  right  far-right  right-side  behind  leftwardsrightwards |
| cue | it is used to set the cue properties in one declaration. | cue-beforecue-after |
| cue-after | it is used to specify a sound to be played after speaking an element's content. | none  url |
| cue-before | it is used to specify a sound to be played before speaking an element's content. | noneurl |
| elevation | it is set where the sound should come from. | angle  below  level  above  higherlower |
| pause | it is used to set the pause properties in one declaration. | pause-beforepause-after |
| pause-after | it is used to specify a pause after speaking an element's content. | time  % |
| pause-before | it is used to specify a pause before speaking an element's content. | time  % |
| pitch | it is used to specify the pitch of the speaking voice. | frequency  x-low  low  medium  high  x-high |
| pitch-range | it is used to specify the variation of the speaking voice. it specifies whether he speaking voice is monotone voice or animated voice. | number |
| play-during | it specifies a sound to be played while speaking an element's content. | auto  none  url  mix  repeat |
| richness | it is used to specifies the sound quality of the speaking voice. it specifies that the voice is rich or thin. | number |
| speak | it specifies whether content will render aurally. | normal  nonespell-out |
| speak-header | it is used to specify how to handle table headers. it also specifies whether the headers be spoken before every cell, or only before a cell with a different header than the previous cell. | always  once |
| speak-numeral | it specifies how to speak numbers. | digits  continuous |
| speak-punctuation | it is used to specify how to speak punctuation characters. | nonecode |
| speech-rate | it specifies the speed of the speaking. | number  x-slow  slow  medium  fast  x-fast  fasterslower |
| stress | it is used to specify the "stress" in the speaking voice. | number |
| voice-family | it specifies the voice family of the speaking voice. | specific-voice  generic-voice |
| volume | it specifies the volume of the speaking voice. | number  %  silent  x-soft  soft  medium  loud  x-loud |

**The Aural Media Azimuth Property**

The Aural Media Azimuth Property describes the radial position (360 degree measure) on this listening plane. Its soundstage is considered to be parallel to the floor means it sets where the sound should come from horizontally.

The possible values of azimuth property are:

* **angle:** It is used to describe a position in terms of an angle within the range -360deg to 360deg. The value 0deg means directly ahead in the center of the sound stage. 90deg is to the right, 180deg behind, and 270deg (or, equivalently and more conveniently, -90deg) to the left.
* **left-side:** It is same as '270deg'. With 'behind', '270deg'.
* **far-left :** It is same as '300deg'. With 'behind', '240deg'.
* **left :** It is same as '320deg'. With 'behind', '220deg'.
* **center-left :** It is same as '340deg'. With 'behind', '200deg'.
* **center :** It is same as '0deg'. With 'behind', '180deg'.
* **center-right:** It is same as '20deg'. With 'behind', '160deg'.
* **right:** It is same as '40deg'. With 'behind', '140deg'.
* **far-right :**It is same as '60deg'. With 'behind', '120deg'.
* **right-side:**It is same as '90deg'. With 'behind', '90deg'.
* **Leftwards:**It is used to move the sound to the left and relative to the current angle. More precisely, subtracts 20 degrees.
* **Rightwards:**It is used to move the sound to the right, relative to the current angle. More precisely, adds 20 degrees.

**Syntax:**

**<style** tyle="text/css"**>**

<!--

h1   { azimuth: 30deg }

td.a { azimuth: far-right }          /\*  60deg \*/

#12  { azimuth: behind far-right }   /\* 120deg \*/

p.comment { azimuth: behind }        /\* 180deg \*/

--**>**

**</style>**

**The Aural Media Cue Property**

The CSS aural media cue property acts as shorthand for setting cue-before and cue-after. If two values are given, the first value is cue-before and the second is cue-after. If only one value is given, it applies to both properties.

The following two rules are equivalent:

**<style** tyle="text/css"**>**

   <!--

   h1 {cue-before: url("pop.au"); cue-after: url("pop.au") }

   h1 {cue: url("pop.au") }

   --**>**

**</style>**

**The Aural Media cue-before Property**

The CSS aural media cue-before property specifies a sound to be played before speaking an element's content to delimit it from other.

Its possible values are:

* **url:**It specifies the URL of a sound file to be played.
* **none:**It specifies that nothing has to be played.

**Syntax:**

**<style** tyle="text/css"**>**

   <!--

   a {cue-before: url("bell.aiff");}

   h1 {cue-before: url("pop.au"); }

   --**>**

**</style>**

**The Aural Media cue-after Property**

The aural media cue-after property specifies a sound to be played before speaking an element's content to delimit it from other.

Its possible values are:

* **url:**It specifies the URL of a sound file to be played.
* **none:**It specifies that nothing has to be played.

Syntax:

**<style** tyle="text/css"**>**

   <!--

   a {cue-after: url("dong.wav");}

   h1 {cue-after: url("pop.au"); }

   --**>**

**</style>**

**The Aural Media Elevation Property**

The CSS aural media elevation property sets where the sound should come from vertically. Its possible values are:

* **angle:**It specifies the elevation as an angle, between -90deg and 90deg. 0deg means on the forward horizon, which loosely means level with the listener. 90deg means directly overhead and -90deg means directly below.
* **below:**It is same as '-90deg'.
* **level:**It is same as '0deg'.
* **above:**It is same as '90deg'.
* **higher:**It adds 10 degrees to the current elevation.
* **lower:**It subtracts 10 degrees from the current elevation.

**Syntax:**

**<style** tyle="text/css"**>**

   <!--

   h1   { elevation: above }

   tr.a { elevation: 60deg }

   tr.b { elevation: 30deg }

   tr.c { elevation: level }

   --**>**

**</style>**

**The Aural Media pause-before Property**

The CSS aural media pause-before property specifies a pause to be observed before speaking an element's content.

Its possible values are:

* **time:**It is used to specify the pause in absolute time units. The time's unit is seconds and milliseconds.
* **percentage:**It specifies the inverse of the value of the speech-rate property. For example, if the speech-rate is 120 words per minute (i.e. a word takes half a second, or 500ms), then a pause-before of 100% means a pause of 500 ms and a pause-before of 20% means 100ms.

**The Aural Media pause-after Property**

The CSS aural media pause-before property specifies a pause to be observed after speaking an element's content.

Its possible values are:

* **time:**It is used to specifies the pause in absolute time units (seconds and milliseconds).
* **percentage:**It specifies the inverse of the value of the speech-rate property. For example, if the speech-rate is 120 words per minute (i.e. a word takes half a second, or 500ms), then a pause-after of 100% means a pause of 500 ms and a pause-after of 20% means 100ms.

**CSS User Interface**

CSS provides many user interface features like resizing elements, outlines and box sizing.

Following is a list of some common properties of CSS3 user interface:

|  |  |
| --- | --- |
| **Values** | **Description** |
| appearance | It facilitates users to make elements as user interface elements. |
| box-sizing | It facilitates users to fix elements on area in clear way. |
| icon | It is used to provide the icon on area. |
| resize | It is used to resize elements which are on area. |
| outline-offset | It is used to set space between an outline and the edge or border of an element. |
| nav-down | It is used to move down while pressing the down arrow button in keypad. |
| nav-left | It is used to move left while pressing the left arrow button in keypad. |
| nav-right | It is used to move right while pressing the right arrow button in keypad. |
| nav-up | It is used to move up while pressing the up arrow button in keypad. |

**Note:**resize and outline-offset are the most important properties of the CSS user interface. Resize property can have 3 common values:

* Horizontal resize
* Vertical resize
* Both (horizontal & vertical) resize.

**CSS3 Resize property**

The CSS3 resize property specifies that whether an element should be resized by the user or not.

**Horizontal Resize**

Let's take an example to resize the width of a <div> element. (Horizontal resize)

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    border: 2px solid;

    padding: 20px;

    width: 300px;

    resize: horizontal;

    overflow: auto;

}

**</style>**

**</head>**

**<body>**

**<p><b>**Note:**</b>** Internet Explorer does not support the resize property.**</p>**

**<div>**This example specifies how to resize the width of a div element.**</div>**

**</body>**

**</html>**

**Example2:**

**Vertical Resize**

Let's take an example to resize the height of a <div> element. (Vertical resize)

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    border: 2px solid;

    padding: 20px;

    width: 300px;

    resize: vertical;

    overflow: auto;

}

**</style>**

**</head>**

**<body>**

**<p><b>**Note:**</b>** Internet Explorer does not support the resize property.**</p>**

**<div>**This example specifies how to resize the height of a div element.**</div>**

**</body>**

**</html>**

**Both (horizontal & vertical) resize**

You can also resize the width and height of a <div> element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    border: 2px solid;

    padding: 20px;

    width: 300px;

    resize: both;

    overflow: auto;

}

**</style>**

**</head>**

**<body>**

**<p><b>**Note:**</b>** Internet Explorer does not support the resize property.**</p>**

**<div>**This example specifies how to resize both the height and the width of this div element.**</div>**

**</body>**

**</html>**

**CSS3 Outline Offset**

The outline-offset property is used to add space between an outline and border of an element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div {

    margin: 20px;

    padding: 10px;

    width: 300px;

    height: 100px;

    border: 1px solid black;

    outline: 1px solid red;

    outline-offset: 10px;

}

**</style>**

**</head>**

**<body>**

**<p><b>**Note:**</b>** Internet Explorer does not support the outline-offset property.**</p>**

**<div>**This example specifies an outline 10px outside the border edge.

**</div>**

**</body>**

**</html>**

**Supporting Browsers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Chrome** | **IE** | **Firefox** | **Opera** | **Safari** |
| resize | 4.0 | not supported | 5.04.0 -moz- | 15.0 | 4.0 |
| outline-offset | 4.0 | not supported | 5.0  4.0 -moz- | 9.5 | 4.0 |

**CSS Pagination**

CSS pagination is a very useful technique for indexing different pages of a website on the homepage. If your website has lots of pages, you have to add some sort of pagination to each page.

Following are the different types of pagination:

**Basic Pagination**

This is the simplest pagination. You have to use pagination class to an <ul> element to attain this pagination.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<meta** http-equiv="Content-Type" content="text/html; charset=windows-1252"**>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

}

**</style>**

**</head>**

**<body>**

**<h2>**Basic Pagination**</h2>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**</ul>**

**</body>**

**</html>**

**Basic Pagination with arrow**

This pagination is used when you have lots of pages. It facilitates you to use arrow for previous and next page.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

}

**</style>**

**</head>**

**<body>**

**<h2>**Basic Pagination with arrow**</h2>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</body>**

**</html>**

**Active/Current link and Hoverable Pagination**

This pagination is used when you want to highlight the current page and change the color of each page-link when you move the mouse over them. You have to use .active class, and the :hover selector for this effect.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

}

ul.pagination li a.active {

    background-color: brown;

    color: white;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

**</style>**

**</head>**

**<body>**

**<h2>**Active and Hoverable Pagination**</h2>**

**<p>**Move the mouse over the numbers.**</p>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</body>**

**</html>**

**Rounded Active and Hoverable Pagination**

In this pagination, you use border-radius property to get the rounded "active" and "hover" button.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    border-radius: 5px;

}

ul.pagination li a.active {

    background-color: brown;

    color: white;

    border-radius: 5px;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

**</style>**

**</head>**

**<body>**

**<h2>**Rounded Active and Hover Buttons**</h2>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</body>**

**</html>**

**Bordered Pagination**

In this pagination, you use border property to add borders to the pagination.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    border: 1px solid black;

}

ul.pagination li a.active {

    background-color: brown;

    color: white;

    border: 1px solid grey;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

**</style>**

**</head>**

**<body>**

**<h2>**Bordered Pagination**</h2>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</body>**

**</html>**

**Rounded Border Pagination**

This pagination method is used to add rounded borders to your first and last link of pagination.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    border: 1px solid black;

}

.pagination li:first-child a {

    border-top-left-radius: 5px;

    border-bottom-left-radius: 5px;

}

.pagination li:last-child a {

    border-top-right-radius: 5px;

    border-bottom-right-radius: 5px;

}

ul.pagination li a.active {

    background-color: brown;

    color: white;

    border: 1px solid grey;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

**</style>**

**</head>**

**<body>**

**<h2>**Pagination with Rounded Borders**</h2>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</body>**

**</html>**

**Space Between Pagination**

The margin property is used to get the space between the links in the pagination.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    border: 1px solid black;

    margin: 0 4px;

}

ul.pagination li a.active {

    background-color: brown;

    color: white;

    border: 1px solid grey;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

**</style>**

**</head>**

**<body>**

**<h2>**Space Between Pagination**</h2>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</body>**

**</html>**

**Pagination Size**

You can change the size of the pagination by using font-size property.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    border: 1px solid black;

    font-size: 22px;

}

ul.pagination li a.active {

    background-color: brown;

    color: white;

    border: 1px solid grey;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

**</style>**

**</head>**

**<body>**

**<h2>**Pagination Size**</h2>**

**<p>**Change the font-size property to make the pagination smaller or bigger.**</p>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</body>**

**</html>**

**Centered Pagination**

You have to wrap a container element around it (like

) and use text-align:center to put the pagination to the center of the page.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    border: 1px solid #ddd;

}

ul.pagination li a.active {

    background-color: brown;

    color: white;

    border: 1px solid grey;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

div.center {text-align: center;}

**</style>**

**</head>**

**<body>**

**<h2>**Centered Pagination**</h2>**

**<div** class="center"**>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**1**</a></li>**

**<li><a** class="active" href="#"**>**2**</a></li>**

**<li><a** href="#"**>**3**</a></li>**

**<li><a** href="#"**>**4**</a></li>**

**<li><a** href="#"**>**5**</a></li>**

**<li><a** href="#"**>**6**</a></li>**

**<li><a** href="#"**>**7**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**</div>**

**</body>**

**</html>**

**Previous/Next and Navigation Pagination**

You can add pagination for previous/next button and also for navigation.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.pagination {

    display: inline-block;

    padding: 0;

    margin: 0;

}

ul.pagination li {display: inline;}

ul.pagination li a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    border: 1px solid black;

    font-size: 18px;

}

ul.pagination li a.active {

    background-color: brown;

    color: black;

    border: 1px solid black;

}

ul.pagination li a:hover:not(.active) {background-color: lightpink;}

**</style>**

**</head>**

**<body>**

**<p><strong>**Next/Previous buttons:**</strong></p>**

**<ul** class="pagination"**>**

**<li><a** href="#"**>**?**</a></li>**

**<li><a** href="#"**>**?**</a></li>**

**</ul>**

**<p><strong>**Navigation pagination:**</strong></p>**

**<ul** class="pagination"**>**

**<li><a** href="#" class="active"**>**Home**</a></li>**

**<li><a** href="#"**>**Link 1**</a></li>**

**<li><a** href="#"**>**Link 2**</a></li>**

**<li><a** href="#"**>**Link 3**</a></li>**

**</ul>**

**</body>**

**</html>**

**Breadcrumb Pagination**

A special type of pagination is called breadcrumb pagination.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ul.breadcrumb {

    padding: 8px 16px;

    list-style: none;

    background-color: #eee;

}

ul.breadcrumb li {display: inline;}

ul.breadcrumb li+li:before {

    padding: 8px;

    color: black;

    content: "/\00a0";

}

ul.breadcrumb li a {color: green;}

**</style>**

**</head>**

**<body>**

**<h2>**Be-practical**</h2>**

**<ul** class="breadcrumb"**>**

**<li><a** href="#"**>**Java**</a></li>**

**<li><a** href="#"**>**Oracle**</a></li>**

**<li><a** href="#"**>**PHP**</a></li>**

**<li>**AngularJS**</li>**

**</ul>**

**<p><strong>**Note:**</strong>**This is an example of Breadcrumb pagination.**</p>**

**</body>**

**</html>**

**CSS Layout**

**CSS layout** is easy to design. We can use CSS layout to design our web page such as home page, contact us, about us etc.

There are 3 ways to design layout of a web page:

1. **HTML Div with CSS**: fast and widely used now.
2. **HTML Table**: slow and less preferred.
3. **HTML Frameset**: deprecated now.

A CSS layout can have header, footer, left pane, right pane and body part. Let's see a simple example of CSS layout.

**CSS layout example**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

.header{margin:-8px -8px 0px;background-image:linear-gradient(145deg,#7379ff,#b524ef);color:white;text-align:center;padding:10px;}

.container{width:100%}

.left{width:15%;float:left;}

.body{width:65%;float:left;background-color:pink;padding:5px;}

.right{width:15%;float:left;}

.footer{margin:-8px;clear:both;background-image:linear-gradient(145deg,#7379ff,#b524ef);color:white;text-align:center;padding:10px;}

**</style>**

**</head>**

**<body>**

**<div** class="header"**><h2>**Be-practical**</h2></div>**

**<div** class="container"**>**

**<div** class="left"**>**

**<p>**Left Page**</p>**

**</div>**

**<div** class="body"**>**

**<h1>**Body Page**</h1>**

**<p>**Page Content goes here**</p><p>**Page Content goes here**</p><p>**Page Content goes here**</p>**

**<p>**Page Content goes here**</p><p>**Page Content goes here**</p><p>**Page Content goes here**</p>**

**<p>**Page Content goes here**</p><p>**Page Content goes here**</p><p>**Page Content goes here**</p>**

**<p>**Page Content goes here**</p><p>**Page Content goes here**</p><p>**Page Content goes here**</p>**

**<p>**Page Content goes here**</p>**

**</div>**

**<div** class="right"**>**

**<p>**Right Page**</p>**

**</div>**

**</div>**

**<div** class="footer"**>**

**<p>**Footer**</p>**

**</div>**

**</body>**

**</html>**

**Output:**

**Be-practical**

Left Page

**Body Page**

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

**CSS Table**

We can apply style on HTML tables for better look and feel. There are some CSS properties that are widely used in designing table using CSS:

* border
* border-collapse
* padding
* width
* height
* text-align
* color
* background-color

**CSS Table Border**

We can set border for the table, th and td tags using the CSS border property.

**<style>**

table, th, td {

    border: 1px solid black;

}

**</style>**

Output:

|  |  |  |
| --- | --- | --- |
| **First\_Name** | **Last\_Name** | **Marks** |
| Sonoo | Jaiswal | 60 |
| James | William | 80 |
| Swati | Sironi | 82 |
| Chetna | Singh | 72 |

**CSS Table Border Collapse**

By the help of border-collapse property, we can collapse all borders in one border only.

**<style>**

table, th, td {

    border: 2px solid black;

    border-collapse: collapse;

}

**</style>**

**Output:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Last Name** | **Marks** |
| Sonoo | Jaiswal | 60 |
| James | William | 80 |
| Swati | Sironi | 82 |
| Chetna | Singh | 72 |

**CSS Table Padding**

We can specify padding for table header and table data using the CSS padding property.

**<style>**

table, th, td {

    border: 1px solid black;

    border-collapse: collapse;

}

th, td {

    padding: 10px;

}

**</style>**

**Output:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Last Name** | **Marks** |
| Sonoo | Jaiswal | 60 |
| James | William | 80 |
| Swati | Sironi | 82 |
| Chetna | Singh | 72 |

**CSS Table: Styling even and odd cells**

We can style even and odd table cells for better look and feel. In this code, we are displaying different background colors on even and odd cells. Moreover, we have changed the background-color and color of <th> tag.

CSS code:

**<style>**

table, th, td {

    border: 1px solid black;

    border-collapse: collapse;

}

th, td {

    padding: 10px;

}

table#alter tr:nth-child(even) {

    background-color: #eee;

}

table#alter tr:nth-child(odd) {

    background-color: #fff;

}

table#alter th {

    color: white;

    background-color: gray;

}

**</style>**

Output:

