MODULE - 4
CSS3 EFFECTS AND ANIMATIONS

Course Topics

- Module 1
  - Deploying the first Website to Amazon S3
- Module 2
  - Creating web pages with HTML5
- Module 3
  - Styling web pages using CSS
- Module 4
  - **CSS3 effects and animations**
- Module 5
  - Handling events with JavaScript
- Module 6
  - Twitter Bootstrap 3
- Module 7
  - Twitter Bootstrap 3 Project
- Module 8
  - Bootstrap ScrollSpy, jQuery and jQuery UI
- Module 9
  - Ajax, Google APIs, Social Plugins
- Module 10
  - Project - Building Website Tour

Objectives

At the end of this module, you will be able to:

→ Learn how to use the attributes of text-effects like text-shadow and word wrap

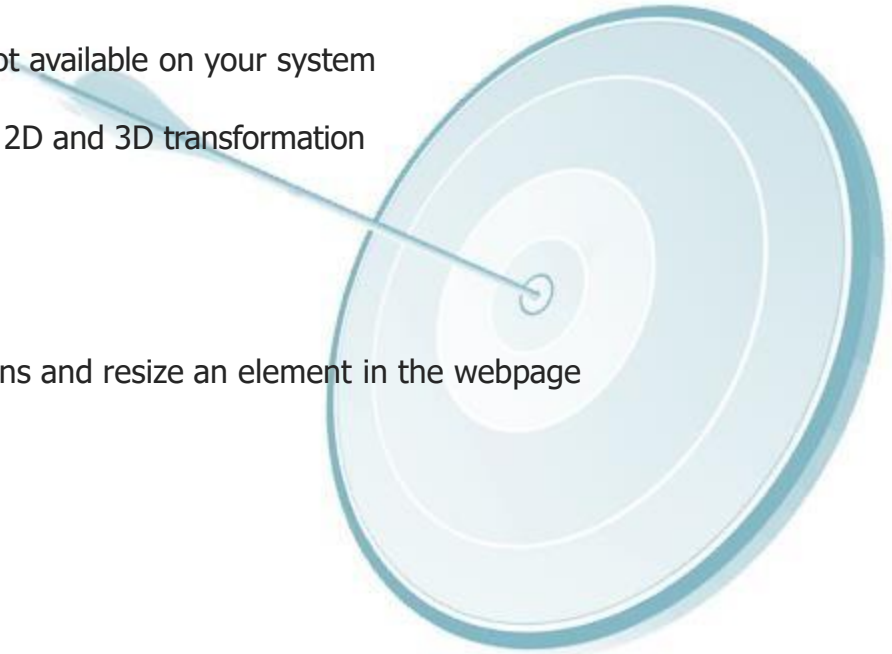
→ Learn how to present the text in a font style that is not available on your system

→ Understand the various attributes associated with the 2D and 3D transformation

→ Apply transition for the element in the webpage

→ Create an animation

→ Understand how to present the text in multiple columns and resize an element in the webpage



CSS3 – Text Effects

- → Sometimes you need text-effects to contrast between the background and the foreground
- → You can change the color, effect, or offset of a shadow using the text-effects property in CSS
- → By doing this, your text looks more effective and attractive

See the difference between the two texts shown below:

EDUREKA

Plain text

Text with text-effects

As seen from the above example, the text with text-effects is more attractive

CSS3 – Text Effects (Contd.)

→ When you want to create newsletters, brochures etc., you will have to make your text look interesting to the users. This can be done in CSS using the CSS Text Effects

You can change the look, add texture to your text using the text-effect property

CSS3 has the following text effects:

→ Text shadow: Gives the text shadow effect

→ Word wrap: Wraps the word

CSS3 – Text Effects – Text Shadow

→ Text-shadow property is used for giving the shadow effect to the text

Syntax: text-shadow: horizontal_shadow vertical_shadow blur_radius color_of_the_shadow;

Example

```
<style>
h1 {
    text-shadow: 5px 5px 5px #FF00FF;
}
</style>
```

Output

CSS3 – Text Effects – Word Wrap

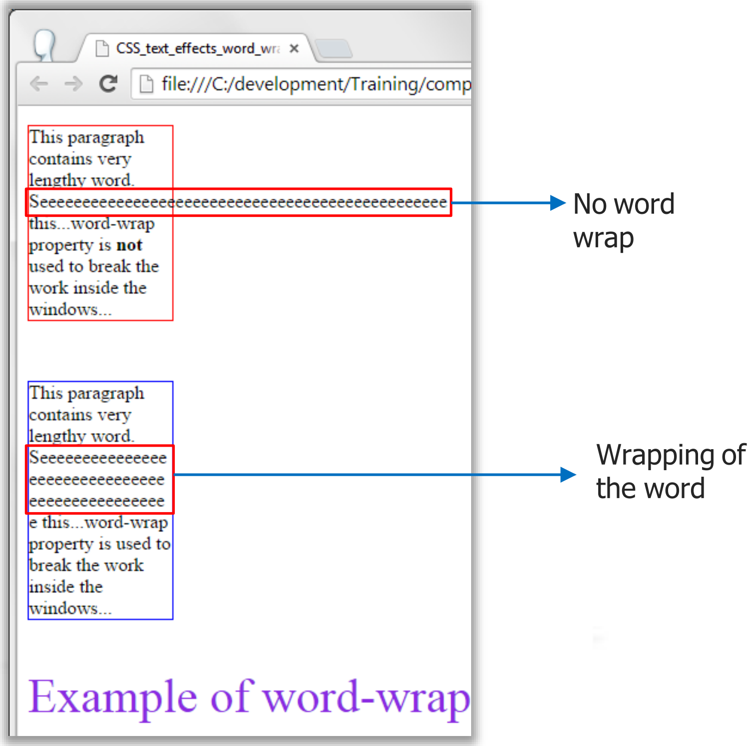
→ When the word is too lengthy to fit in, we use the word wrap property to split the word inside the window

Example

```
p.no_wordwrap {
    width: 120px;
    border: 1px solid #ff0000;

}
p.word_wrap {
    width: 120px;
    border: 1px solid #0000FF;
    word-wrap: break-word;
}
```

Output



Annie’s Question

What is the significance of Text effects in CSS?

Annie’s Answer

Ans. To highlight the text and represent the data in a better way text effects are used

CSS3 – Fonts

→ When users visit your website, they are probably brushing up your webpage looking for the content they want. So you have to highlight the important phrases in the text in a readable format

NORMAL IS BORING

v/s

Normal

CSS Font

CSS3 – Fonts (Contd.)

→ When you have found a new font-style and you want to present your text in the new style, you can use the CSS3 font-face property to display the text in the font which is not installed on your system

→ To do this, you have to download the font-style file and apply it to the required part of text

→ Font styles:

» TrueType Fonts(TTF file): This is the standard font which is developed by Apple and Microsoft

» OpenType Font(OTF file): This type has advanced features for fonts and it is developed by Adobe and Microsoft

» Web Open Font Format(WOFF file): This is either TTF file or OTF file with additional metadata

CSS3 – Fonts (Contd.)

→ Open Type Font (OTF) file is downloaded from net and stored in a directory. In this code, Felipa-Regular.otf is downloaded

→ Using font-face, name of the font is defined and font file is loaded by using src:url (font file)

→ This font name can be used for the HTML elements to apply the font style

Example

```
<style>
@font-face {
    font-family: myFirstFont;
    src: url(Felipa-Regular.otf);
}
div.font_file,h1 {
    font-size :40px;
    font-family: myFirstFont;
    color : #9400D3;
}
div.regular
{
    font-size:20px;
    font-family:sans serif;
}
```

Output

Annie’s Question

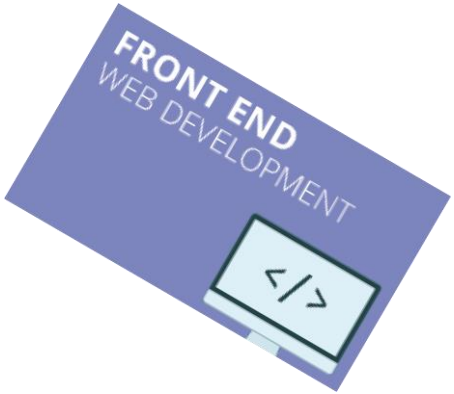
Why do we need to download the fonts when there are already available fonts?

Annie’s Answer

Ans. Windows or any other OS does not have all the fonts copied in the Operating System. Many beautiful calligraphy files are available in OTF/WOFF format so that we can download and use them

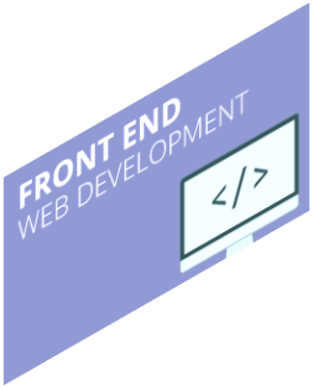
CSS3 – 2D Transformation

→ 2D transformations are a part of computer graphics. You can rotate, scale, position, shape and change the view of the element using the CSS3 2D transform properties



Rotate

Scale



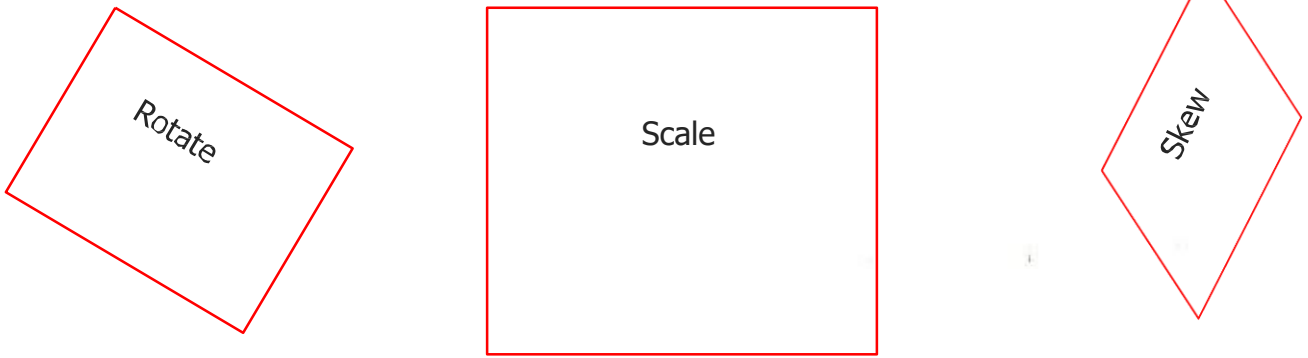
Skew

CSS3 – 2D Transformation (Contd.)

→ With 2D transformation HTML elements can be stretched, spinned, moved, scaled etc.

→ Some of the methods of 2D transformation are:

- » scale() : Increases or decreases the size of the elements
- » translate() : Moves the current element to another position
- » rotate() : Rotates the element by taking a degree
- » skewX() : Skews an element along the X-axis by the given angle
- » skewY() : Skews an element along the Y-axis by the given angle
- » skew() : Skews an element along the X and Y-axis by the given angles
- » matrix() : Can combine all the 2D methods



CSS3 – 2D Transformation – Translate

→ When you want to display an element from one position to another, translate() method is used

→ translate(x,y) – Translates the element to the specified position. Here the element is text

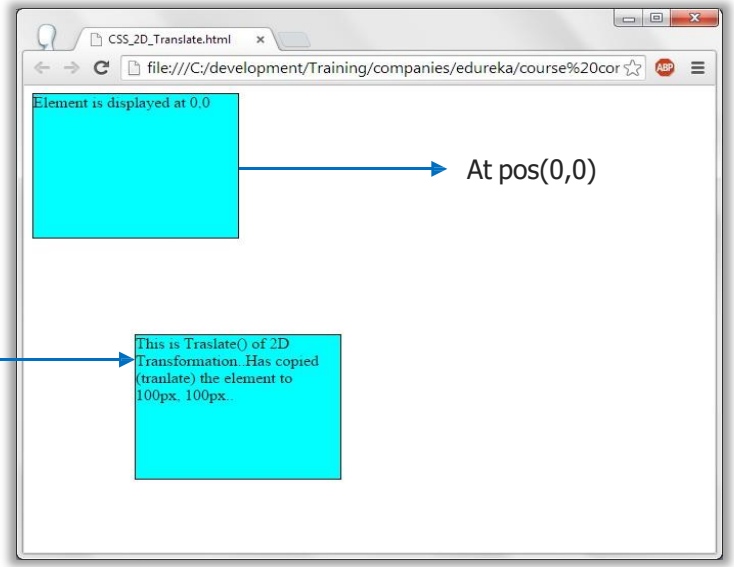
→ In our example, using the translate property of 2D transformation, the element from position(0,0) is translated to position(100,100)

Example

```
<style>
div {
    width: 200px;
    height: 150px;
    background-color: #00FFFF;
    border: 1px solid black;
}

div#div2 {
    -webkit-transform: translate(100px,100px);
}
</style>
```

Output



Different browsers support different transform properties. So make sure you add transformation properties for all the browsers e.g. Chrome, IE, Mozilla, Safari, Opera. Otherwise transformations will not work across different browsers

CSS3 – 2D Transformation – Rotate

→ When you want to rotate an element by a certain degree, rotate() method is used

→ rotate(xdeg) - Rotates the element to x degree. Here the element is image

→ In our example, using the rotate property of 2D transformation, the image is rotated to 30deg

Example

```
img#id1 {
        position:absolute;
        left: 600px;
         top: 100px;
    -webkit-transform: rotate(30deg);
}
img#id2 {
         position:absolute;
        left: 300px;
         top: 100px;
    -webkit-transform: rotate(30deg);
}
```

The image is rotated by 30 degrees

Output

CSS3 – 2D Transformation – Scale

→ When you want to enlarge the size of the element, scale() method is used

→ scale() - Scales the element up/down depending on the axis mentioned in the code

→ Syntax: -webkit-transform: scale(x,y);

» Scales the element to the x (horizontally) and y (vertically) position appropriately

» If scale (2,2) is used, the width and height is scaled twice as the original element (e.g. image)

Example

```
img#id1{
        position: absolute;
        left: 400px;
        top: 100px;
       -webkit-transform: scale(1,1);
}

img#id2 {
       position: absolute;
        left: 400px;
        top: 300px;
     -webkit-transform: scale(2,2);
}
```

Output

CSS3 – 2D Transformation – Skew

→ When you want the element to be displayed in an oblique angle, skew() method is used

→ skew() - Rotates the element in horizontal and vertical axis

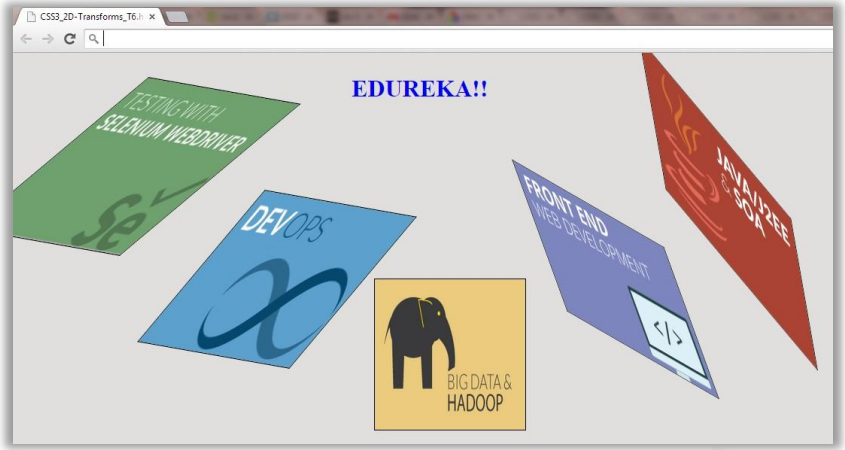
→ Syntax: webkit-transform: skew(20deg,30deg);

» This will skew the element by 20 degrees horizontally(x-axis) and 30 degrees vertically(y-axis)

Example

```
img#id1 {
        position:absolute;
        left: 650px;
         top: 200px;
    -webkit-transform: skew(20deg,30deg);
}
img#id2 {
         position:absolute;
        left: 800px;
         top: 100px;
    -webkit-transform: skew(10deg,50deg);
}
```

Output



Annie's Question

Where can we use 2D scale() transformation?

Annie’s Answer

Ans. To increase the size of the image or decrease the size of the image scale() can be used.

CSS3 – 3D Transformation

→ 3D transforms operate horizontally(x-axis) and vertically(y-axis)

→ They make the elements look interactive so that the user is always associated with the webpage

→ It gives you a look of a virtual reality

In the above image, you can see the transformation of number 3 to number 5, using the 3D transform
Now let us learn how to write code to perform such transformations

CSS3 – 3D Transformation (Contd.)

→ With CSS3 elements can also be rotated in 3D direction using following methods :

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

→ rotateY() : rotates an element around its Y-axis at a given degree

→ rotateZ() : rotates an element around its Z-axis at a given degree

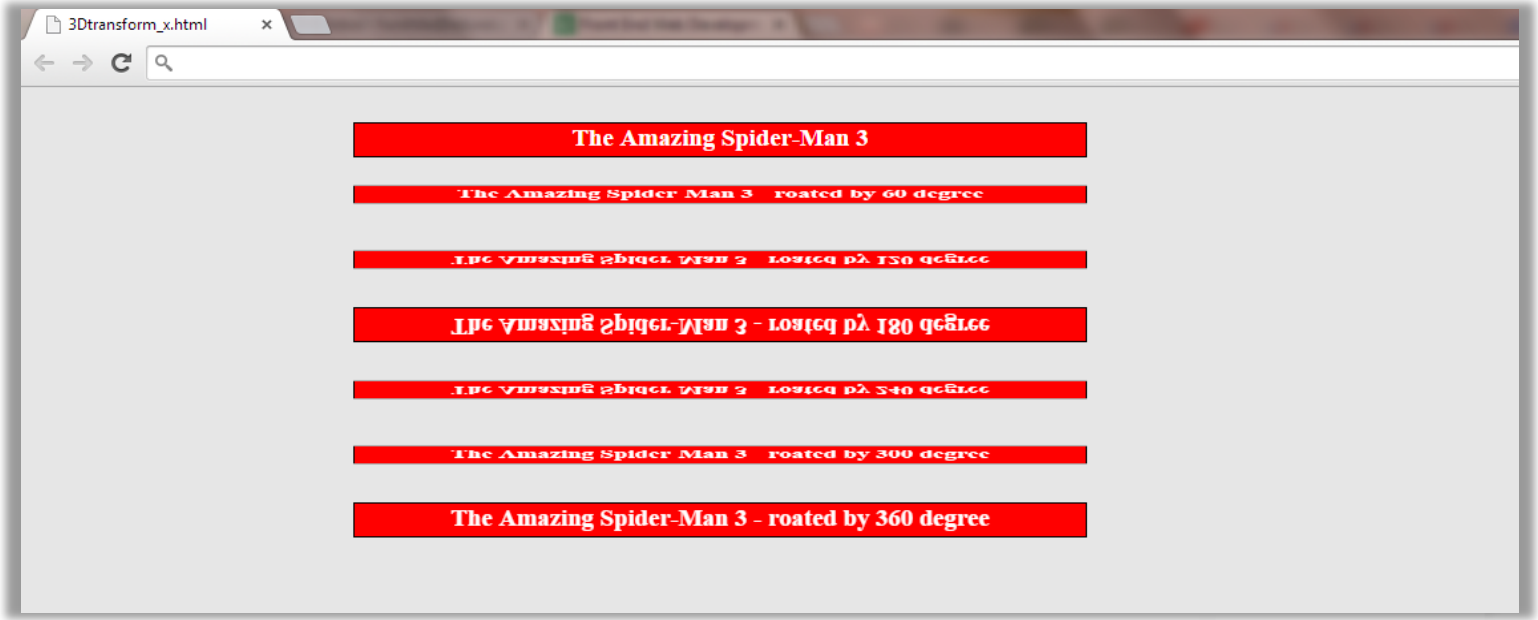
```
div {
    -webkit-transform: rotateX(30deg); /* Safari */
    transform: rotateX(30deg);
   }
```

```
div {
    -webkit-transform: rotateY(90deg); /* Safari */
    transform: rotateY(90deg);
}
```

```
div {
    -webkit-transform: rotateZ(60deg); /* Safari */
    transform: rotateZ(60deg);
}
```

CSS3 – 3D Transformation – X-axis

→ Now let us see the code to create a 3D transform, horizontally



CSS3 – 3D Transformation – Y-axis

→ Now let us see the code to create a 3D transform, vertically

This type of image rotation can be seen in the e-commerce websites

CSS3 – Transition

→ When you want to change the state of an element, you can use the CSS3 transition property

CSS3 transitions allows you to change an element's properties smoothly over a given duration

Here you see a transition of a circle from red to yellow color

CSS3 – Transition (Contd.)

→ Transitions are the effects which changes from one state to another. To generate changes like this transition can be used. We use transitions in all the places where the object is required to change its state in a graphical way

→ It can be done with styles in CSS3 without using JavaScript or flash

→ -webkit-transition: width 2s, height 2s, -webkit-transform 2s;

- » In transitioning, for width it takes 2 seconds, for height it takes 2 seconds and to show the effect of transformation will take 2 seconds

→ -webkit-transform: rotate(360deg);

- » Rotates the elements by 360 degree
- » If the degree is 720 then it takes 360degress x 2. Complete rotation will take place twice
- » If the given degree is 3600, the element will be rotated 10 times

CSS3 – Transition (Contd.)

Example

```
<style>
div {

    width: 100px;
    height: 100px;
    background: DeepPink;
    -webkit-transition: width 2s, height 2s, -webkit-transform 5s;

}

div:hover {
    width: 400px;
    height: 400px;
    -webkit-transform: rotate(3600deg); /* Chrome, Safari, Opera */

}
</style>
```

Output

Before Hovering

After Hovering

CSS3 – Animation

→ Animation is a technique where movements are created using a sequence of images

» In animation, several images are taken together and are displayed one after the other

» To use CSS3 animation, you must first specify some keyframes for the animation

» Keyframes hold what styles the element will have at certain times

CSS3 – Animation (Contd.)

→ With CSS3 animation properties, it is possible to replace flash and JavaScript animations

→ Syntax:

```
-webkit-animation: name 5s;          → Animation is defined under name and it displays in 5 seconds
@-webkit-keyframes name {            → This is the definition of how colors have to change
0%  {background: red;}               → Initially red color is displayed
25%  {background: green;}            → After 25% of the animation, color changes to green
50%  {background: blue;}             → After 50% of animation, color changes to blue
100% {background: yellow;}           → After 100% of animation, color changes to yellow
}
```

→ In animation, several images are taken together and are displayed one after the other

→ Animations are used to display the images to generate very good visual effects

CSS3 – Animation (Contd.)

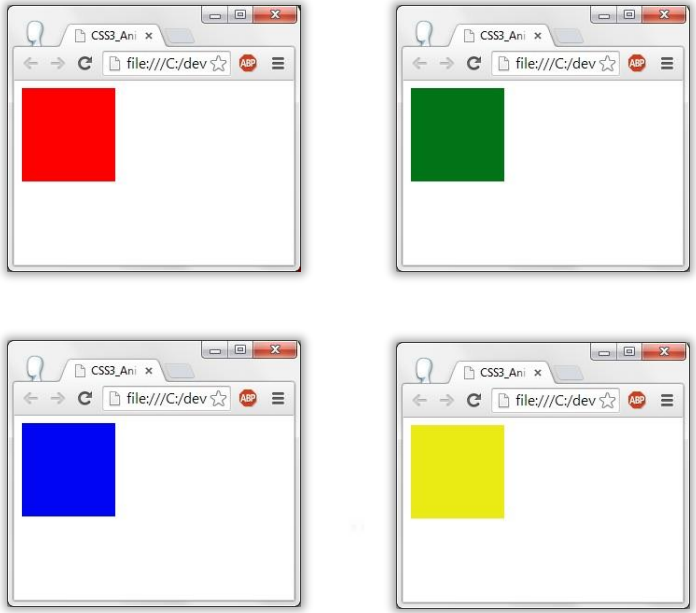
→ In our example, we are 1st displaying the block in red color, then color changes to green and then to red and finally to yellow

The block here does not move. It just changes the color, and the script is executed in 5 seconds

Example

```
<style>
div {
width: 100px;
height: 100px;
background: red;
-webkit-animation: myfirst 5s;
}
@-webkit-keyframes myfirst {
0%   {background: red;}
25%  {background: green;}
50%  {background: blue;}
100% {background: yellow;}
}
</style>
```

Output



CSS3 – Animation (Contd.)

→ Here the block moves from one position to another, forming a pattern. The block is of height and width 100px

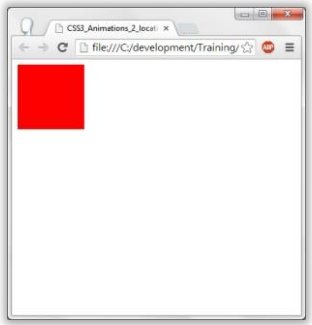
Example

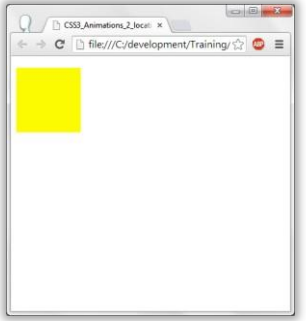
```
<style>
div {
width: 100px;
height: 100px;
background: red;
position:relative;
-webkit-animation: myfirst 5s;
}
@-webkit-keyframes myfirst {
0%   {background: red;left:0px; top:0px;}
25%  {background: green;left:500px; top:0px;}
50%  {background: blue;left:500px; top:500px;}
75%  {background: brown;left:0px; top:500px;}
100% {background: yellow;left:0px; top:0px;}
}
</style>
```

Displays at 0,0

Displays at 500,0. So the box moves from 0,0 to 500,0 as animation and so on

Output





CSS3 – Sprite Sheet Animation

→ Whenever you want to break an animation into different segments, sprite sheet animations is used

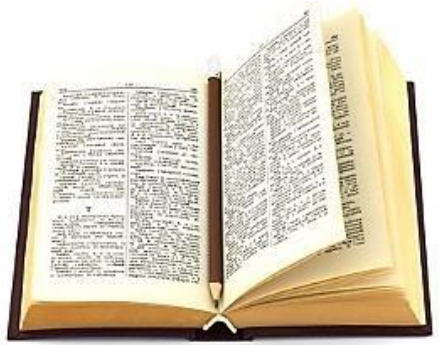
→ Create n number of frames of animation in a sheet (Sprite sheet) for an image (jpg/png)

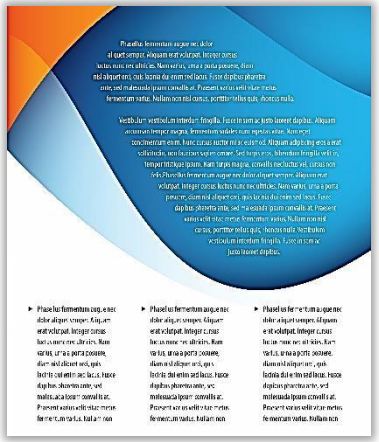
→ Use the steps() function to display the number of frames and the duration of the display

→ In sprite animation, a single image is divided into portions and each portion of the image is displayed one after the other

CSS3 – Multiple Columns

→ All the text content in books, newspapers or a brochure will be printed in multiple columns





CSS3 – Multiple Columns (Contd.)

→ Multiple Column property allows you to print the content in multiple columns

→ The properties of Multiple Columns are as follows:

» Column-count: Number of columns to display

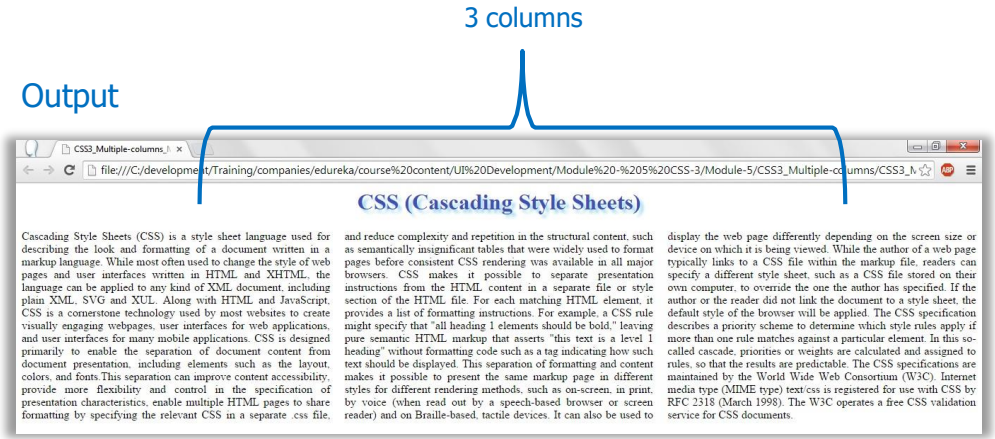
» Column-gap: Indicates the gap size between columns

» Column-rule: Specifies the column separator

Example

```
p{
    -webkit-column-count: 3;
    -webkit-column-gap: 20px;
    text-align: justify;
}
```

Output



CSS3 – Multiple Columns with Image Display

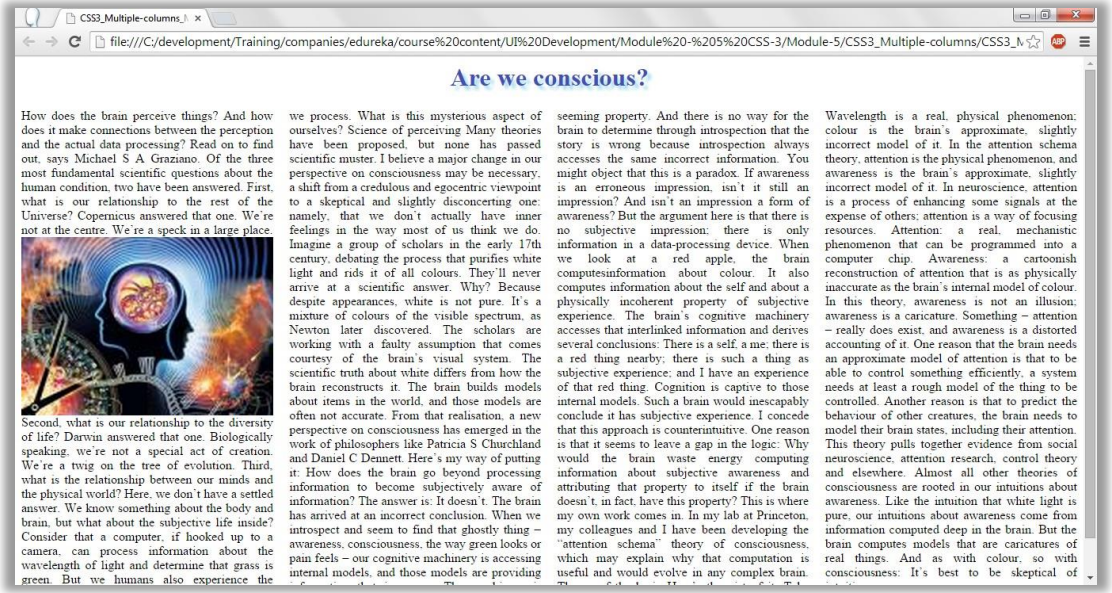
→ Here we are adding an image in-between the text and displaying the text in 4 columns with a column gap of 20px

Example

```
p{
-webkit-column-count: 4;
-webkit-column-gap: 20px;
text-align: justify;
}

img{
display: block;
float: right;
width: 100%;
}
```

Output



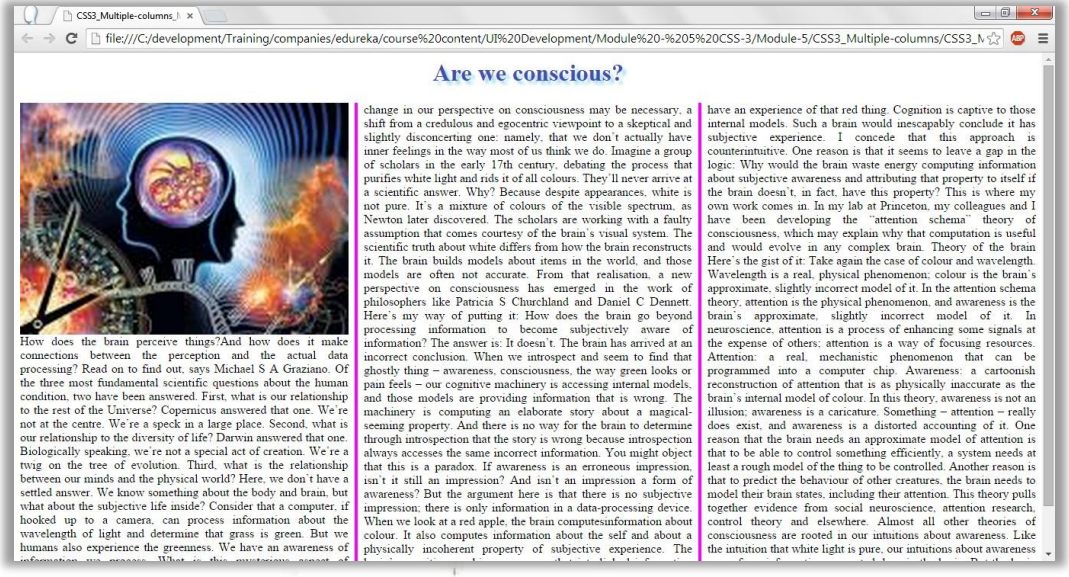
CSS3 – Multiple Columns with Column Rule

→ Here we are adding a column separator, i.e., the column gap is displayed as a pink color line of 4px width

Example

```
p{
-webkit-column-count: 3;
-webkit-column-gap: 20px;
-webkit-column-rule: 4px solid #ff00ff;
text-align: justify;
}
img {
display: block;
float: left;
width: 100%;
}
```

Output



Annie’s Question

Where can we use Multiple column display in a web page?

Annie’s Answer

Ans. To display newspaper on internet as web edition, multiple column display can be used.

CSS3 – New UI

→ CSS3 has new user interface features such as resizing elements, outlines, and box sizing

→ It becomes necessary to resize an image when you want to fit a larger image in a smaller space and display it on your webpage

CSS3 – New UI (Contd.)

→ In CSS3, resizing elements have been introduced

→ It is done through resize property

→ resize: both - Specifies resizing can be done in both horizontal and vertical directions

→ overflow: auto - If the text overflows in the given area then horizontal/vertical scroll bars are added automatically

CSS3 – New UI Resize

Example

```
p{
    border: 2px solid;
    padding: 10px 40px;
    width: 710px;
    resize: both;
    overflow: auto;
}
```

Output

Before Resizing

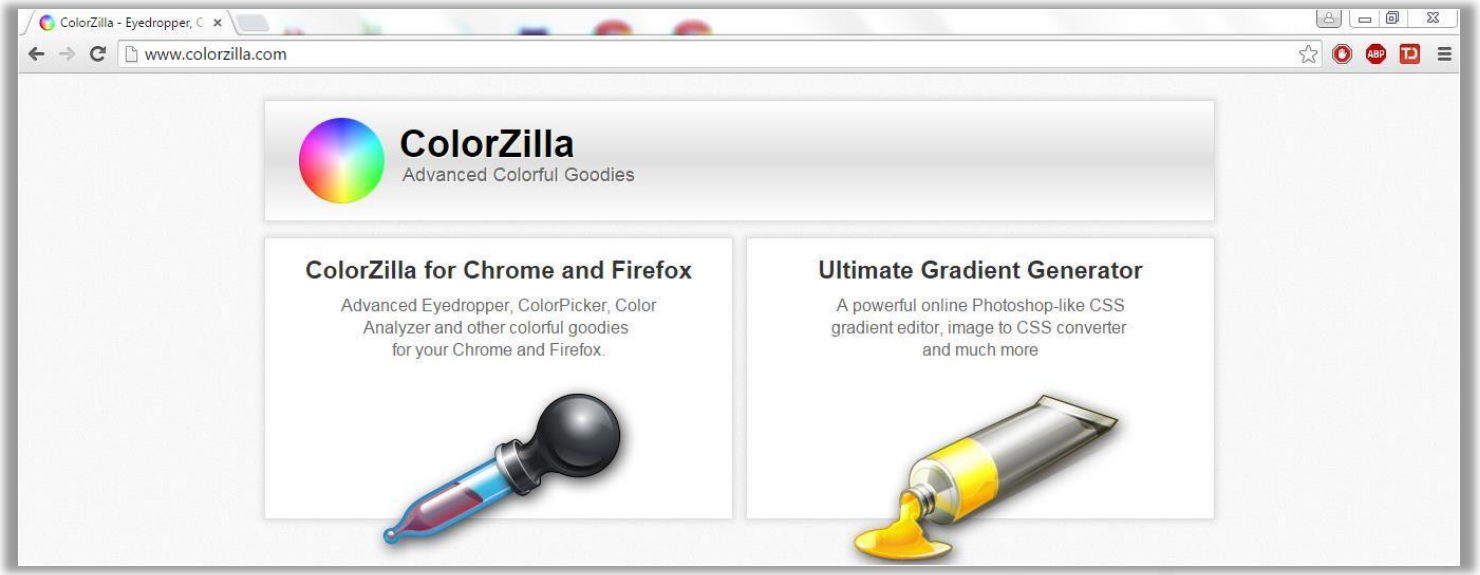
Use this for resizing

After Resizing

Browser Plugins/Extensions

ColorZilla

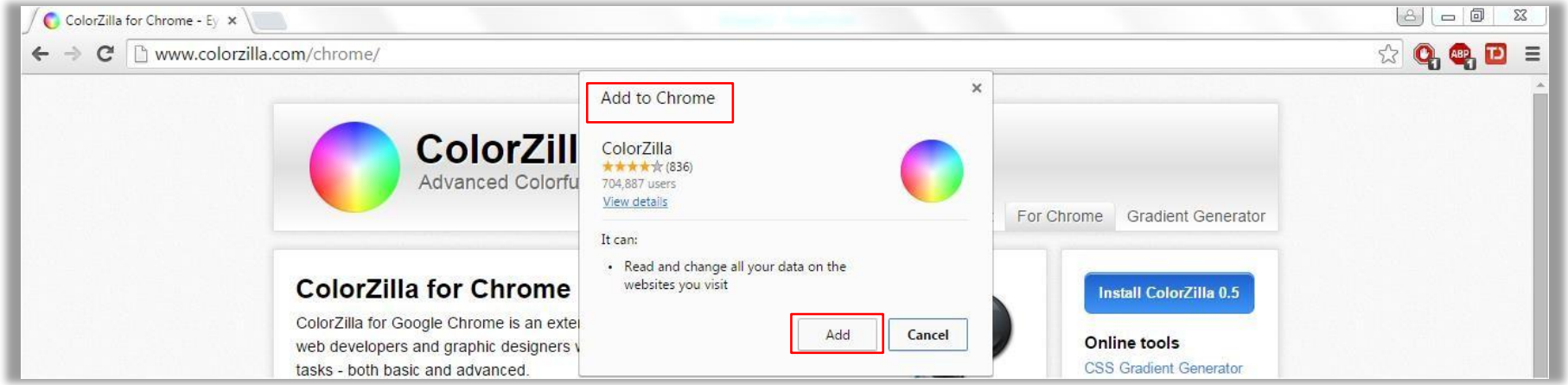
→ There are times when we want to use a particular color that we saw on an another website. "ColorZilla" is a great plugin that can be used to pick up color codes from any webpage



ColorZilla is available for both Chrome and Firefox

Adding ColorZilla to Chrome

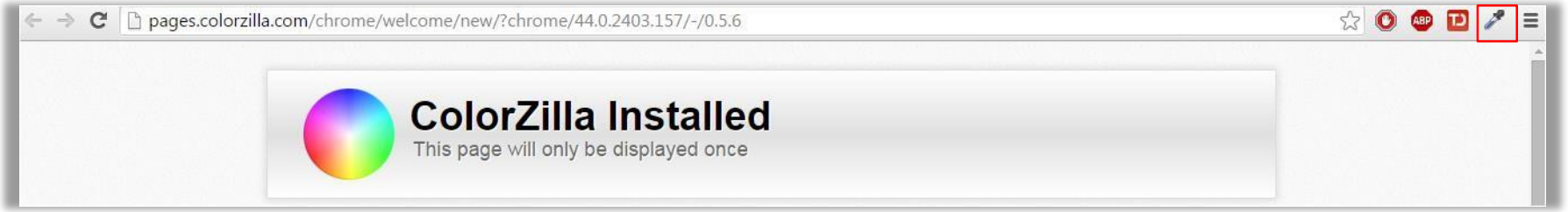
→ To add ColorZilla to Chrome, go to http://www.colorzilla.com/chrome/ and click on Install ColorZilla button



On clicking Install ColorZilla button Chrome will ask you to whether or not add the ColorZilla to Chrome. Just click Add

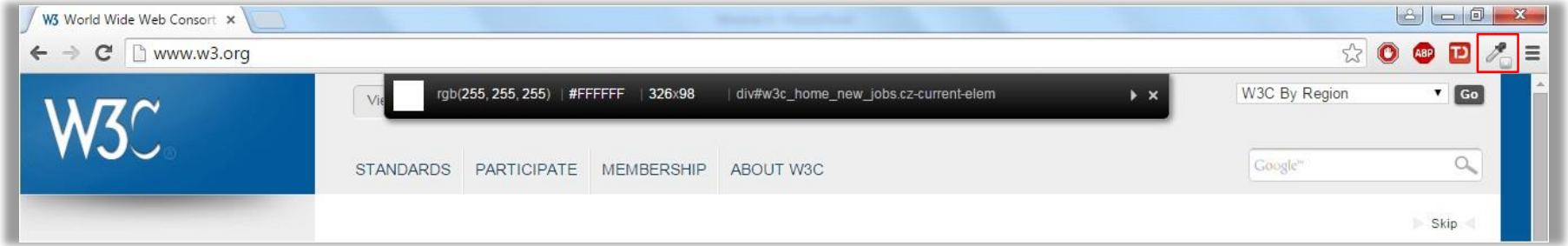
Adding ColorZilla to Chrome

→On clicking add button ColorZilla will be installed. And you will be able to see ColorZilla plugin on top right corner



Using ColorZilla

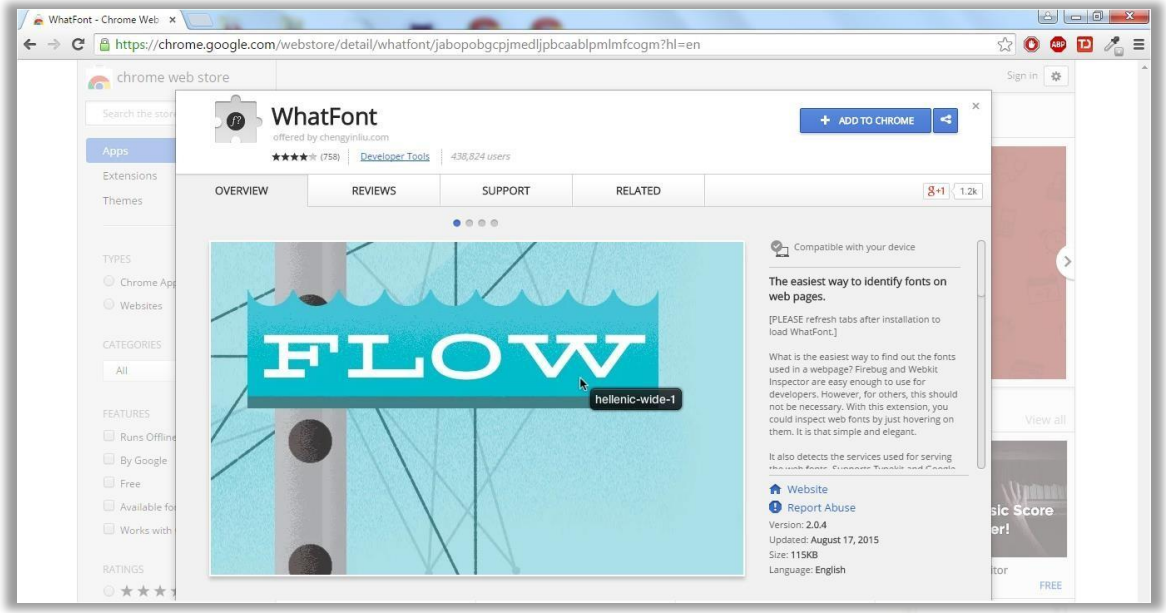
→ To use ColorZilla lets go to http://www.w3.org/ . To pick up colors from any place on the webpage first click on ColorZilla button placed at upper right corner of your browser



Now move your cursor anywhere on the webpage and you will be able to see the color codes in both RGB and HEX form

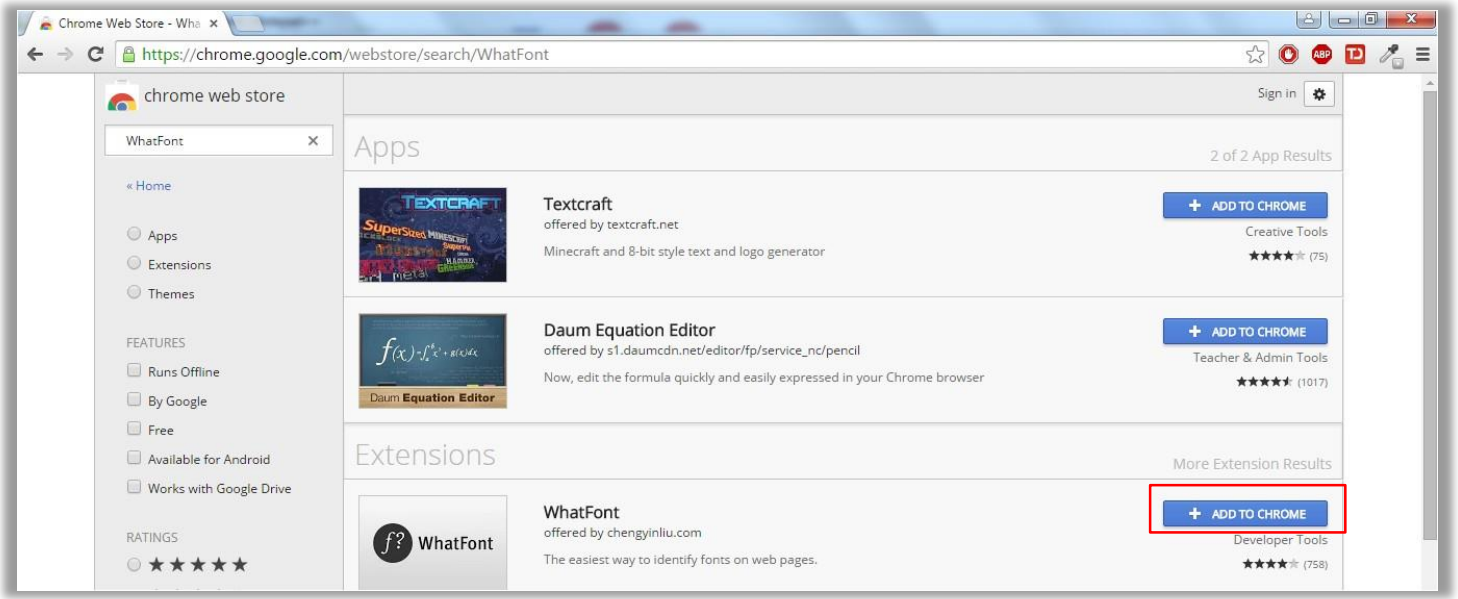
WhatFont

→ WhatFont provides and easy way to identify fonts on web pages



Adding WhatFont to Chrome

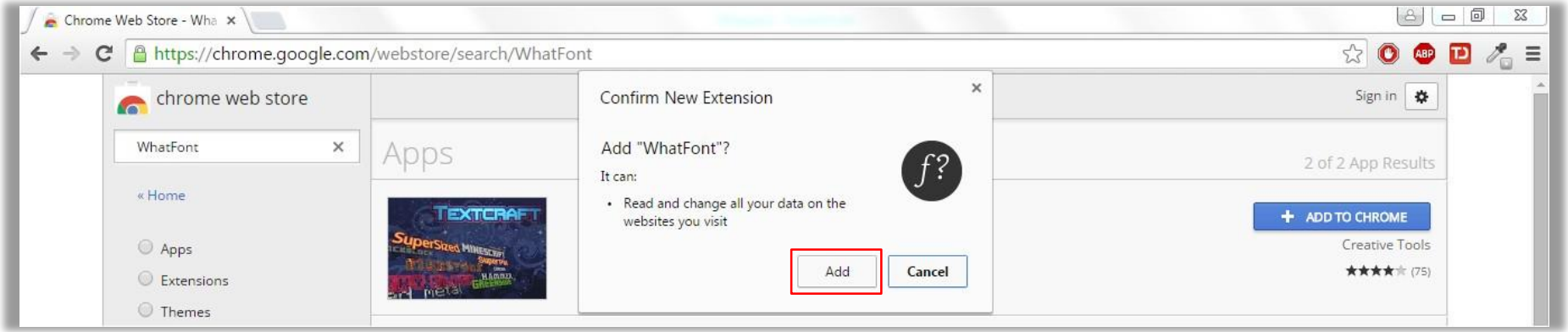
→Go to chrome webstore https://chrome.google.com/webstore/category/apps and search for WhatFont



Click on Add To Chrome button to add WhatFont to Chrome

Adding WhatFont to Chrome (Contd.)

→Click on Add button to allow WhatFont plugin to be added to Chrome

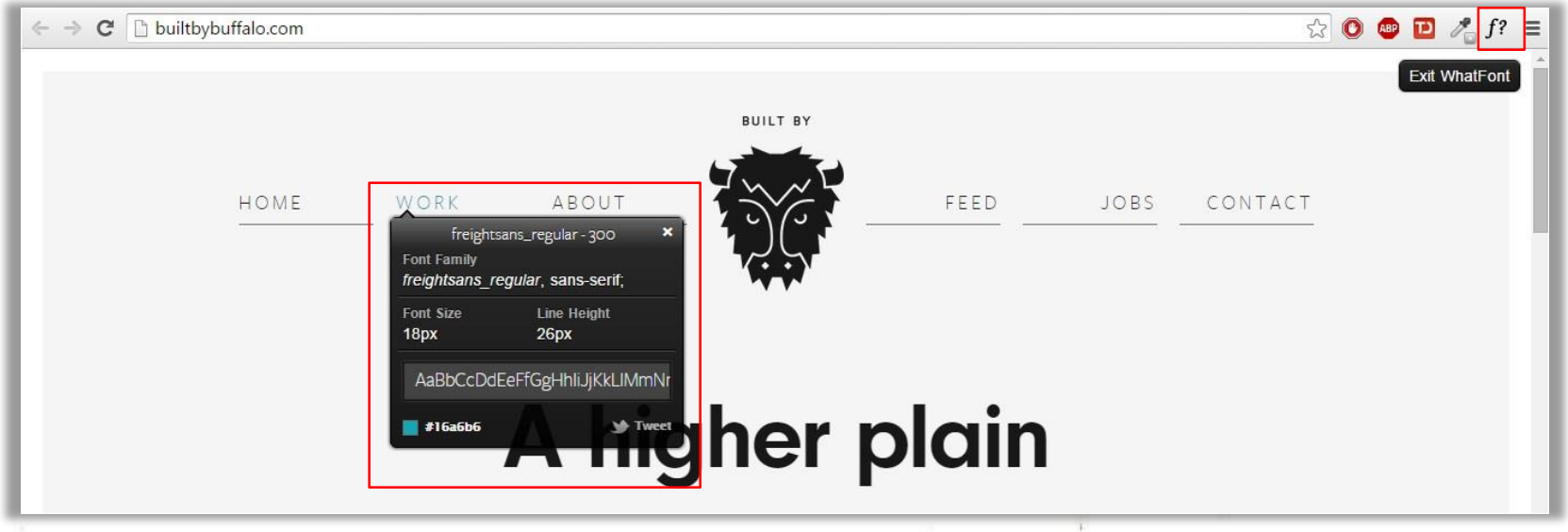


Using WhatFont

→Once WhatFont plugin is added to chrome you will be locate it on upper right corner of Chrome window

Using WhatFont (contd.)

→ Lets see WhatFont in action , go to any website e.g. http://builtbybuffalo.com/. Once webpage is loaded click on WhatFont icon (on upper right corner). Now move your cursor on any text and click on the text to see the detailed information about font-family, font-size etc. as shown below



Once done click on WhatFont icon again to exit WhatFont window

QUESTIONS

Assignment

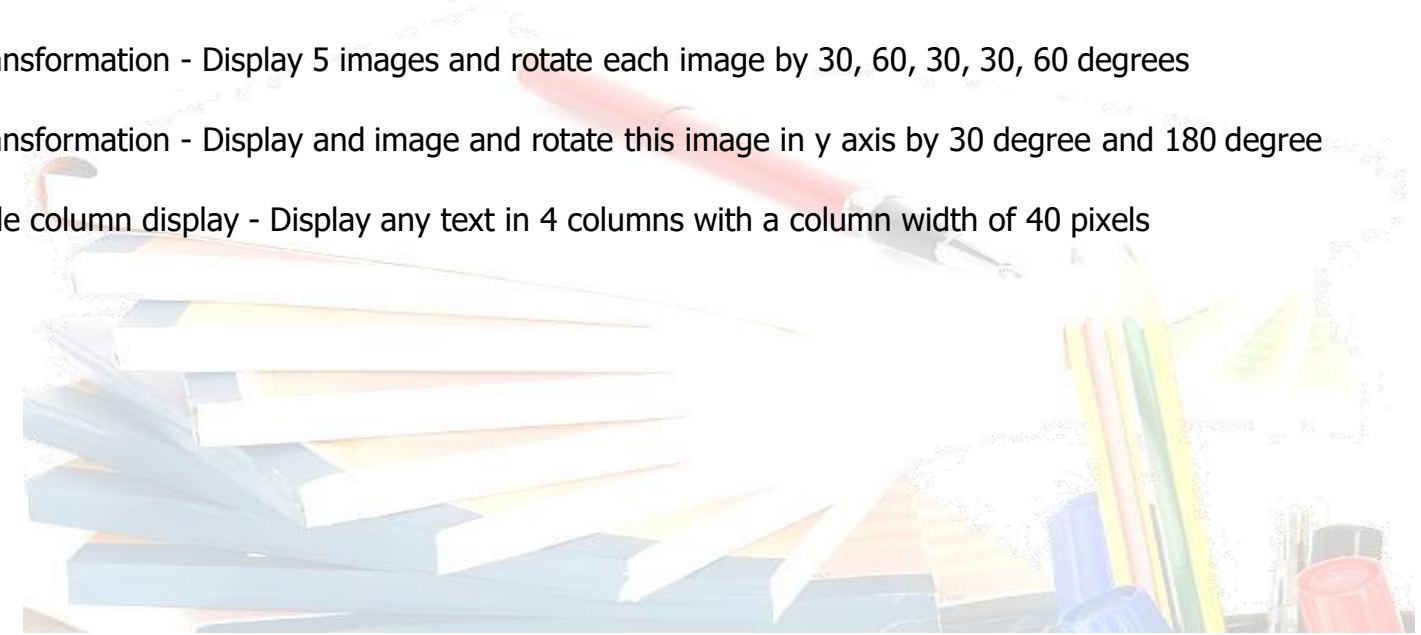
→ Text Shadow - Display some text and give the text shadow effect for it. Provide 1px each for x and y shadow and blur effect as 5 pixels. Display the shadow effect in red color

→ Font - Download any TTF, OTF or WOFF font file and display your name and about you using this font

→ 2d Transformation - Display 5 images and rotate each image by 30, 60, 30, 30, 60 degrees

→ 3d Transformation - Display and image and rotate this image in y axis by 30 degree and 180 degree

→ Multiple column display - Display any text in 4 columns with a column width of 40 pixels



Further Reading

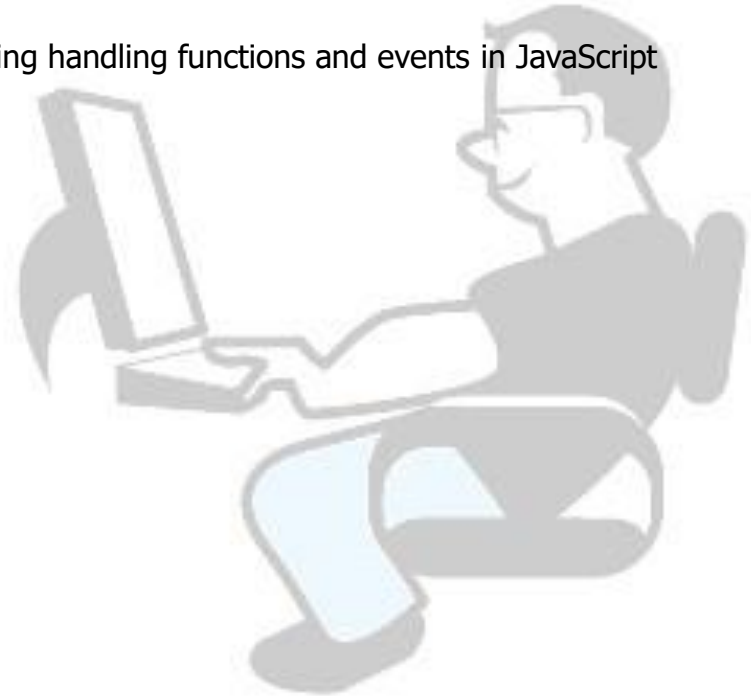
→ http://www.git-tower.com/blog/css3-transforms/

→ http://bootstrapbay.com/blog/css-transitions-buttons/

→ http://www.the-art-of-web.com/css/css-animation/

Pre-work for Next Class

- Explore the different coding techniques in JavaScript
- Go through the various string handling functions and events in JavaScript



Agenda for the next class

In the next module you will be able to:

→ Learn how to write code in JavaScript

→ Features of JavaScript

→ How to handle events and strings in JavaScript

Survey

Your feedback is important to us, be it a compliment, a suggestion or a complaint. It helps us to make the course better!

Please spare few minutes to take the survey after the webinar.


Thank you!