**CSS3**

CSS3 is the latest evolution of the Cascading Style Sheets language and aims at extending CSS2.1.

It brings a lot of long-awaited novelties, like rounded corners, shadows, gradients, transitions or animations, as well as new layouts like a multi-columns, flexible box or grid layouts.

**TEXT EFFECTS IN CSS3 – TEXT SHADOW**

A CSS3 Text Effect is such a term which is used to implement some extra features on the normal text. There are mainly two properties of CSS3 Text Effects, which have been described as follows:

 1:  **text-shadow** - text-shadow is used to create the shadow around the text, we can change the shadow color

2:  **word-wrap** - word-wrap is used to break the continued text in another line.

<p class = “text\_effect”>Hello</p>

<div id = “wraptext”>heloooooooooooooooooooooooooooooooooooooo</div>

**External CSS**

.text\_effect{

text-shadow: 2px 4px 5px blue; **//2px from horizontal, 4px from vertical and 5px blur**

}

# wraptext{

word-wrap: break-word; **//it will break your text in another line**

width: 100px;

}

**FONT IN CSS3 – FONT-FACE**

With the help of CSS3 Fonts feature, we can create different types of font style.

The font file can found in ttf(True type font) format or otf(Open type font) format.

The @font-face rule you must first define a name for the font (e.g. myFont), and then point to the font file.

<p class = “text\_effect font\_about”>Hello</p> **//two classes included**

**External CSS**

@font-face{

font-family: kuchbhi; **//you can create custom fonts using @font-face**

**src: url(file url should be placed of font which you have download in format ttf or otf);**

}

.font\_about{

font-family: kuchbhi;

}

**ANIMATION IN CSS3**

An Animation is such a property of CSS3, which is used to animate the object, without using flash or any other animation application. When the animation is created in the @keyframe, bind it to a selector, otherwise, the animation will have no effect.

animation   Required. Defines the name and duration of the animation.

keyframes-selector Required. Percentage of the animation duration.

Legal values: 0-100% from (same as 0%) to (same as 100%)

<div class = “batman”>

<img src = “URL of IMAGE” alt = “batman”>

</div>

**External CSS**

.batman{

width: 100%;

height: 100px;

background-color: red;

position: relative;

}

.batman img{

width: 140px;

height: 120px;

position: absolute;

left: 0; top: -15px;

**animation-name: batmanrun; //animation name**

**animation-duration: 6s; //animation duration**

**animation-timing-function: linear; //linear means at constant speed**

**animation-iteration-count: infinite; //no of times animation will perform function**

}

**@keyframes batmanrun{ //it will give the area which it covers**

**from{left: 0}**

**to{left: 85%}**

**}**

**TRANSITION IN CSS3**

A CSS3 Transition Effect is such an effect that lets an element gradually change from one style to another style when hover.

There are mainly four properties of CSS3 Transition Effects, which has been described as follows:

1: transition-property

2: transition-duration

3: transition-timing-function

4: transition-delays

<div class = “love”>

<img src = “URL of IMAGE” alt = “love”>

</div>

**External CSS**

**.love img{**

width: 50px;

height: 50px;

**transition-property: width, height; //what we want to change when image gets hover**

**transition-duration: 1s;**

**transition-timing-function: linear; //linear means at constant speed**

**}**

**.love img:hover{ //apply effects on hover**

width: 100px;

height: 100px;

**}**

**MULTIPLE COLUMN LAYOUT IN CSS3**

Multiple Columns is such an advanced feature of CSS3 which is used for creating the newspaper layout. You can create your articles to Multiple Columns, even if it is in one paragraph

The three properties of CSS3 Multiple Columns have been described as follows:

1: column-count

2: column-rule

3: column-gap

<div class = “about\_para”><p>Paragraph</p></div>

**External CSS**

.about\_para{

column-count: 3; **//create number of columns for div tag**

column-gap: 50px; **//it represent gap between two columns**

column-rule: solid/dashed; **//from which you separate different column**

column-rule-color: red;

text-align: justify;

}

**BACKGROUNDS IN CSS3 – BACKGROUND SIZE**

A CSS3 Backgrounds is an affords, which is used to resizing of the background properties.

background-size is such property which is used to fix the size of a background

**background-origin**

background-origin:border-box

background-origin:content-box

<div class = “mybg”></div>

**External CSS //create a div under which an image is to be shown**

.mybg{

width: 300px; height: 100px;

background-image: url(URL of IMAGE); **//image to be given in background**

background-repeat: no-repeat;

background-size: 100% 100%; **//if this not given, image will be shown in original size and may be possible it will not show completely in particular div tag(you can manage your image with background-size property of CSS3)**

}

**BORDERS IN CSS3 – BORDER-RADIUS**

A CSS3 Border is such an affords of style sheet which reduces the human efforts of Photoshop and other graphical applications.

border-radius is such property of CSS3 by which we can create the rounded corners

box-shadow is such property of CSS3 by which we can create the shadow of the border.

<div class = “card”></div>

<img scr = “URL of IMAGE” class = “imgborder”>

**External CSS**

.imgborder{

border-radius: 20px; **//it will give round shape around your corner**

}

.card{

box-shadow: 4px 4px 5px 10px grey; **//2px from horizontal, 4px from vertical, 5px blur and 10px stretch**

}

**USER INTERFACE IN CSS3 – BOX-SIZING, RESIZE**

CSS3 has introduced mainly three types of user interface that has been described:

* **resize:** defines if (and how) an element is resizable by the user (it can resize or stretch your div tag or an element according the value given in this property). This property does not apply to inline elements or to block elements where overflow=”visible”

**resize: vertical/horizontal/none;**

* **box-sizing:** it allows us to include the padding and border in an element’s total width and height. When you set the width/height of an element, the element often appears bigger than you have set (because the element's border and padding are added to the element's specified width/height).

<div><p>Box is Important</p></div>

**External CSS**

div{

width: 200px; height: 200px;

background-color: black;

padding-left: 30px;

**box-sizing: border-box; //it will set your div with specified width and height irrespective of padding and border**

}

p{

color: white;

}

* outline-offset

**2D/3D TRANSFORMATIONS IN CSS3**

CSS3 2D Transform has introduced mainly five types of methods that have been described as follows:

**translate()** method, the element moves from its current position, depending on the parameters given for the left (X-axis) and the top (Y-axis) position-->

**p{**

**transform: translatex(50px); //it will shift your content 50px on x axis (almost same as padding)**

**transform: translatey(50px); ////it will shift your content 50px on y axis**

**}**

**rotate()** method, the element rotates clockwise at a given degree. Negative values are allowed and rotate the element counter-clockwise. It will perform on hover.

**.text\_effect{**

**transition: transform 2s; //it will give smoothness to your rotation**

**}**

**.text\_effect:hover{**

**transform: rotate(360deg); //it will rotate your content 360 degree**

**}**

**scale()** method, the element increases or decreases the size, depending on the parameters given for the width (X-axis) and the height (Y-axis).

<div class = “love”>

<img src = “URL of IMAGE” alt = “love”>

</div>

**External CSS**

.love img{

width: **50px**;

height: **50px;**

transition-property: **transform; //what we want to change when image gets hover**

transition-duration: 1s;

}

.love img:hover{ **//apply effects on hover**

width: **100px**; **//we can use scale() function instead of giving width and height over hover**

height: **100px**;

**transform: scale(2,2); //here 2 means double the width and height of original image**

**}**

the skew() method, the element turns in a given angle, depending on the parameters given for the horizontal (X-axis) and the vertical (Y-axis) lines.

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

**LINEAR GRADIENT IN CSS3**

<section class = “join\_gradient”>

</section>

**External CSS**

.join\_gradient{

background-image: linear-gradient (to right red, yellow); **//it will mix your background image with different color(to right means red on left side, to left means red on right side)**

**}**

**FLEXBOX IN CSS3 (If you have multiple DIV under a DIV, then use flexbox OR if you have one row and multiple column, then use flexbox)**

<div class = “flexbox”>

<div><button>Like</button><button>Share</button><button>Subscribe</button></div>

</div>

**External CSS**

.flexbox{

width: 100%;

height: 100px;

background-image: linear-gradient (to right red, yellow);

**display: flex;**

**flex-direction: row;**

**justify-content: space-around;**

**align-items: center;**

}



**NOTE: Every page should contain this CSS at top.**

**\*{**

**margin: 0;**

**padding: 0;**

**box-sizing: border-box;**

**}**