CSS3 FEATURES

1. **Border**:The CSS3 provides two new properties for styling the borders of an element in a more elegant way — the border-image property for adding the images to borders, and the border-radius property for making the rounded corners without using any images.

```
   EXAMPLE:        .box {
width: 300px;
height: 150px;
background: #ffb6c1;
border: 2px solid #f08080;
border-radius: 20px;
border-img:url("c.jpg") 30    30;
}
```

2.color: CSS3 adds some new functional notations for setting color values for the elements which are rgba(), hsl() and hsla().

```
Example:          h1 {
             color: rgba(0,0,255,0.5);
             color: hsl(360,70%,60%);
            color: hsla(360,80%,50%,0.5);
```

Color Format	Syntax	Sample CSS
Short Hex Code	#RGB	p{color: #F0F;}
Hex Code	#RRGGBB	p{color: #FFFF00;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color: rgb(75%,0%,50%);}
keyword	red, yellow, etc.	p{color: aqua;}
RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(255,255,0);}

3.Background:The background-size property can be used to specify the size of the background images.   Prior to CSS3, the size of the background images was determined by the actual size of the images. The background image size can be specified using the pixels or percentage values as well as the keywords auto, contain, and cover. Negative values are not allowed.

```
        Example:.box {
width: 250px;
height: 150px;
background: url("images/sky.jpg") no-repeat;
background-size: contain;
border: 6px solid #333;}
```

4. FONT: The various font properties of CSS are used to define the font family, boldness, size, and style of your text that you will be using on your web page. You can set various font properties for your HTML element provided by the CSS. Some of these are:

Font Properties	Description
font-style	CSS font-style property is implemented for making a font italic, normal, and oblique.

font-family	CSS font-family property is implemented for transforming the face or look of your font.
font-variant	CSS font-variant property is implemented for creating a small-caps effect to your font.
font-weight	CSS font-weight property is implemented for enhancing or reducing how bold or light your font will appear.
font-size	CSS font-size property is implemented for enhancing or reducing the size of your font.

The font property of CSS is implemented as a shorthand notation to represent many other font properties.
CSS provides two different types of font family names:
Font family - are those with a particular font family (such as: "Times New Roman" or "Arial").
Generic family - are those with a cluster of font families having a similar look (like "Serif" or "Monospace").
The font-family property can be written with several font names separated by a comma as a "fallback" system. The fallback system means, in case the browser does not hold up or support or recognize the primary font name, it will try to deal with the next font, and so on. You can keep multiple font-family names as alternative options.

5.TEXT:It is possible to assign various CSS text properties in your HTML element using the CSS text properties listed below:

CSS Text Property	Description
color	The color property can be applied to set the color in your text.
direction	The direction property can be applied to set the direction of your text.
letter-spacing	The letter-spacing property can be applied to add or subtract space between letters, which will form a word.
text-decoration	The text-decoration property can be applied for underlining, overlining, and strikethrough text.
text-shadow	The text-shadow property can be applied for setting the text-shadow on your text.
text-transform	The text-transform property can be applied to capitalize text or convert text to uppercase or lowercase letters.
word-spacing	The word-spacing property can be applied to add or subtract space between different words in your sentence.
text-indent	The text-indent property can be applied to indent the text of your paragraph.
text-align	The text-align property can be applied to align text in your document.
white-space	The white-space property can be applied to control flow, as well as to the formatting of your text.

Let us discuss these in detail and see how to implement them and how it looks in the browser.
The color property of CSS is implemented for assigning colors to your texts. Three different approaches can specify these colors:
By using the color name, such as red, aqua, blue
By using the HEX value, such as #ff0000, #ffff00
By using the RGB value, such as rgb(255,0,0), rgb(255,255,0)
Here is a code snippet of how to implement:

```
<style>
p {
    color: red;}

body {
    color: aqua;}</style>
```

6.PADDING : With the help of CSS, it is possible to have full control over the HTML element's padding feature. The padding value implemented within CSS should have to be either of the following:
A percentage value (specify the width of the containing element).

A Length (px, pt, cm, etc).
A word inherit (padding value can be inherited from the parent element).
With the following properties, a different set of values can be assigned to the CSS padding for each side of the HTML element.

CSS Property	Description
padding	The CSS padding property is used as a shorthand for its different properties. It will be explained below with examples.
padding-top	The CSS padding-top is used to indicate the top padding for your element.
padding-right	The CSS padding-right is used to indicate the right padding for your element.
padding-bottom	The CSS padding-bottom is used to indicate the bottom padding for your element.
padding-left	The CSS padding-left is used to indicate the left padding for your element.

Here is a sample code snippet showing the implementation of all four (top, right, bottom, and left) CSS padding properties:

```
<style>
    selector {
        padding-top: 10px;
        padding-right: 20px;
        padding-bottom: 10px;
        padding-left: 20px;
    }</style>
```

7.MARIGIN: With the help of CSS, it is possible to control the margin of any HTML element. Different properties are available that can be used to set the margins for all sides of any HTML element. The margin value implemented within CSS should have to be either of the following:
An automatic, i.e., auto value where your browser will calculate the margin for you
A percentage value (specify the width of containing element)
A Length (px, pt, cm, etc.)
A word inherit (margin value can be inherited from parent element)
It should be noted that negative values are also allowed. Here is a list of CSS properties that are implemented to provide margins on HTML elements.

CSS Property	Description
margin	The CSS margin property is used as shorthand for its different properties as a single declaration.
margin-top	The CSS margin-top is used to indicate the top margin for your element.
margin-right	The CSS margin-right is used to indicate the right margin for your element.
margin-bottom	The CSS margin-bottom is used to indicate the bottom margin for your element.
margin-left	The CSS margin-left is used to indicate the left margin for your element.

Here is a sample code snippet showing the implementation of all four (top, right, bottom, and left) CSS margin properties:

```
<style>
    selector {
        margin-top: 10px;
        margin-right: 20px;
        margin-bottom: 10px;
        margin-left: 20px;
    }</style>
```

It is probable for shortening your CSS to specify all four (top, right, bottom, and left) margin properties under a single property. The margin property, as discussed above, is the individual margin properties. To use shorthand, you must use the margin property, which includes four other margin properties.

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

```
            .main_container {
                border: 1px solid #eee;
                width: 100%;
            }

            h2 {
                margin: 30px 40px 65px 90px;
                text-align: center;
            }
        </style></head>
<body>
        <div class="main_container">
            <h2>Margin shorthand property</h2>
        </div>

        <div>The concept of margin in CSS is used to define a completely transparent outer space, that is, without background color for your elements.</div>
</body></html>
```

8.BOARDER: CSS's border properties allow its users to specify the style, color, and border width of any HTML element for which it is specified. There are three properties based on which border properties can be changed:

CSS Border Property	Description
border-color	border-color is used to specify the color of the border of your element.
border-style	border-style is used to specify whether your element's border should be solid, line, dashed, double-line, or any other possible value.
border-width	The border-width is used for assigning the width value for your border.

Let us now see how they can be implemented within a program.
The border color can be implemented within a program.

```
<style>.classname {
        border:2px solid;
        border-bottom-color: red;
        border-top-color: #009900; /* Green */
        border-left-color: #0000CC;   /* Blue */
        border-right-color: orange}</style>
```

The style property (implemented using border-style) dictates what type of style your border will display. Following values can be assigned to this property:

Border Value	Description
dashed	It creates a dashed border.
dotted	It creates a dotted border.
solid	It creates a solid border.
none	It removes existing borders from the element.
hidden	It hides existing borders from the element.
double	It creates a double line border.
ridge	It creates a border with 3-D rigged. Its outcome depends largely on the border-color value.
inset	It creates a 3-D inset border. Its outcome depends largely on the border-color value.
groove	It creates a border with 3-D grooved. The outcome depends largely on the border-color value.
outset	It creates a 3-D outset border.

This is a code snippet of how they can be implemented:

```
<style>
```

```
h2.solid { border-style: solid; }

h2.dotted { border-style: dotted; }

h2.dashed { border-style: dashed; }

h2.none { border-style: none; }

h2.hidden { border-style: hidden; }

h2.double { border-style: double; }

h2.ridge { border-style: ridge; }

h2.groove { border-style: groove; }

h2.inset { border-style: inset; }

h2.outset { border-style: outset; }

h2.mix { border-style: dotted dashed solid double; }</style>
```

The width of the border (applied using the border-width property) is determined using a particular size (typically measured in px, em, pt, cm, etc.) or by using any of the 3 pre-defined words (thin, medium, or thick) as values. This property can be one of the four values, such as (top border, right border, bottom border, and left border).

Here is a CSS code snippet which shows how you can implement them:

```
<style>.classname {
        border-style: solid;
        border-width: 6px;}
.classname1 {
        border-style: solid;
        border-width: medium;}</style>
```

For shortening your code, you can probably identify all the individual border properties under a single property. You have to make use of the border property for implementing the shorthand notation for implementing border properties individually on all borders:

border-width

border-style (required)

border-color

Here is a program showing the implementation for shorthand:

```
<style>.classname {
        border: 4px solid aqua;}</style>
```

You can use the border-radius property to round the boundaries of any HTML element. Here is a CSS code snippet of how to implement:

```
<style>.classname {
        border: 4px solid aqua;
        border-radius: 6px;}</style>
```

9.MARQUEES: Marquee is an animation effect for web pages used to create horizontal or vertical scrolling text and images. The <marquee> element of HTML is not a standard-compliant, ie the element is not part of the W3 HTML specifications.

For creating a marquee using CSS, you have to use the CSS animation property together with the @keyframes rule. Let us now see its variations with implementation.

Here you have to implement translateX() for determining the position of the text at the beginning or end of the animation. Your text will then move among these two assigned points as the animation progresses.

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

```
<body>
    <style style="text/css">
        .marquee {
            height: 50px;
            overflow: hidden;
            position: relative;
            background: #fefefe;
            color: #333;
            border: 1px solid #4a4a4a;
        }

        .marquee p {
            position: absolute;
            width: 100%;
            height: 100%;
            margin: 0;
            line-height: 50px;
            text-align: center;
            -moz-transform: translateX(100%);
            -webkit-transform: translateX(100%);
            transform: translateX(100%);
            -moz-animation: scroll-left 2s linear infinite;
            -webkit-animation: scroll-left 2s linear infinite;
            animation: scroll-left 20s linear infinite;
        }

        @-moz-keyframes scroll-left {
            0% {
                -moz-transform: translateX(100%);
            }
            100% {
                -moz-transform: translateX(-100%);
            }
        }

        @-webkit-keyframes scroll-left {
            0% {
                -webkit-transform: translateX(100%);
            }
            100% {
                -webkit-transform: translateX(-100%);
            }
        }

        @keyframes scroll-left {
            0% {
                -moz-transform: translateX(100%);
                -webkit-transform: translateX(100%);
                transform: translateX(100%);
            }
            100% {
                -moz-transform: translateX(-100%);
                -webkit-transform: translateX(-100%);
                transform: translateX(-100%);
            }
        }
    </style>
    <div class="marquee">
        <p> Marquee in CSS </p>
    </div></body>
</html>
```

For making a scrolling image or animated content for your page, you have to replace your text (from the previous example) simply with any image of your choice. In addition, you have to make use of the <div> element and as a nested tag put the <img> tag within the <div>.

This animation feature of the marquee will bounce back once it reaches the end (left or right) of your page.

```
<!DOCTYPE html><html>
<body>
    <style style="text/css">
        .bounce {
            height: 50px;
            overflow: hidden;
            position: relative;
            background: #fefefe;
            color: #333;
            border: 1px solid #4a4a4a;
        }

        .bounce p {
            position: absolute;
            width: 100%;
            height: 100%;
            margin: 0;
            line-height: 50px;
            text-align: center;
            -moz-transform: translateX(50%);
            -webkit-transform: translateX(50%);
            transform: translateX(50%);
            -moz-animation: bouncing-text 5s linear infinite alternate;
            -webkit-animation: bouncing-text 5s linear infinite alternate;
            animation: bouncing-text 10s linear infinite alternate;
        }

        @-moz-keyframes bouncing-text {
            0% {
                -moz-transform: translateX(50%);
            }
            100% {
                -moz-transform: translateX(-50%);
            }
        }

        @-webkit-keyframes bouncing-text {
            0% {
                -webkit-transform: translateX(50%);
            }
            100% {
                -webkit-transform: translateX(-50%);
            }
        }

        @keyframes bouncing-text {
            0% {
                -moz-transform: translateX(50%);
                -webkit-transform: translateX(50%);
                transform: translateX(50%);
            }
            100% {
                -moz-transform: translateX(-50%);
                -webkit-transform: translateX(-50%);
                transform: translateX(-50%);
            }
        }
    </style>

    <div class="bounce">
        <p> Bouncy Marquee </p>
    </div></body>
</html>
```