CSS

1.INLINE CSS

  <body>

    <p style="color:green; font-size:300%">{inline css}

      Deciding what not to do is as important as deciding what to do.

    </p>

    </body>

2.internal css

    <style>// internal css

      p {

        color: green;

        font-size: 300%;

  }

  </style>

  <body>

    <p>

      Deciding what not to do is as important as deciding what to do.

    </p>

    <p>

      Second paragraph.

    </p>

3.External css

    <link rel="stylesheet" type="text/css" href="css/style.css">{css link}

  <body>

    <p>

      Deciding what not to do is as important as deciding what to do.

    </p>

    <p>{paragraph tags}

      Second paragraph.

    </p>

  </body>

p {

    color: green;

    font-size: 300%;//style file

}

4.Classes and ids

  <head>

    <style>

// for ids

      #decide {

        color: green;

        font-size: 300%;

      }

//for classes

      .blue{

        color: blue;

      }

    </style>

  </head>

  <body>

    <p id="decide">

      Deciding what not to do is as important as deciding what to do.

    </p>

4.div and span:

 The div tag is known as Division tag. The div tag is used in HTML to make divisions of content on the web page like (text, images, header, footer, navigation bar, etc).

The HTML span element is a generic inline container for inline elements and content.

<body>

  <div class="bluebox">

    <p>

        Facebook is

        <span class="great">

          great

        </span>

        .

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

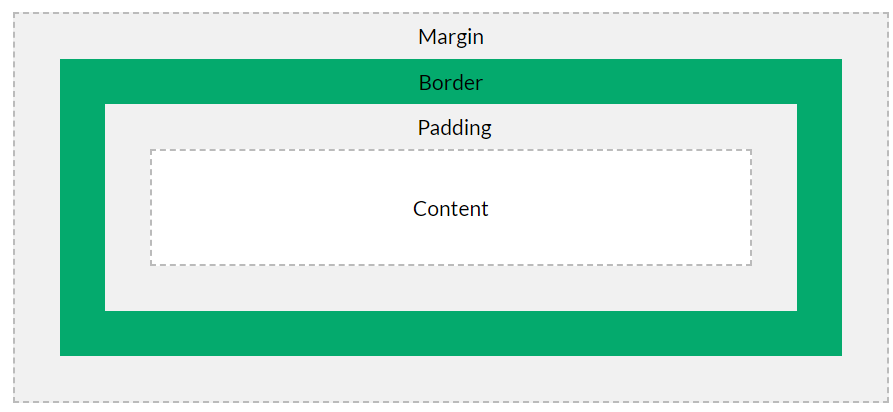
    </div>

  </body>

  </html>

5.Box model

* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent



Total element width = width + left padding + right padding + left border + right border + left margin + right margin.

When we say padding/border we mean both right and left.

6.Box model padding:

The height and width properties are used to set the height and width of an element.

<!DOCTYPE html>

<html>

<head>

<meta charset ="utf-8">

<title>Box Model Padding</title>

<style>

    .bluebox{

        background-color: #3A5795;

        color: white;

        height: auto;

        width:auto;

        padding: 16px;

    }

    .yellowbox{

        background-color: rgb(222, 224, 85);

        height: auto;

        width: auto;

        padding: 10px 5px 7px 3em;

    }

    .greenbox{

        background-color: #31af37;

        height: auto;

        width: auto;

        padding: 40px 10px;

    }

    p{

        margin: 0;

    }

</style>

</head>

<body>

    <div class="bluebox">

        <p>Facebook.</p>

    </div>

    <div class="yellowbox">

        <p>Apple.</p>

    </div>

    <div class="greenbox">

        <p>Microsoft.</p>

    </div>

</body>

</html>

7.Box model:BORDER

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

The border-radius property is used to add rounded borders to an element:

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      Box Model Border

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: auto;

        width:auto;

        padding: 16px;

        border: 10px #e8c14e ridge;// border

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: auto;

        width: auto;

        padding: 10px 5px 7px 3em;

        border-width: 10px 15px 5px 10px;

        border-color: black green orange yellow;

        border-style: ridge;

    }

    .greenbox{

      background-color: #31af37;

      height: 100px;

      width: 100px;

      padding: 10px;

      border-width: 10px;

      border-style: groove;

      border-color: #28ccb6;

      border-radius:80px;

    }

    p{

      margin: 0;

      background-color: #8bbbbf;

    }

  </style>

</head>

<body>

  <div class="bluebox">

    <p>

      Facebook.

    </p>

</div>

<div class="yellowbox">

  <p>

    Apple.

  </p>

</div>

<div class="greenbox">

  <p>

    Microsoft.

  </p>

</div>

</body>

</html>

8.box model:outline

An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      Box Model Outline

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: auto;

        width:auto;

        padding: 16px;

        border: 10px #e8c14e ridge;

        outline-style: solid;

        outline-color: #bc8df5;

        outline-width: thick;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: auto;

        width: auto;

        padding: 10px 5px 7px 3em;

        border-width: 10px 15px 5px 10px;

        border-color: black green orange yellow;

        border-style: ridge;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: 100px;

        padding: 10px;

        border-width: 10px;

        border-style: groove;

        border-color: #28ccb6;

        border-radius:80px;

      }

      p{

        margin: 0;

        background-color: #8bbbbf;

      }

    </style>

  </head>

  <body>

    <div class="bluebox">

      <p>

        Facebook.

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

    </div>

  </body>

  </html>

9.margin:

Margins are used to create space around elements, outside of any defined borders.

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      Box Model Margin

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: auto;

        width:auto;

        padding: 16px;

        border: 10px #e8c14e ridge;

        outline-style: solid;

        outline-color: #bc8df5;

        outline-width: thick;

        margin: 20px;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: auto;

        width: auto;

        padding: 10px 5px 7px 3em;

        border-width: 10px 15px 5px 10px;

        border-color: black green orange yellow;

        border-style: ridge;

        margin: 60px 10px 22px 20px;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: 100px;

        padding: 10px;

        border-width: 10px;

        border-style: groove;

        border-color: #28ccb6;

        border-radius:80px;

        margin: 60px 10px;

      }

      p{

        margin: 0;

        background-color: #8bbbbf;

    }

</style>

</head>

<body>

<div class="bluebox">

  <p>

    Facebook.

  </p>

</div>

<div class="yellowbox">

  <p>

    Apple.

  </p>

</div>

<div class="greenbox">

  <p>

    Microsoft.

  </p>

</div>

</body>

</html>

10.BACkground images:

The background-color property specifies the background color of an element.

The background-image property specifies an image to use as the background of an element.

By default, the background-image property repeats an image both horizontally and vertically.

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

The background-size property specifies the size of the background images.

The background-origin property specifies the origin position (the background positioning area) of a background image.

|  |  |  |
| --- | --- | --- |
| padding-box | Default value. The background image starts from the upper left corner of the padding edge | [Demo ❯](https://www.w3schools.com/cssref/playdemo.asp?filename=playcss_background-origin) |
| border-box | The background image starts from the upper left corner of the border | [Demo ❯](https://www.w3schools.com/cssref/playdemo.asp?filename=playcss_background-origin&preval=border-box) |
| content-box | The background image starts from the upper left corner of the content |  |

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      Background

    </title>

    <style>

      .bluebox{

        background-image: url(images/picture.jpg);

        background-size: 100% 100%;

        background-repeat: no-repeat;

        background-origin: content-box;

        color: white;

        height: 100px;

        width:auto;

        padding: 16px;

        border: 10px #e8c14e ridge;

        outline-style: solid;

        outline-color: #bc8df5;

        outline-width: thick;

        margin: 20px;

      }

      .yellowbox{

        background-image: url(images/picture.jpg);

        background-size: 50px 50px;

        background-repeat: no-repeat;

        background-origin: border-box;

        /\*        background-position: left;

        \*/

        height: 100px;

        width: auto;

        padding: 10px 5px 7px 3em;

        border-width: 10px 15px 5px 10px;

        border-color: black green orange yellow;

        border-style: ridge;

        margin: 60px 10px 22px 20px;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: 100px;

        padding: 10px;

        border-width: 10px;

        border-style: groove;

        border-color: #28ccb6;

        border-radius:80px;

        margin: 60px 10px;

      }

      p{

        margin: 0;

        background-color: #8bbbbf;

    }

</style>

</head>

<body>

<div class="bluebox">

  <p>

    Facebook.

  </p>

</div>

<div class="yellowbox">

  <p>

    Apple.

  </p>

</div>

<div class="greenbox">

  <p>

    Microsoft.

      </p>

    </div>

  </body>

</html>

11.Float and clear

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

When we use the float property, and we want the next element below (not on right or left), we will have to use the clear property.

The clear property specifies what should happen with the element that is next to a floating element.

The clear property can have one of the following values:

* both - The element is pushed below both left and right floated elements

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      floating

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

        float: left;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

        float: right;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: auto;

        clear: both;

      }

      .great{

        background-color: #5a3479;

      }

    </style>

  </head>

  <body>

    <div class="bluebox">

      <p>

        Facebook is

        <span class="great">

          great

        </span>

        .

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

    </div>

  </body>

  </html>

12.positioning: The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

|  |  |
| --- | --- |
| [top](https://www.w3schools.com/cssref/pr_pos_top.asp) | Sets the top margin edge for a positioned box |
| [left](https://www.w3schools.com/cssref/pr_pos_left.asp) | Sets the left margin edge for a positioned box |

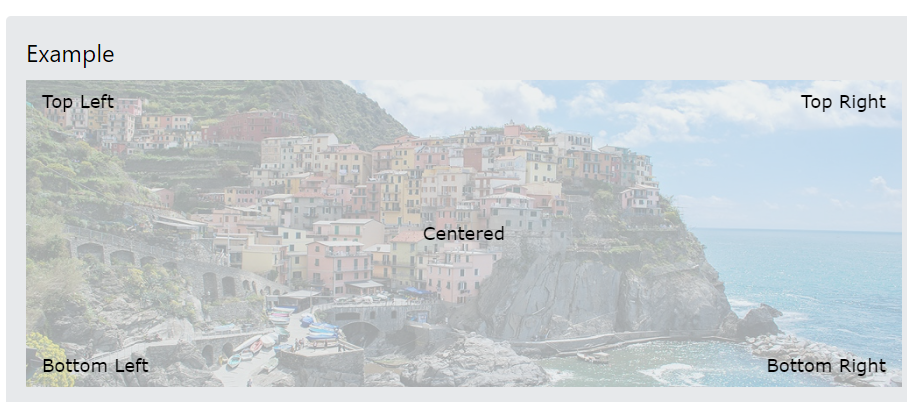
The z-index property specifies the stack order of an element.

When elements are positioned, they can overlap other elements.

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

z-index only works on [positioned elements](https://www.w3schools.com/css/css_positioning.asp) .

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



<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      positioning

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

        position: fixed;

        top: 22px;

        left: 22px;

        /\*        z-index: 1;

        \*/

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

        position: relative;

        top: 20px;

        left: 20px;

        /\*        z-index: 0;

        \*/

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: auto;

        position: absolute;

        top: 40px;

        left: 40px;

      }

      .great{

        background-color: #5a3479;

      }

    </style>

  </head>

  <body>

    <div class="bluebox">

      <p>

        Facebook is

        <span class="great">

          great

        </span>

        .

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

    </div>

</body>

</html>

13.display in css: The display property specifies the display behavior (the type of rendering box) of an element.

|  |  |
| --- | --- |
| block | :Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width |
| none | The element is completely removed |

The visibility property specifies whether or not an element is visible.

|  |  |
| --- | --- |
| hidden | The element is hidden (but still takes up space) |

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      display

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

        /\*        visibility: hidden;

        \*/

        /\*        display: none;

        \*/

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

    }

    .greenbox{

      background-color: #31af37;

      height: 100px;

      width: auto;

    }

    .great{

      background-color: #5a3479;

      display: block;

    }

  </style>

</head>

<body>

  <div class="bluebox">

    <p>

      Facebook is

      <span class="great">

        great

      </span>

      .

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

    </div>

  </body>

  </html>

14.text font

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      text font

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

        font-size: 100px;

        font-family: sans-serif;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

        font-size: 100px;

        font-family: "Times New Roman", Times, serif;

    }

    .greenbox{

      background-color: #31af37;

      height: 100px;

      width: auto;

      font-size: 100px;

      font-family: monospace;

    }

    .great{

      background-color: #5a3479;

    }

  </style>

</head>

<body>

  <div class="bluebox">

    <p>

      Facebook

    </p>

</div>

<div class="yellowbox">

  <p>

    Apple.

  </p>

</div>

<div class="greenbox">

  <p>

    Microsoft.

  </p>

</div>

</body>

</html>

15.text decoration:

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      div and span

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

        text-decoration: underline;

        font-weight: bold;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

        text-decoration: overline;

        font-style: italic;

        font-size: 1.5em;

    }

    .greenbox{

      background-color: #31af37;

      height: 100px;

      width: auto;

      letter-spacing: 10px;

      text-shadow: 3px 3px #c18c04;

      text-indent: 40px;

    }

    .great{

      background-color: #5a3479;

    }

  </style>

</head>

<body>

  <div class="bluebox">

    <p>

        Facebook is

        <span class="great">

          great

        </span>

        .

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

    </div>

</body>

</html>

15.text align:

The text-align property specifies the horizontal alignment of text in an element.

|  |  |  |
| --- | --- | --- |
| left | Aligns the text to the left | [Demo ❯](https://www.w3schools.com/cssref/playdemo.asp?filename=playcss_text-align) |
| right | Aligns the text to the right | [Demo ❯](https://www.w3schools.com/cssref/playdemo.asp?filename=playcss_text-align&preval=right) |
| center | Centers the text | [Demo ❯](https://www.w3schools.com/cssref/playdemo.asp?filename=playcss_text-align&preval=center) |
| justify | Stretches the lines so that each line has equal width (like in newspapers and magazines) |  |

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      text align

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

        text-decoration: underline;

        font-weight: bold;

        /\*        text-align: justify;

        \*/

        /\*        text-align: center;

        \*/

        /\*        text-align: left;

        \*/

        text-align: right;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

        text-decoration: overline;

        font-style: italic;

        font-size: 1.5em;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: auto;

        letter-spacing: 10px;

        text-shadow: 3px 3px #c18c04;

        text-indent: 40px;

      }

      .great{

        background-color: #5a3479;

      }

    </style>

  </head>

  <body>

    <div class="bluebox">

      <p>

        Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

    </div>

  </body>

</html>

16.text effects:

The CSS text-overflow property specifies how overflowed content that is not displayed should be signaled to the user.

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

word-wrap: break-word;Allow long words to be able to be broken and wrap onto the next line:

|  |  |
| --- | --- |
| [word-break](https://www.w3schools.com/cssref/css3_pr_word-break.asp) | Specifies line breaking rules for non-CJK scripts |

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      text effects

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

        overflow: hidden;

        white-space: nowrap;

        text-overflow: ellipsis;

      }

      .bluebox:hover{

        overflow: visible;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: 400px;

        /\*        word-break: break-all;

        \*/

        /\*        word-break: keep-all;

        \*/

        /\*        word-wrap: break-word;

        \*/

        overflow: hidden;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: auto;

      }

      .great{

        background-color: #5a3479;

    }

</style>

</head>

<body>

<div class="bluebox">

  Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook

  Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook

  Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook

  Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook

  Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook

  Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook Facebook

</div>

<div class="yellowbox">

  <p>

    Apple Apple Apple Apple Apple Apple LoooooooooongApple Apple Apple Apple Apple Apple Apple Apple Apple Apple

    Apple Apple Apple Apple Apple Apple Apple LoooooooooooooooooooooooooooooooooooooooooooooooooooongApple Apple

    Apple Apple Apple Apple Apple Apple Apple Apple Apple Apple Apple Apple

  </p>

</div>

<div class="greenbox">

  <p>

    Microsoft.

  </p>

</div>

</body>

</html>

18.image spirites

An image sprite is a collection of images put into a single image.

n the following example the CSS specifies which part of the "img\_navsprites.gif" image to show:

* width: 46px; height: 44px; - Defines the portion of the image we want to use
* background: url(img\_navsprites.gif) 0 0; - Defines the background image and its position (left 0px, top 0px)

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      imagesprites

    </title>

    <style>

      #image1{

        width: 500px;

    height: 500px;

    background: url(images/picture.jpg) -700px -200px;

  }

  </style>

  </head>

  <body>

    <img id="image1">

  </body>

  </html>

19.image opacity

<!DOCTYPE html>

<html>

    <head>

        <meta charset ="utf-8">

        <title>

            image opacity

        </title>

        <style>

            .image1{

                width: 200px;

                height: 150px;

                opacity: 0.4;

                margin-right: 22px;

            }

            .image2{

                width: 200px;

                height: 150px;

                opacity: 1;

            }

        </style>

    </head>

    <body>

        <img class="image1" src="feathers.jpg">

        <img class="image2" src="feathers.jpg">

    </body>

</html>

20.lists

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      lists

    </title>

    <style>

      ul.list1{

        /\*    list-style: square;

        \*/

        /\*    list-style: lower-roman;

        \*/

        /\*    list-style: upper-roman;

        \*/

        list-style-image: url(images/bluesquare.gif)

      }

      ul.list2{

        list-style: none;

        padding: 0;

        margin: 0;

      }

      ul.list2 li{

        background-image: url(images/bluesquare.gif);

        background-repeat: no-repeat;

        background-position: 0px center;

        padding-left: 22px;

      }

    </style>

  </head>

  <body>

    <p>

      A list of European capitals:

    </p>

    <ul class="list1">

        <li>

          Paris

        </li>

        <li>

          London

        </li>

        <li>

          Madrid

        </li>

    </ul>

    <p>

      A list of European capitals:

    </p>

    <ul class="list2">

      <li>

        Paris

        </li>

        <li>

          London

        </li>

        <li>

            Madrid

        </li>

    </ul>

    </body>

  </html>

12.styling links

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      div and span

    </title>

    <style>

      a:link{

        color: #2a8e1f;

      }

      a:hover{

        font-size: 3em;

      }

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:auto;

      }

      div.bluebox a:active {

        font-size: 4em;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

      }

      div.yellowbox a:visited{

        color: #f2a145;

        font-size: 6em;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: auto;

      }

      .great{

        background-color: #5a3479;

      }

    </style>

</head>

<body>

  <div class="bluebox">

    <p>

      Facebook.

    </p>

    <p>

      <a href="http://www.amazon.com">

        Visit Link.

      </a>

    </p>

  </div>

  <div class="yellowbox">

    <p>

      Apple.

    </p>

    <p>

        <a href="http://www.ebay.com">

            Visit Link.

          </a>

        </p>

      </div>

      <div class="greenbox">

        <p>

          Microsoft.

        </p>

        <p>

          <a href="http://www.cnn.com">

            Visit Link.

          </a>

        </p>

      </div>

    </body>

  </html>

16.gradients: CSS gradients let you display smooth transitions between two or more specified colors.

CSS defines three types of gradients:

* **Linear Gradients (goes down/up/left/right/diagonally)**
* **Radial Gradients (defined by their center)**
* **Conic Gradients (rotated around a center point)**
* CSS Linear Gradients

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

radial gradients

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

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      gradients

    </title>

    <style>

      .bluebox{

        /\*        background-color: #3A5795;

        \*/

        color: white;

        height: 100px;

        width:auto;

        /\*        background: linear-gradient(blue,white);

        \*/

        /\*

        background: -webkit-linear-gradient(blue,white);

        background: -o-linear-gradient(blue,white);

        background: -moz-linear-gradient(blue,white);

        \*/

        /\*        background: linear-gradient(blue,white,orange);

        \*/

        /\*        background: linear-gradient(to left, blue,white,orange);

        \*/

        /\*        background: linear-gradient(to bottom left, blue,white,orange);

        \*/

        /\*        background: linear-gradient(90deg, blue,white,orange);

        \*/

        /\*        background: linear-gradient(to left, rgba(0,0,255,0.5),rgba(0,0,255,1));

        \*/

        /\*        background: radial-gradient(blue,white,orange);

        \*/

        /\*        background: radial-gradient(blue 5%,white 15%,orange 80%);

        \*/

        background: radial-gradient(circle,blue 5%,white 15%,orange 80%);

    }

    .yellowbox{

      background-color: rgb(222, 224, 85);

      height: 100px;

      width: auto;

    }

    .greenbox{

        background-color: #31af37;

        height: 100px;

        width: auto;

      }

      .great{

        background-color: #5a3479;

      }

    </style>

  </head>

  <body>

    <div class="bluebox">

      <p>

        Facebook.

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

    </p>

</div>

<div class="greenbox">

  <p>

    Microsoft.

  </p>

</div>

</body>

</html>

23.2d transforms

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

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

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

|  |  |
| --- | --- |
| rotate(*angle*) | Defines a 2D rotation, the angle is specified in the parameter |

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      2D tranforms

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:100px;

        /\*        transform: rotate(30deg);

        \*/

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

        \*/

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

        \*/

        /\*        transform: translate(20px,22px);

        \*/

        /\*        transform: scale(0.5,1);

        \*/

        transform: skew(10deg,10deg);

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: auto;

      }

      .great{

        background-color: #5a3479;

    }

</style>

</head>

        <body>

            <div class="bluebox">

              <p>

                Facebook.

              </p>

            </div>

            <div class="yellowbox">

              <p>

                Apple.

              </p>

            </div>

            <div class="greenbox">

              <p>

                Microsoft.

              </p>

            </div>

          </body>

        </html>

3d transforms

With the CSS transform property you can use the following 3D transformation methods:

* rotateX()
* rotateY()
* rotateZ()

The rotateX() method rotates an element around its X-axis at a given degree:

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      3D tranforms

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:100px;

        /\*        transform: rotateX(30deg);

        \*/

        transform: rotateY(30deg);

        /\*        -ms-transform: rotateY(30deg);

        \*/

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

        \*/

    }

    .yellowbox{

      background-color: rgb(222, 224, 85);

      height: 100px;

      width: auto;

    }

    .greenbox{

      background-color: #31af37;

      height: 100px;

      width: auto;

    }

    .great{

      background-color: #5a3479;

    }

  </style>

</head>

25.transitions and hover:

CSS transitions allows you to change property values smoothly, over a given duration.

To create a transition effect, you must specify two things:

* the CSS property you want to add an effect to
* the duration of the effect

The transition effect will start when the specified CSS property (width) changes value.

To specify a new value for the width property when a user mouses over the <div> element:

<!DOCTYPE html>

<html>

  <head>

    <meta charset ="utf-8">

    <title>

      transitions

    </title>

    <style>

      .bluebox{

        background-color: #3A5795;

        color: white;

        height: 100px;

        width:100px;

        transition: width 1s;

      }

      .bluebox:hover{

        width: 300px;

      }

      .yellowbox{

        background-color: rgb(222, 224, 85);

        height: 100px;

        width: auto;

      }

      .greenbox{

        background-color: #31af37;

        height: 100px;

        width: 100px;

        transition: width 1s, height 1s, transform 1s;

      }

      .greenbox:hover{

        width: 300px;

        height: 300px;

        transform: rotateX(180deg);

      }

      .great{

        background-color: #5a3479;

      }

    </style>

  </head>

  <body>

    <div class="bluebox">

      <p>

        Facebook.

      </p>

    </div>

    <div class="yellowbox">

      <p>

        Apple.

      </p>

    </div>

    <div class="greenbox">

      <p>

        Microsoft.

      </p>

    </div>

  </body>

</html>

Animations

CSS allows animation of HTML elements without using JavaScript or Flash!

When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Animations</title>

<style>

.bluebox{

background-color:#3B5697;

color:white;

height:100px;

width:100px;

position: relative;

-webkit-animation: myanimation 3s;

}

/\*

@keyframes myanimation {

from{background:green}

to{background:red}

}

 \*/

@-webkit-keyframes myanimation{

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

25% {background:yellow;left 200px;top 0px;}

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

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

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

}

.yellowbox{

background-color:rgb(250, 255, 0);

height:100px;

width:100px;

}

.greenbox{

background-color:rgb(10, 173, 18);

height:100px;

width:auto;

}

.great{

background-color:#593479;

}

</style>

</head>

<body>

<div class="bluebox">

<p>Facebook </p>

</div>

<div class="yellowbox">

<p>Apple.</p>

</div>

<div class="greenbox">

<p>Microsoft.</p>

</div>

</body>

</html>