

Lecture 10

CSS for Presentation

1. Formatting Text

In CSS, fonts are specified using font properties such as font family (typeface), size, weight, and font style.

1.1 Font-Family

Use the font-family property to specify a font or list of fonts (known as a font stack) by name as shown in these examples.

```
h { font-family: Arial; }
```

```
p { font-family: "Duru Sans", Verdana, sans-serif; }
```

font-family

Values: one or more font or generic font family names, separated by commas.

Default: depends on the browser.

Generic font family names do not need to be capitalized. Some example are listed below:

serif

Examples: Times, Times New Roman, Georgia

sans-serif

Examples: Arial, Arial Black, Verdana, Trebuchet MS, Helvetica, Geneva

monospace

Examples: Courier, Courier New, and Andale Mono

cursive

Examples: Apple Chancery, Zapf-Chancery, and Comic Sans

Some important syntax requirements:

- All font names, with the exception of generic font families, must be capitalized. For example, use “Arial” instead of “arial”.
- Use commas to separate multiple font names, as shown in the second example.

- Notice that font names that contain a character space (such as Duru Sans) must appear within quotation marks.

Note: CSS provides a list of back-up fonts. If the first specified font is not available, the browser tries the next one, and down through the list until it finds one that works. In the previous example, if the browser does not find Duru Sans, it will use Verdana, and if Verdana is not available, it will substitute some other sans-serif font.

1.2 Specifying Font Size

Use the font-size property to specify the size of the text.

font-size

Values: length unit | percentage | xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger

Default: medium

The text size can specify in several ways:

- Specific size using one of the CSS length units using CSS units of measurement as shown here:

```
h1 { font-size: 1.5em; }
```

When specifying a number of units, be sure the unit abbreviation immediately follows the number, with no extra character space in between:

```
INCORRECT h1 { font-size: 1.5 em; } /*space before the em*/
```

- As a percentage value, sized up or down from the element's default or inherited font size:

```
h1 { font-size: 150%; }
```
- Using one of the absolute keywords (xx-small, x-small, small, medium, large, x-large, xx-large). On most current browsers, medium corresponds to the default font size:

```
h1 { font-size: x-large; }
```
- Using a relative keyword (larger or smaller) to make the text larger or smaller than the surrounding text:

```
strong { font-size: larger; }
```

CSS Units of Measurement

CSS3 provides a variety of units of measurement. They fall into two broad categories: [absolute](#) and [relative](#).

Relative units

Relative units are based on the size of something else, such as the default text size or the size of the parent element.

- px** pixel, considered relative in CSS2.1 because it varies with display resolution
- em** a unit of measurement equal to the current font size
- ex** x-height, approximately the height of a lowercase "x" in the font

The following units are new in CSS3. Browser support may take a while to ramp up.

- rem** root em, equal to the em size of the root element (`html`)
- ch** zero width, equal to the width of a zero (0) in the current font and size
- vw** viewport width unit, equal to 1/100 of the current viewport (browser window) width
- vh** viewport height unit, equal to 1/100 of the current viewport height
- vm** viewport minimum unit, equal to the value of vw or vh, whichever is smaller

Absolute units

Absolute units have predefined meanings or real-world equivalents.

- px** pixel, defined as an absolute measurement equal to 1/96 of an inch in CSS3
- pt** points (1/72 inch in CSS2.1)
- pc** picas (1 pica = 12 points)
- mm** millimeters
- cm** centimeters
- in** inches

1.3 Font Weight (Boldness)

The font-weight property is used to adjust the boldness text (text in bold).

font-weight

Values: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

Default: normal

```
p { font-weight: bold; }
```

1.4 Font Style (italics)

The font-style property affects the posture of the text, that is, whether the letter shapes are vertical (normal) or slanted (italic and oblique).

font-style

Values: normal | italic | oblique

Default: normal

```
p { font-style: italic; }
```

1.5 Text Underlines and Decorations

To put a line under, over, or through text, or to turn the underline off under links, then text-decoration property is used.

text-decoration

Values: none / underline / overline / line-through

Default: none

I've got laser eyes.

text-decoration: underline

I've got laser eyes.

text-decoration: overline

~~I've got laser eyes.~~

text-decoration: line-through

Note: The most popular use of the text-decoration property is turning off the underlines that appear automatically under linked text, as shown here:

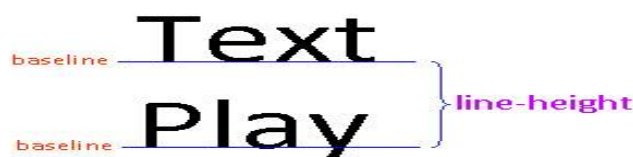
```
a { text-decoration: none; }
```

1.6 Text Line Adjustments

The line adjustments properties deal with whole lines of text rather than the shapes of characters. They allow web authors to format web text with indents, extra space between lines, and different horizontal alignments, similar to print.

- **Line height**

The line-height property defines the minimum distance from baseline to baseline in text. A baseline is the imaginary line upon which the bottoms of characters set.



line-height

Values: number | length measurement | percentage | normal

Default: normal

These examples show three different ways to make the line height twice the height of the font size.

```
p { line-height: 2; }
```

```
p { line-height: 2em; }
```

```
p { line-height: 200%; }
```

Note: When a number is specified alone, as shown in the first example, it acts as a scaling factor that is multiplied by the current font size to calculate the line-height value.

- **Horizontal alignment**

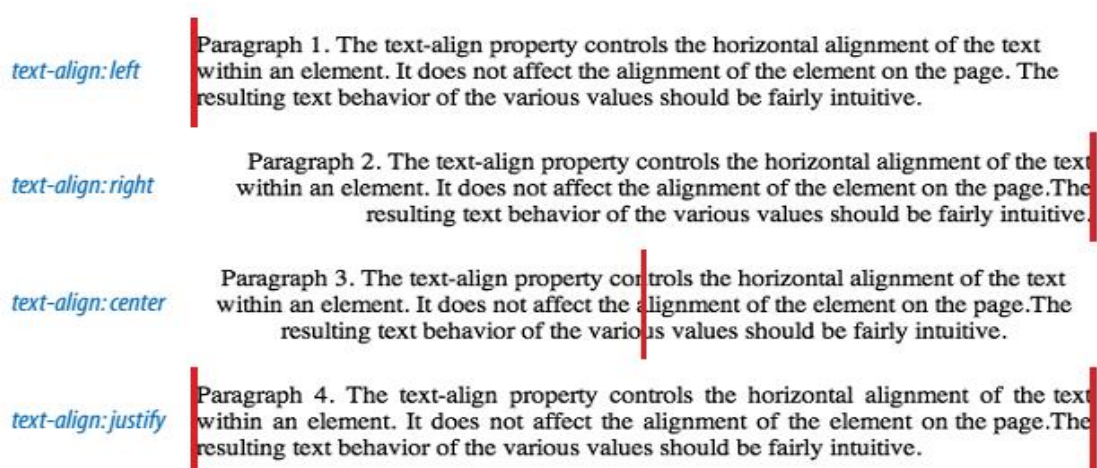
The text is aligning for web pages just as in a word processing or desktop publishing program with the text-align property.

text-align

Values: left | right | center | justify

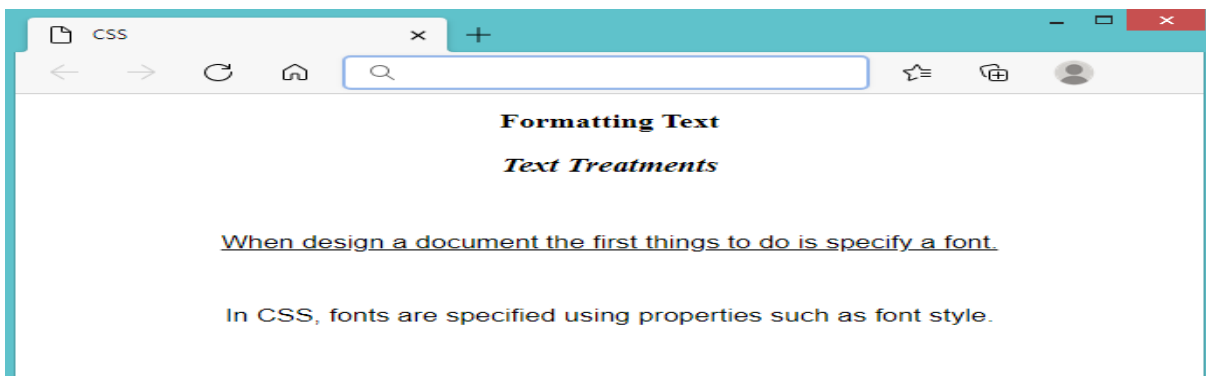
Default: left for languages that read left to right; right for languages that read right to left.

The results of the various text-align values are shown in Figure below.



Example 1: using inline style to formatting text.

```
<html>
<head>
<title> CSS </title>
</head>
<body>
<h2 style=" font-family: Times; font-size:large; text-align:center"> Formatting Text </h2>
<h3 style=" font-family: Times; font-weight: bold; font-style:italic; text-align:center ">
Text Treatments </h3>
<br>
<p style=" font-family: sans-serif; font-weight:lighter; text-decoration:underline; text-align:center"> When design a document the first things to do is specify a font. </p>
<br>
<p style=" font-family: sans-serif; font-weight:lighter; text-align:center">
In CSS, fonts are specified using properties such as font style.</p>
</body>
</html>
```



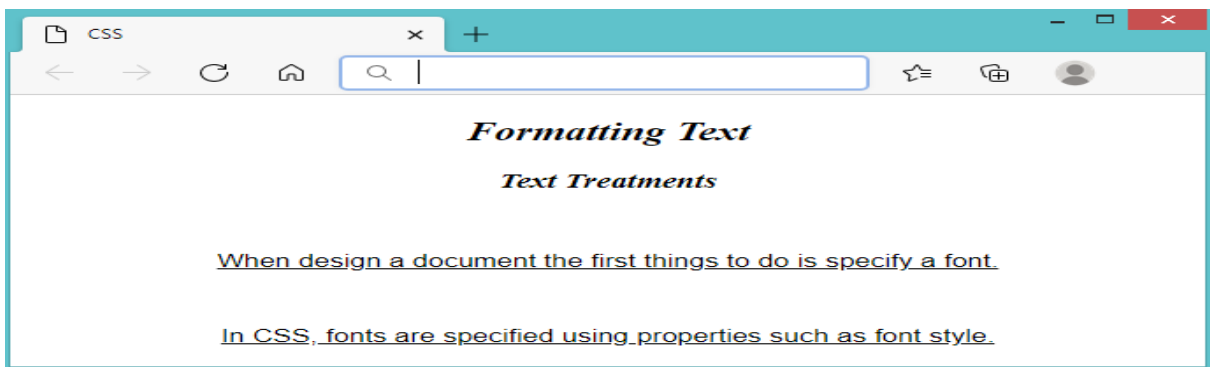
Example 2: using embedded style to formatting text.

```
<html>
<head>
<title> CSS </title>
<style type="text/css">
h2,h3{ font-family:Times;
font-weight: bold;
font-style:italic;
text-align:center;
```

```

}
p{
font-family: sans-serif;
font-weight:lighter;
text-decoration:underline;
text-align:center;
}
</style>
</head>
<body>
<h2> Formatting Text </h2>
<h3> Text Treatments </h3>
<br>
<p> When design a document the first things to do is specify a font. </p>
<br>
<p>In CSS, fonts are specified using properties such as font style.</p>
</body>
</html>

```



2. Changing List Bullets and Numbers

Browsers automatically insert bullets before unordered list items and numbers before items in ordered lists. CSS provides a few properties that allow authors to choose the type and position of the marker, or turn them off entirely.

2.1 Choosing a Marker

Use the list-style-type property to select the type of marker that appears before each list item.

list-style-type

Values: none | disc | circle | square | decimal | decimal-leading-zero | lower-alpha | upper-alpha | lower-latin | upper-latin | lower-roman | upper-roman | lower-greek

Default: disc

Note: There is no way to change the appearance (size, color, etc.) of generated bullets.

2.2 Make own bullets

The developer can also use own image as a bullet using the list-style-image property.

list-style-image

Values: url | none

Default: none

The value of the list-style-image property is the URL of the image that is used as a marker. The list-style-type is set to disc in case the image does not display or the property isn't supported by the browser.

For example:

```
ul {  
list-style-image: url(/images/happy.gif);  
list-style-type: circle;  
font-size: small;  
}
```

Example 3:

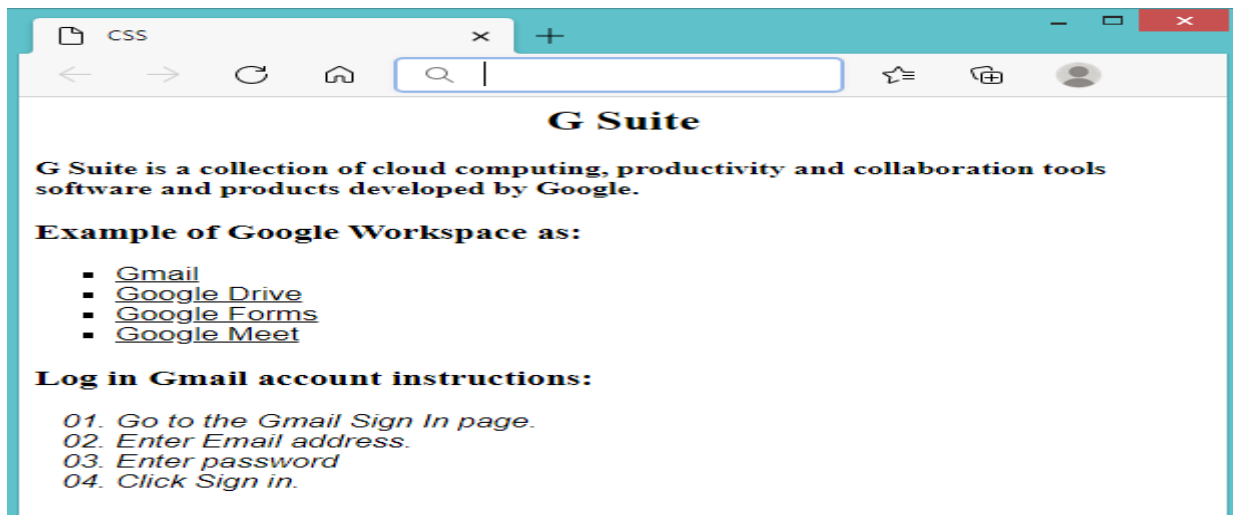
```
<html>  
<head>  
<title> CSS </title>  
<style type="text/css">  
  h2{  
font-family: "Andale Mono";  
font-weight: 900;  
text-align:center;  
  }  
h3,p {
```



```

font-family: "Andale Mono";
font-weight: 900;
text-align:left;
}
ol{
font-family: sans-serif;
font-weight: 400;
font-style:italic;
list-style-type:decimal-leading-zero;
}
ul{
font-family: sans-serif;
font-weight:400;
text-decoration:underline;
list-style-type:square;
}
</style>
</head>
<body>
<h2> G Suite </h2>
<p> G Suite is a collection of cloud computing, productivity and collaboration tools software
and products developed by Google.</p>
<h3>Example of Google Workspace as:</h3>
<ul>
<li> Gmail </li>
<li>Google Drive </li>
<li>Google Forms </li>
<li>Google Meet </li>
</ul>
<h3>Log in Gmail account instructions:</h3>
<ol>
<li> Go to the Gmail Sign In page. </li>
<li>Enter Email address. </li>
<li>Enter password </li>
<li>Click Sign in. </li>
</ol>
</body>
</html>

```



Note: li element can be used instead of **ol** or **ul** element to format the list text.

3. Colors and Backgrounds

3.1 Specifying Color Values

The color of element is changed using the color property. The value of the color property can be a predefined color name or a numeric value describing a specific RGB color (the color model on computer monitors).

color

Values: color value (name or numeric)

Default: depends on the browser and user's preferences

Here are a few examples, all of which make the h1 elements in a document gray:

```
h1 { color: gray; }
```

```
h1 { color: rgb(102,102,102); }
```

```
h1 { color: #666666; }
```

- **Name value**

The most intuitive way to specify a color is to call it by name.

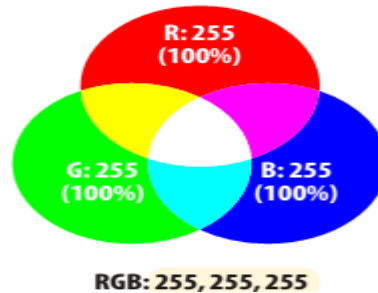
			
Black #000000	Gray #808080	Silver #C0C0C0	White #FFFFFF
			
Maroon #800000	Red #FF0000	Purple #800080	Fuchsia #FF00FF
			
Green #008000	Lime #00FF00	Olive #808000	Yellow #FFFF00
			
Navy #000080	Blue #0000FF	Teal #008080	Aqua #0000FF

- **RGB color values**

Names are easy, but they are limited. The most common way to specify a color is by its RGB value. It gives millions of colors to choose from.

Computers create the colors on a monitor by combining three colors of light: red, green, and blue. This is known as the RGB color model. The amount of light in each color channel is typically described on a scale from 0 to 255. The closer the three values get to 255 the closer the resulting color gets to white as shown in figure below.

The RGB color model



- **Hexadecimal value**

Instead of decimal (base-10 system) these values are written in hexadecimal, or base-16. The structure of the hex RGB value is shown in figure below.



3.2 Background Color

Use the background-color property to apply a background color to any element.

background-color

Values: color value (name or numeric) | transparent

Default: transparent

A background color fills the canvas behind the element that includes the content area, and any padding (extra space) added around the content, extending behind the border out to its outer edge. For example:

```
h2 {  
color: #508C19;  
background-color: #B4DBE6;  
}
```

3.3 Background Images

The background-image property adds a background image to any element. Its primary job is to provide the location of the image file.

background-image

Values: url (location of image) | none

Default: none

The value of background-image is a sort of URL that contains the location of the image. The URL is relative to wherever the CSS rule is at the time. If the rule is written in an embedded style sheet (a style element in the HTML document), then the pathname in the URL should be relative to the location of the HTML file. If the CSS rule is written in an external style sheet, then the pathname to the image should be relative to the location of the .css file.

The examples show background images applied behind a whole page (body) and a single element.

```
body {  
background-image: url("star.gif");  
}
```

```
ul {  
background-image: url("dot.gif");  
}
```

- Repeating background image

In order to appear the a background image more than one time, background-repeat property is used

background-repeat

Values: repeat | repeat-x | repeat-y | no-repeat

Default: repeat

The image can restricted to tiling only horizontally (repeat-x) or vertically (repeat-y). If a background image appears just once, the no-repeat keyword value is used, as shown in these examples.

```
body {  
background-image: url(star.gif);  
background-repeat: no-repeat;  
}
```

```
body {  
background-image: url(star.gif);  
background-repeat: repeat-x;  
}
```

Background position

The background-position property specifies the position of the origin image in the background.

background-position

Values: length measurement | percentage | left | center | right | top | bottom

Default: 0% 0% (same as left top)

4. Borders

A border is simply a line drawn around the content area.

4.1 Border Style

Border style is the most important of the border properties because, if there is no border style specified, the border does not exist. In other words, the border style must declare, or the other border properties will be ignored.

border-style

Values: none | dotted | dashed | solid | double | groove | ridge | inset | outset

Default: none

4.2 Border Width (Thickness)

Use one of the border width properties to specify the thickness of the border.

border-width

Values: length units | thin | medium | thick

Default: medium

4.3 Border Color

Border colors are specified using the border-color property.

border-color

Values: color name or RGB value | transparent

Default: the value of the color property for the element

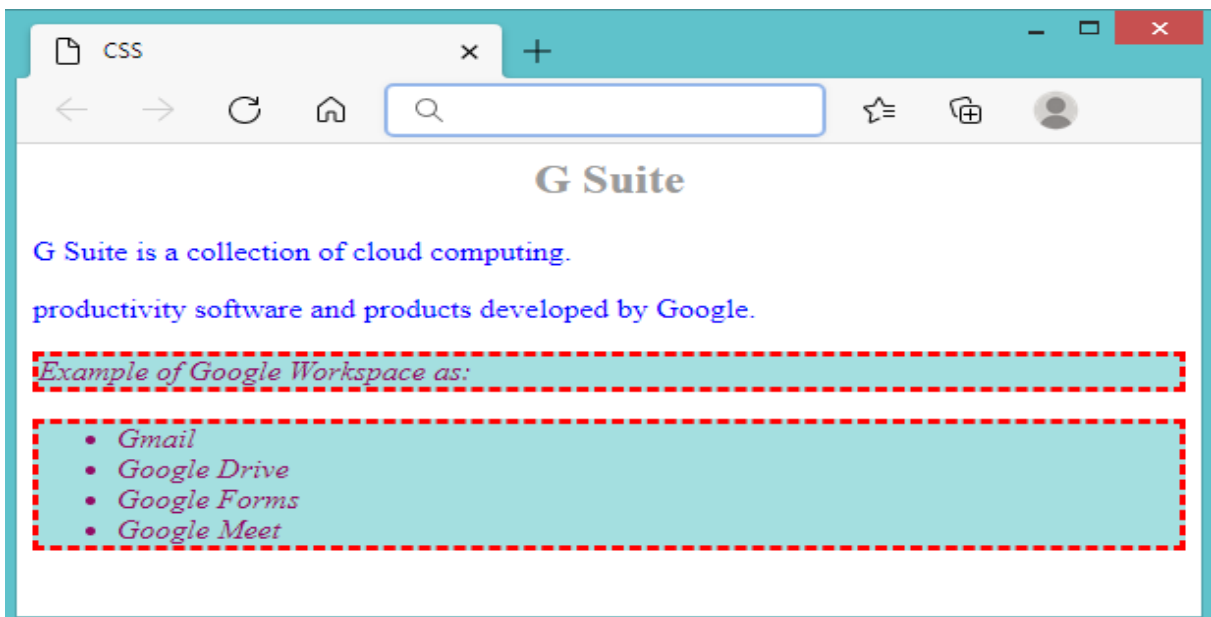
Example 4:

```
<html>
<head>
<title> CSS </title>
<style type="text/css">
h2{
font-weight: 900;
text-align:center;
color:#999999
}
p {
font-weight:400 ;
text-align:left;
color:blue;
}
#EX p,ul{
font-style:italic;
color: rgb(145,10,100);
border-style: dashed;
border-color:red;
```

```

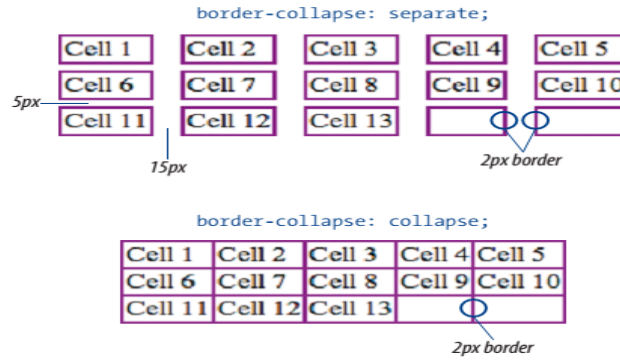
background-color: #A4DFE0;
}
</style>
</head>
<body>
<h2> G Suite </h2>
<p> G Suite is a collection of cloud computing. </p>
<p> productivity software and products developed by Google.</p>
<p> </p>
<div id="EX" >
<p>Example of Google Workspace as:</p>
<ul>
<li> Gmail </li>
<li>Google Drive </li>
<li>Google Forms </li>
<li>Google Meet </li>
</ul>
</div>
</body>
</html>

```



5. Styling Tables

CSS provides two methods for displaying borders between table cells: separated or collapsed. When borders are separated, a border is drawn on all four sides of each cell. In the collapsing border model, the borders of adjacent borders "collapse" so that only one of the borders is visible and the space is removed.



The border-collapse property allows authors to choose which of these border-rendering methods to use.

border-collapse

Values: separate | collapse

Default: separate

Example 5:

```

<html>
<head>
<title> CSS </title>
<style type="text/css">

body{
background-image:url("Image.jpg");
background-repeat :repeat-y ;
}
h2{
font-weight: bold;
text-align:center;
color:red;
}
table{
border-collapse:collapse;
color: red;
background-color: navy;
}
</style>
</head>
<body>
<h2> CSS Colors </h2>
<table border="1">
<tr>
<th colspan="2"> Colors Values</th>
</tr>

```



```

<tr>
<td> RGB Value </td>
<td> Hex Value </td>
</tr>
</table>
</body>
</html>

```



Note: when using the external CSS file the example 5 will be:

The file Web.html

```

<html>
<head>
<title> CSS </title>
</head>
<body>
<link href="WebStyle.css" rel="stylesheet" type="text/css"/>
<h2> CSS Colors </h2>
<table border="1">
<tr>
<th colspan="2"> Colors Values</th>
</tr>
<tr>
<td> RGB Value </td>
<td> Hex Value </td>
</tr>
</table>
</body>
</html>

```

The file WebStyle.css

```

body{
background-image:url("Image.jpg");
background-repeat :repeat-y ;
}

```

```
h2{
font-weight: bold;
text-align:center;
color:red;
}
table{
border-collapse:collapse;
color: red;
background-color: navy;
}
```

5. Page Layout Strategies

Web pages appear on browsers of all sizes, from tiny phone screens to cinema displays. In addition, users can resize their text, which has an impact on the layout of the page. Over time, several standard page layout approaches have emerged that address these issues in various ways:

- Fixed layouts stay put at a specific pixel width regardless of the size of the browser window or text size.
- Fluid (or liquid) layouts: resize proportionally when the browser window resizes.
- Elastic layouts: resize proportionally based on the size of the text.
- Hybrid layouts: combine fixed and scalable areas.

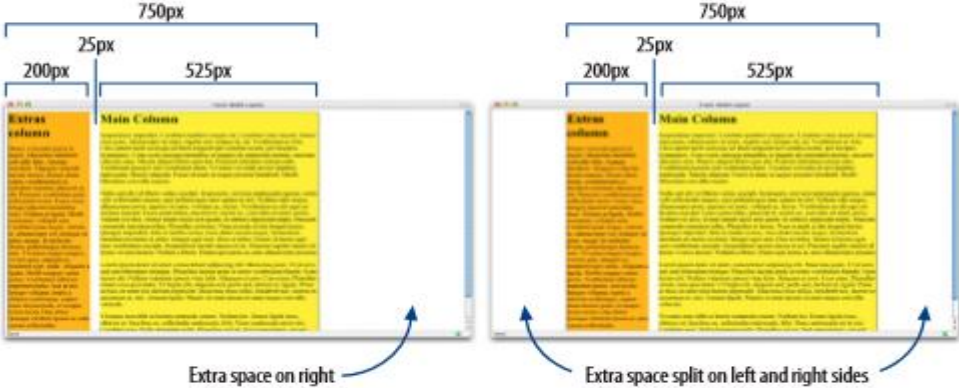
Fixed layouts

Fixed layouts, as the name implies, are created at a specific pixel width as determined by the designer. This layout allows the designer to control the relationship of page elements, alignment, and line lengths. When design a page to be a specific width, it need to decide a couple of things.

First, it is important to pick the width, usually based on common monitor resolutions. Second decide where the fixed-width layout should be positioned in the browser window.

By default, it stays on the left edge of the browser, with the extra space to the right of it. Also it can center the page, splitting the extra space over left and right

margins, which may make the page look as though it better fills the browser window. Following figure shows two fixed width layouts, positioned differently in the browser window. Fixed layout is easier to design and produce. But Content on the right edge will be hidden if the browser window is smaller than the page.



Fluid page design

In fluid page layouts (also called liquid layouts), the page area and columns within the page get wider or narrower to fill the available space in the browser window. There is no attempt to control the width of the content or line breaks; the text is permitted to reflow as required and as is natural to the medium.



Elastic layouts

A third layout approach marries resizable text with predictable line lengths. Elastic layouts expand or contract with the size of the text. If the user makes the text larger, the box that contains it expands proportionally. Likewise, if the user likes her text size very small, the containing box shrinks to fit. The result is that line lengths (in terms of words or characters per line) stay the same regardless of

the text size. This is an advantage over liquid layouts, where line lengths can get too long, and fixed layouts, where very large text may result in awkwardly few characters per line.



Hybrid layouts

Layouts that use a combination of pixel, percentage, and em measurements are sometimes called hybrid layouts. In many scenarios, it makes sense to mix fixed and scalable content areas. For example, you might have a sidebar that contains a stack of ad banners that must stay a particular size. You could specify that sidebar at a particular pixel width and allow the column next to it to resize to fill the remaining space.