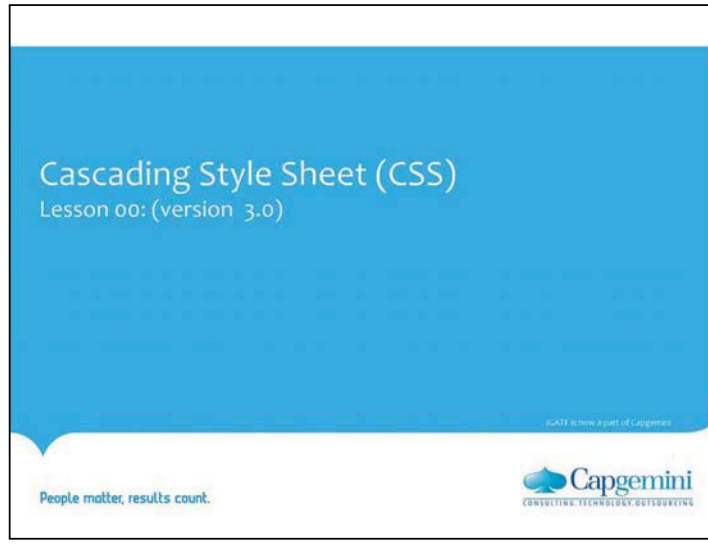
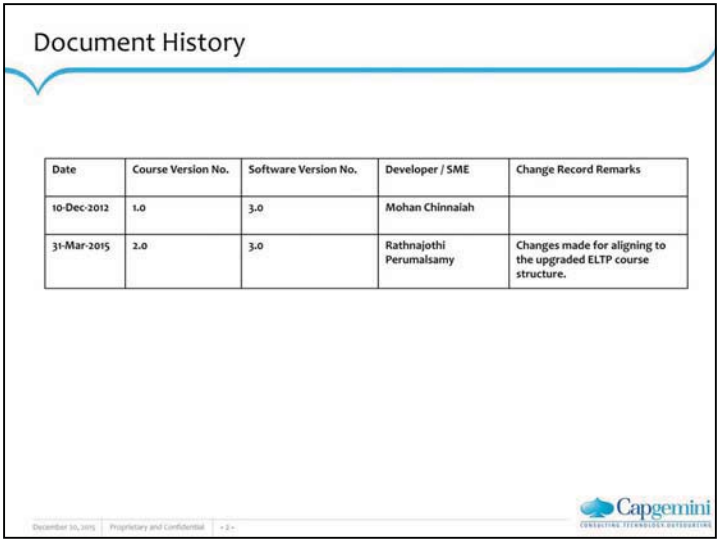




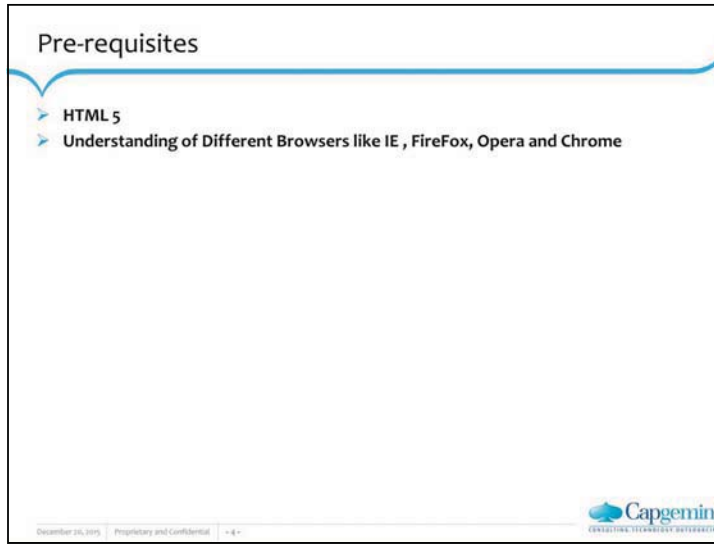
Document History

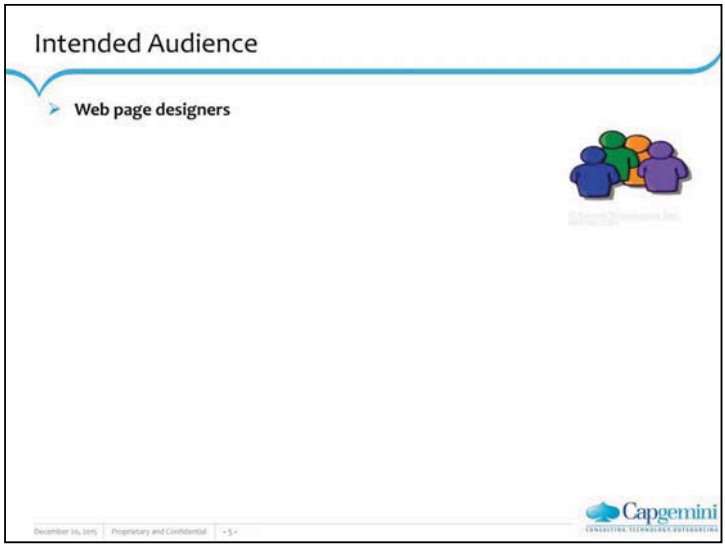
Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
10-Dec-2012	1.0	3.0	Mohan Chinnaiah	
31-Mar-2015	2.0	3.0	Rathnajothi Perumalsamy	Changes made for aligning to the upgraded ELTP course structure.

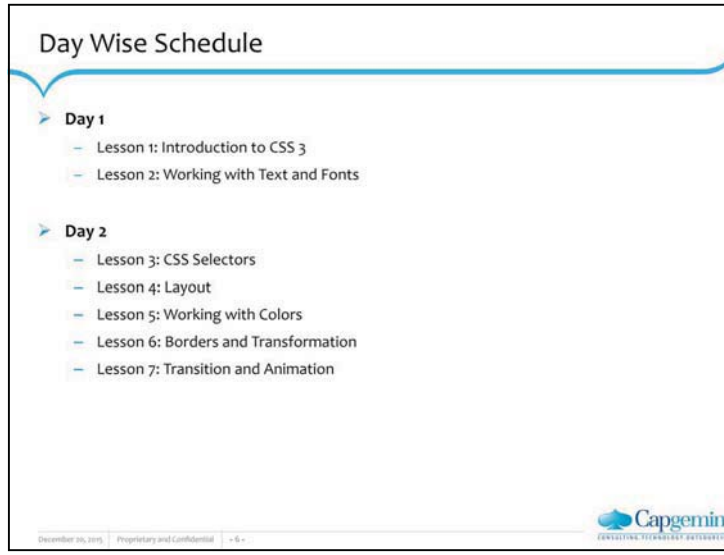
Course Goals and Non Goals

- **Course Goals**
  - Understand Cascading Style Sheet 3.0
  - Understand new features of CSS 3.0
  - Be able to use Custom Fonts
  - Exploring Layouts supported by CSS 3.0
  - Understanding RGBA and HSLA color schemes
  - Animation
- **Course Non Goals**
  - N/A

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING.


Pre-requisites
HTML 5
Understanding of Different Browsers like IE , FireFox, Opera and Chrome
Capgemini


Intended Audience
Web page designers
Capgemini


Day Wise Schedule
Day 1
Lesson 1: Introduction to CSS 3
Lesson 2: Working with Text and Fonts
Day 2
Lesson 3: CSS Selectors
Lesson 4: Layout
Lesson 5: Working with Colors
Lesson 6: Borders and Transformation
Lesson 7: Transition and Animation
Capgemini

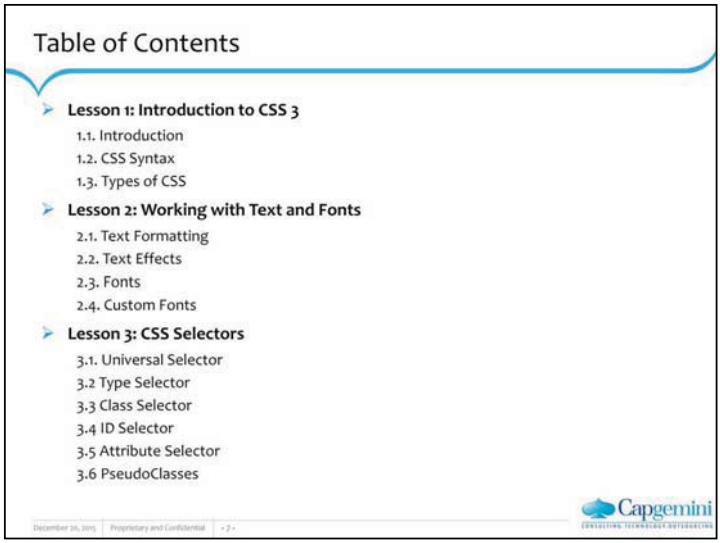


Table of Contents
Lesson 1: Introduction to CSS 3
1.1. Introduction
1.2. CSS Syntax
1.3. Types of CSS
Lesson 2: Working with Text and Fonts
2.1. Text Formatting
2.2. Text Effects
2.3. Fonts
2.4. Custom Fonts
Lesson 3: CSS Selectors
3.1. Universal Selector
3.2 Type Selector
3.3 Class Selector
3.4 ID Selector
3.5 Attribute Selector
3.6 PseudoClasses
Capgemini

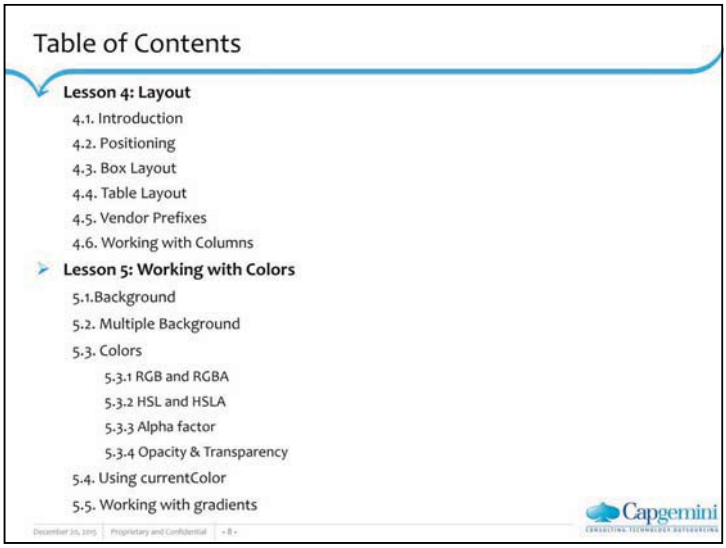


Table of Contents
Lesson 4: Layout
4.1. Introduction
4.2. Positioning
4.3. Box Layout
4.4. Table Layout
4.5. Vendor Prefixes
4.6. Working with Columns
Lesson 5: Working with Colors
5.1.Background
5.2. Multiple Background
5.3. Colors
5.3.1 RGB and RGBA
5.3.2 HSL and HSLA
5.3.3 Alpha factor
5.3.4 Opacity & Transparency
5.4. Using currentColor
5.5. Working with gradients
Capgemini

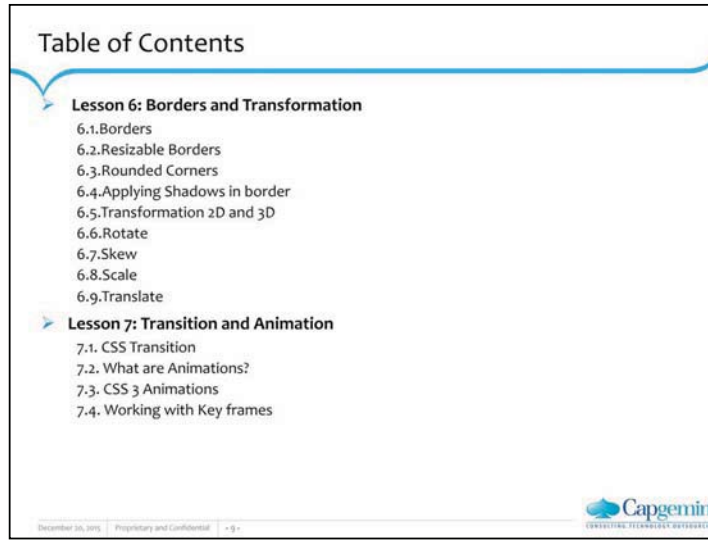
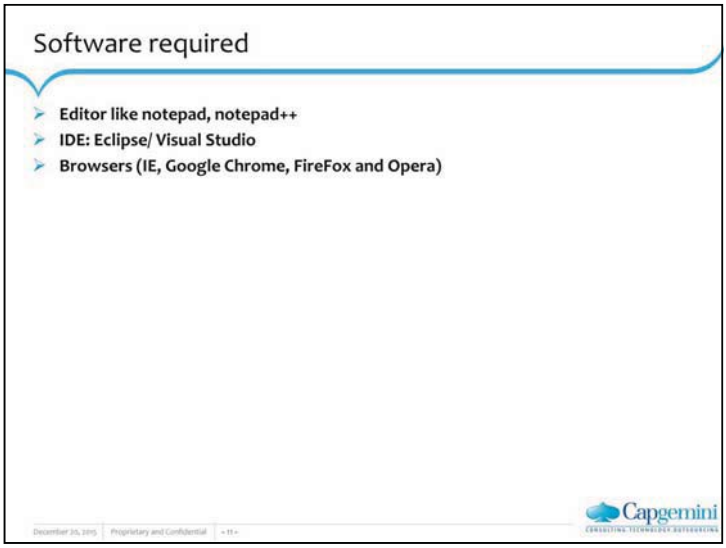
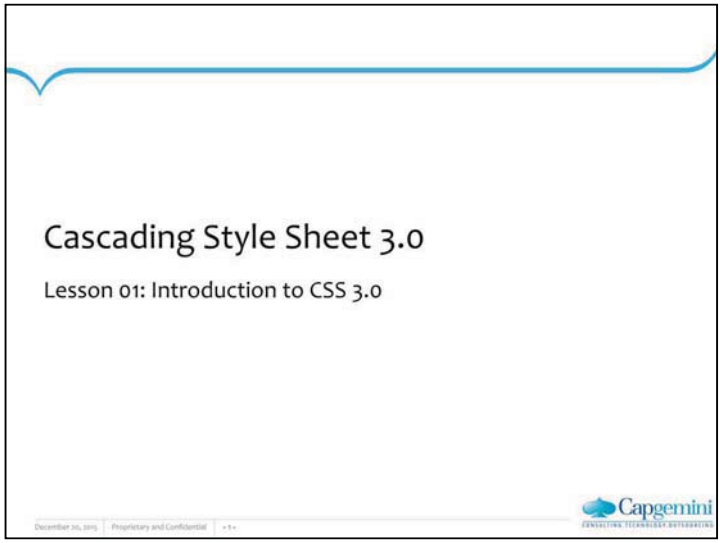


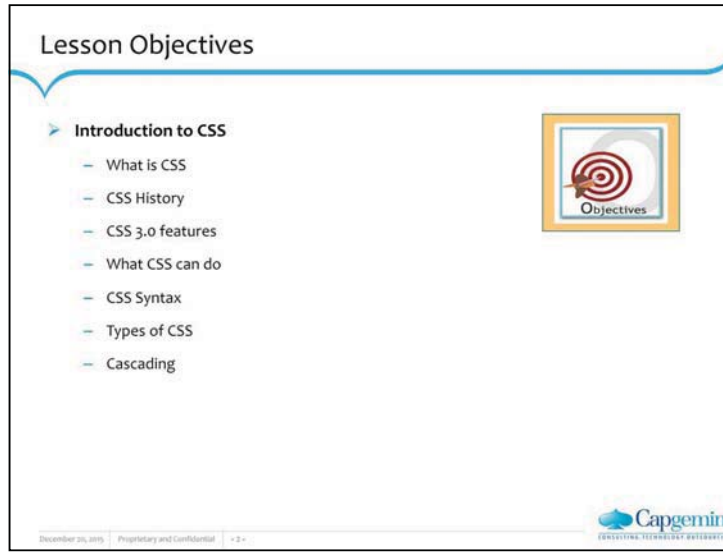
Table of Contents
Lesson 6: Borders and Transformation
6.1.Borders
6.2.Resizable Borders
6.3.Rounded Corners
6.4.Applying Shadows in border
6.5.Transformation 2D and 3D
6.6.Rotate
6.7.Skew
6.8.Scale
6.9.Translate
Lesson 7: Transition and Animation
7.1. CSS Transition
7.2. What are Animations?
7.3. CSS 3 Animations
7.4. Working with Key frames
Capgemini

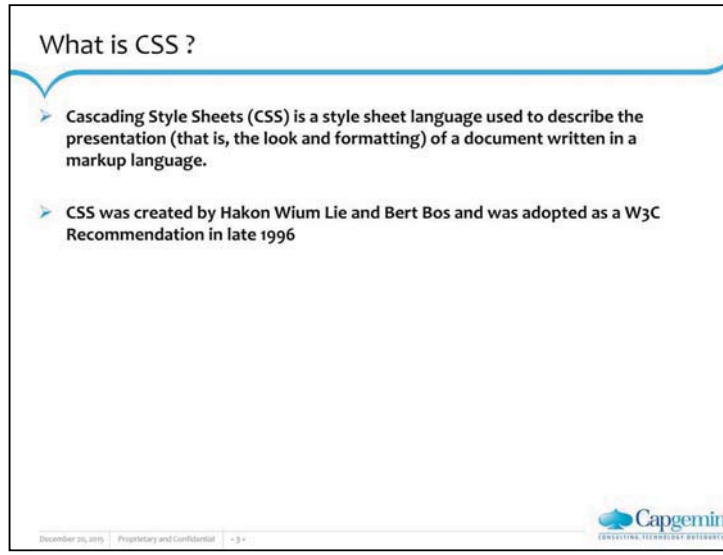
References

- W3 Schools
- Sitepoint


Software required
Editor like notepad, notepad++
IDE: Eclipse/ Visual Studio
Browsers (IE, Google Chrome, FireFox and Opera)
Capgemini


Cascading Style Sheet 3.0
Lesson 01: Introduction to CSS 3.0
Capgemini


Lesson Objectives
Introduction to CSS
What is CSS
CSS History
CSS 3.0 features
What CSS can do
CSS Syntax
Types of CSS
Cascading
Objectives
Capgemini

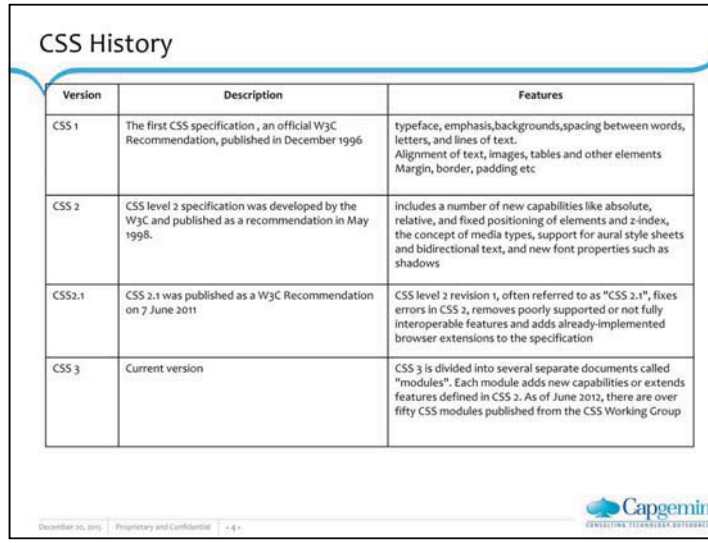


Cascading Style Sheet:

- CSS allows complete and total control over the style of a hypertext document
- A standards-based method for controlling the look and feel of HTML content.
- Comprised of Rules to control elements in the document.
- Designed to separate formatting from the content while being flexible and scalable

What is a Style Sheet?

- Style sheets define how to display HTML elements.
- Style sheets (SS) provide a means for web authors to separate the appearance of web pages from the content.
- Style sheets are an accepted standard on the W3C. The standards are referred to as Cascading Style Sheets 1 (CSS1) and Cascading Style Sheets 2 (CSS2).



CSS History

Version	Description	Features
CSS 1	The first CSS specification , an official W3C Recommendation, published in December 1996	typeface, emphasis,backgrounds,spacing between words, letters, and lines of text. Alignment of text, images, tables and other elements Margin, border, padding etc
CSS 2	CSS level 2 specification was developed by the W3C and published as a recommendation in May 1998.	includes a number of new capabilities like absolute, relative, and fixed positioning of elements and z-index, the concept of media types, support for aural style sheets and bidirectional text, and new font properties such as shadows
CSS2.1	CSS 2.1 was published as a W3C Recommendation on 7 June 2011	CSS level 2 revision 1, often referred to as "CSS 2.1", fixes errors in CSS 2, removes poorly supported or not fully interoperable features and adds already-implemented browser extensions to the specification
CSS 3	Current version	CSS 3 is divided into several separate documents called "modules". Each module adds new capabilities or extends features defined in CSS 2. As of June 2012, there are over fifty CSS modules published from the CSS Working Group

CSS 1 Features:

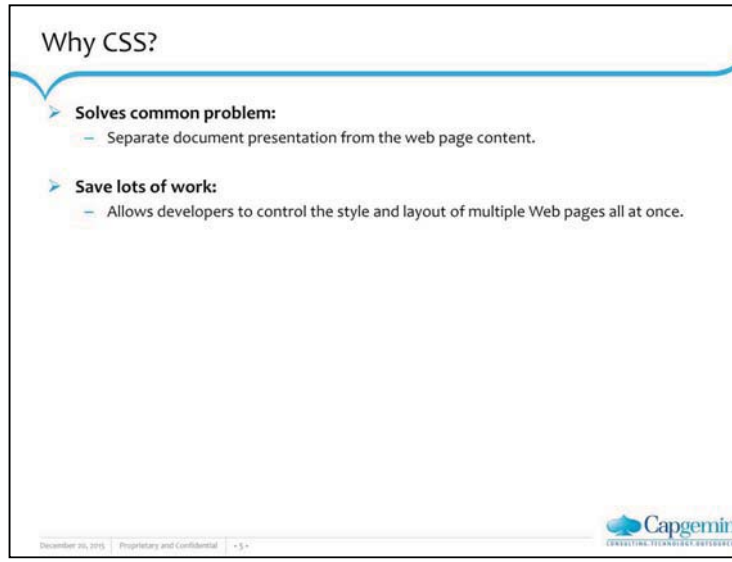
- Font :properties such as typeface and emphasis
- Color of text, backgrounds, and other elements
- Text attributes such as spacing between words, letters, and lines of text
- Alignment of text, images, tables and other elements
- Margin, border, padding, and positioning for most elements
- Unique identification and generic classification of groups of attribute

CSS 2 Features:

CSS level 2 specification was developed by the W3C and published as a recommendation in May 1998. A superset of CSS 1, CSS 2 includes a number of new capabilities like absolute, relative, and fixed positioning of elements and z-index, the concept of media types, support for aural style sheets and bidirectional text, and new font properties such as shadows.

CSS 2.1 Features:

CSS level 2 revision 1, often referred to as "CSS 2.1", fixes errors in CSS 2, removes poorly supported or not fully interoperable features and adds already-implemented browser extensions to the specification



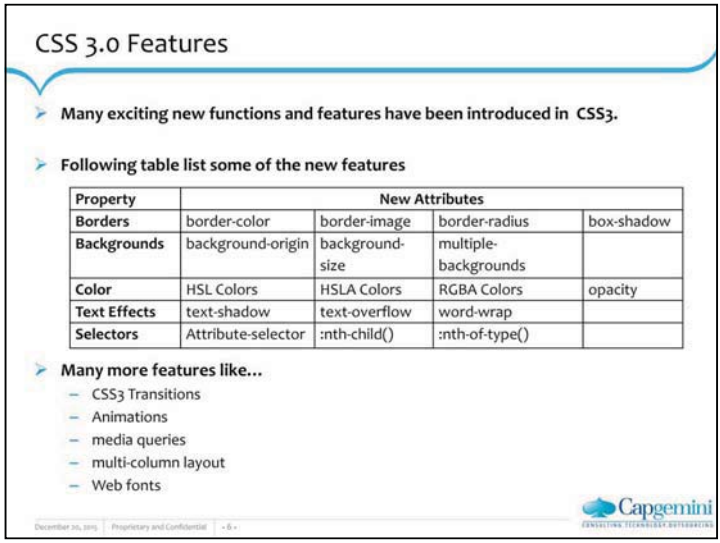
Why use CSS?

Styles solve a common problem : HTML tags were originally designed to define the document content. They were supposed to say "This is a header", "This is a paragraph", "This is a table", by using tags like <h1>, <p>, <table>, and so on. Browser was to take care of the layout of the document without using any formatting tags.
Two major browsers - Netscape and Internet Explorer - continued to add new HTML tags and attributes (like the <font> tag and the color attribute) to the original HTML specification. Subsequently, it became more difficult to create HTML documents with content clearly separate from the presentation layout. To solve this problem, W3C, the non-profit, standard setting consortium responsible for standardizing HTML, created STYLES in addition to HTML.

Style Sheets Save a Lot of Work

Styles in HTML define how HTML elements are displayed, just like the *bold tag.* Styles are saved in files external to your HTML documents. External style sheets allow you to change the appearance and layout of all pages in your website. Simply, edit a single CSS document. If you have ever had to change the font or color of all the headings in all your Web pages, you will understand how CSS can save you a lot of work.

CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. As a Web developer you can define a style for each HTML element and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in the Web are updated automatically.



CSS 3 has introduced many features using which we can much more powerful and flexible websites. Some of the new features of CSS 3 are as follows:

Border Radius: Creating rounded corners in web design isn't always the easiest of things to accomplish. Thanks to the power of CSS3, it has since become one of the more popular and easier techniques to implement. By taking advantage of the **border-radius** property, you can easily round off those corners in seconds

Box Shadow: Creating box shadows is another pretty cool example for adding some stylish elements to your web designs. The best part being the fact that it is completely executed without the use of images. There are even ways to add multiple box shadows to your rounded corners, allowing for the possibility of creating some very slick designs

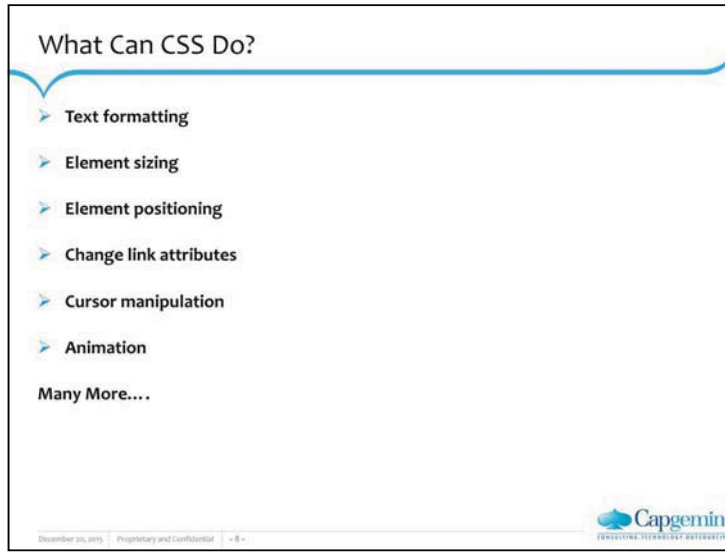
Multiple Background Images: Another cool example of CSS3 is the ability to apply multiple backgrounds to a single DIV without having to create extra child DIV's whose only purpose is to support an image

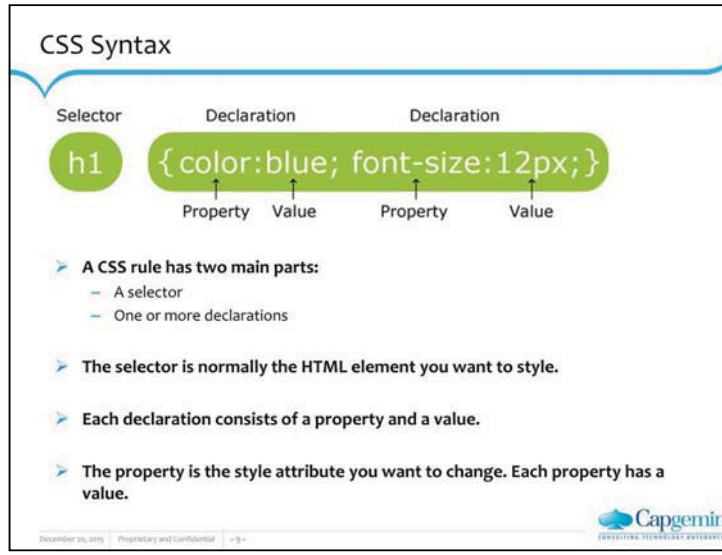
Text Shadow: You know how easy it is to double click a layer in Photoshop and say hey, I want to add a quick drop shadow to that text? This may be even easier than that. You aren't just restricted to just one shadow either. By combing multiple text shadows of varying colors, the possibilities are endless.

@Font-Face: With this feature we can include custom fonts into our web pages. We can now begin to take advantage of various other fonts, whether installed on the readers computer or not, assuming that they can be pulled via an online directory. Just upload the desired font to your server and pull it via the @font-face feature.

Multi-column layout: W3C offers a new way to arrange text "news-paper wise", in columns. Multi-column layout is actually a module on its own. It allows a web developer to let text be fitted into columns

CSS3 Animations: Traditionally, the Web was a very static place. Achieving animations was not really possible unless we use JavaScript, animated GIFs and Flash . But With CSS3, we can create animations, which can replace animated images, Flash animations, and Java Scripts in many web pages.


What Can CSS Do?
Text formatting
Element sizing
Element positioning
Change link attributes
Cursor manipulation
Animation
Many More....
Capgemini

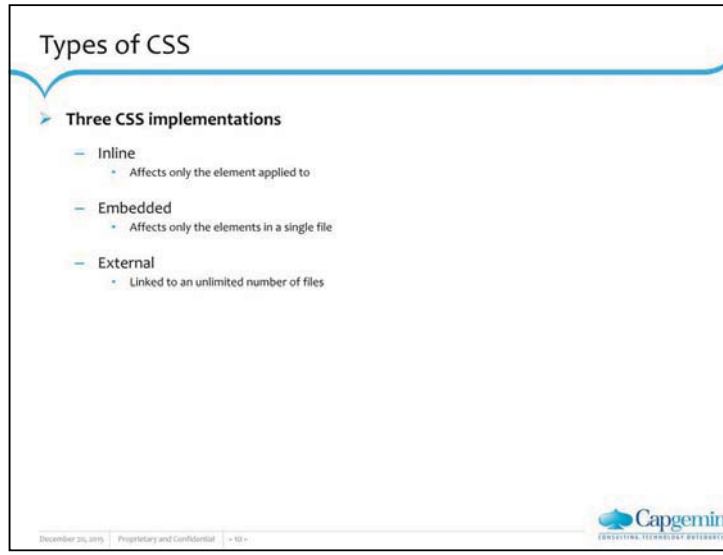


A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets:

p {color:red;text-align:center;}

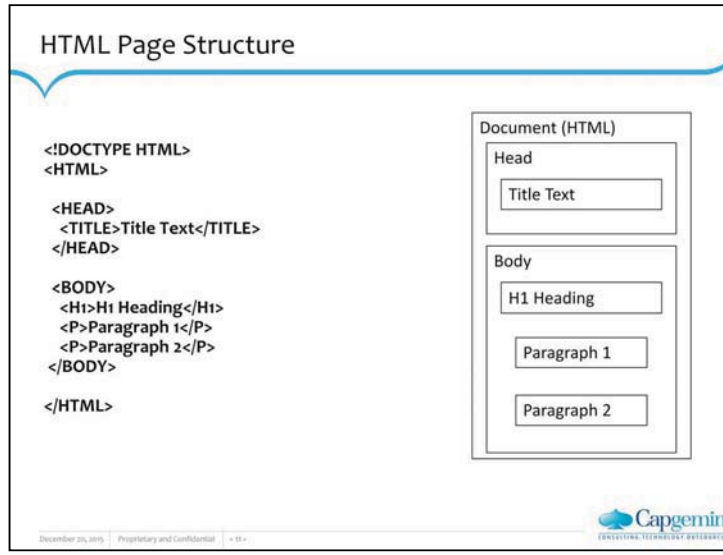
To make the CSS more readable, you can put one declaration on each line, like this:

```
p
{
color:red;
text-align:center;
}
```

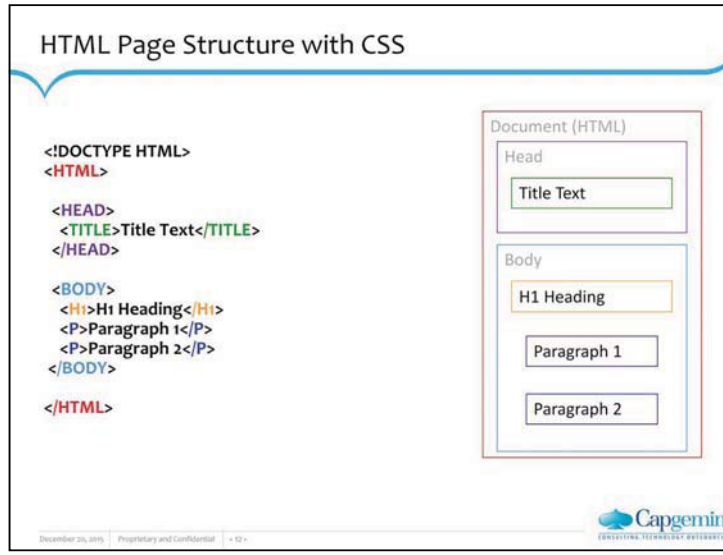


Types of CSS:

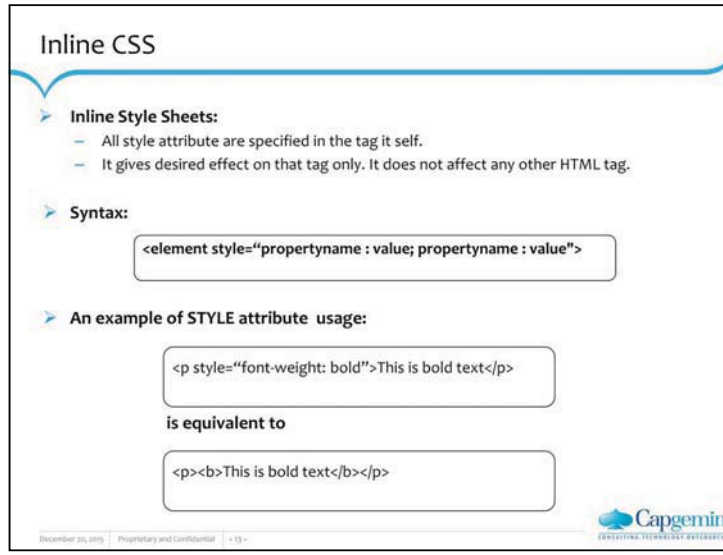
- Inline: Style sheet definition only applies to the tag contents that contain it. It is used to control a single tag element. Each tag does not need to have its style defined as it inherits from its parent.
- Embedded: Embedded style sheets are placed within HTML code of the page they are to be applied to. Style sheet syntax comes between opening and closing <STYLE> tags. These tags are placed either in the <HEAD> section or between the </HEAD> and <BODY> tags.
- Linked: Linked style sheets exist as separate files that are linked to a page with the <LINK> tag. They have the css extension and are referenced with a URL. Inside the css file, style attributes are contained within opening and closing <STYLE> tags. Placing a single <LINK> tag within the <HEAD> tags links the page that needs these styles.



The above given HTML document content is not formatted using CSS.

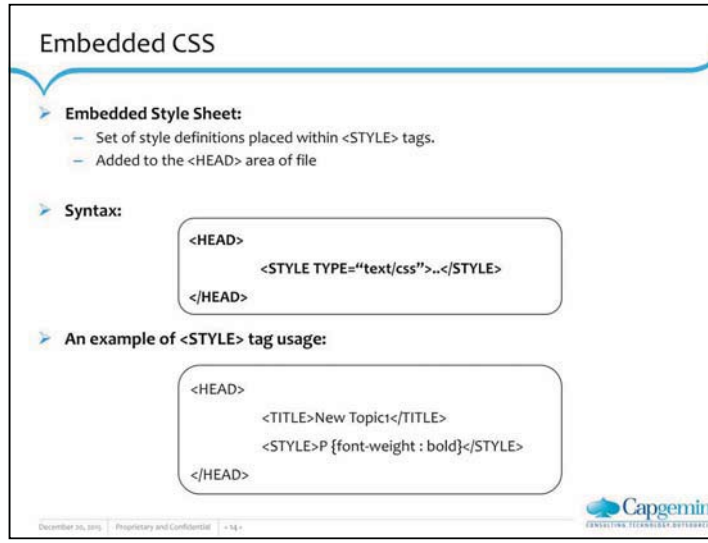


The above given HTML document content is formatted using CSS.



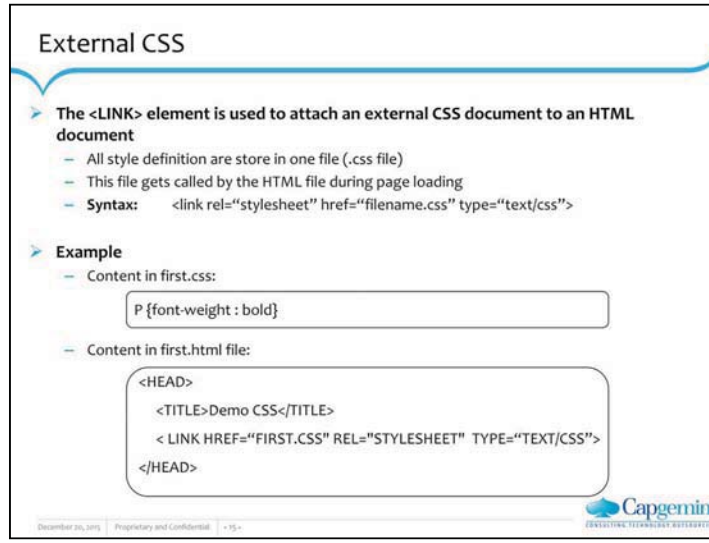
Inline Style Sheet:
Definitions appear next to other tag attributes. You need to remember to place the style sheet description within quotes, like the following:

```
<!DOCTYPE HTML>
<html>
<head><title>Inline Style Sheet</title></head>
<body style="background: white; color:green">
<h2 style="background: gold; font-family: Arial, Impact, Sans Serif;
color:red">
This is Level 2 Heading, with style</h2>
<h1 style="background: orange; font-family: Arial, Impact, Sans serif;
color: blue;font-size:30pt; text-align: center">
This is Level 1 Heading, with style</h1>
<h3 style="background: gold; font-family: Arial, Impact, Sans Serif;
color:red">
This is Level 3 Heading, with style</h3>
<h4>This is Level 4 Heading, without style</h4>
<h1>This is again Level 1 heading with default styles</h1>
</body>
</html>
```



Embedded Style Sheet:
An embedded style sheet is a set of style definitions placed within <STYLE> tags and located in the HEAD section of the HTML document. It sets the style attributes for the entire page where it is located.
Following style sheet description applies to the <H1> tag. It sets the font face to be either Arial, Impact, or Sans Serif, depending on which one it finds first on the user's system. Text color is also defined as blue.
H1 {font-family: Arial, Impact, Sans Serif; color: blue}
You can also group tags together by separating them with commas:
H1, H2, H3 {font-family: Arial, Impact, Sans Serif; color: blue}<html>

```
<!DOCTYPE HTML>
<html>
<head>
<style>    body {background: black; color:green}
h1 {background: orange; font-family: Arial; color:blue}
 h2, h3 {background: gold; font-family: Arial, Impact, Sans Serif; color:red}
</style> </head>
<body>
<h2>This is Level 2 Heading, with style</h2>
<h1>This is Level 1 Heading, with style</h1>
<h3>This is Level 3 Heading, with style</h3>
<h4>This is Level 4 Heading, without style</h4>
</body>
</html>
```

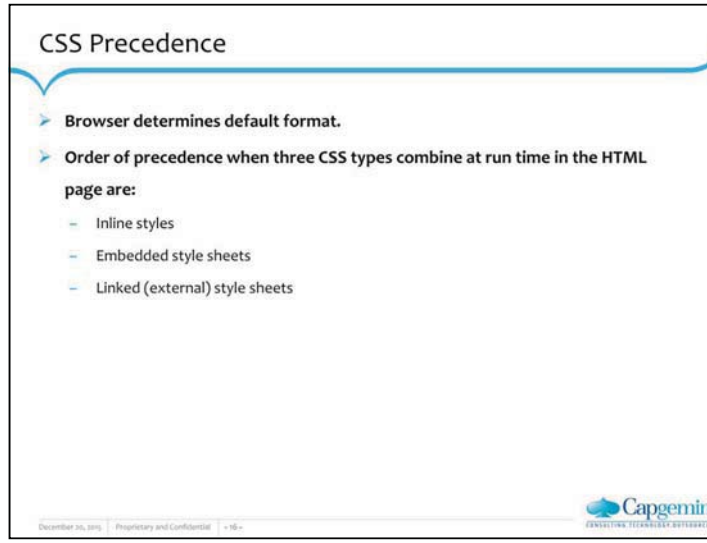


External CSS

External CSS is same as embedded style sheet. The only difference is that the separate css file contains all styles, and gets called by the HTML file.

Example:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Linked Style Sheet</title>
<link rel=stylesheet href="linked_ex2.css" type="text/css">
</head>
<body>
<h2>This is Level 2 Heading, with style</h2>
<h1>This is Level 1 Heading, with style</h1>
<h3>This is Level 3 Heading, with style</h3>
<h4>This is Level 4 Heading, without style</h4>
</body></html>
```



Style Sheet Precedence
There are several rules that apply to the order of precedence of style sheets. All tags have a default format determined by the browser. This is what you see if no style sheet attributes are set. This also represents the lowest priority.

Another level of priority is established by how close the style definition is to the tag. For this order, linked style sheets are lower than embedded style sheets, which are lower than inline style sheets. If you accidentally include the same property in a linked style sheet as in inline style sheet, then the priority goes to the definition closest to the tag, which would be inline style.

Style sheets for more specific tags have priority over general tags. For example, if a Web page marks the <BODY> tag with a certain style sheet definition and an <H3> tag with same property and a different value, then the <H3> tag has the priority, even though it is also part of the body.

Demo : CSS Syntax and CSS Types

- **Lesson01**
  - demo1.html
  - Embeddedstylesheet.htm
  - Linkedstylesheet.htm
  - Inlinestylesheet.htm

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

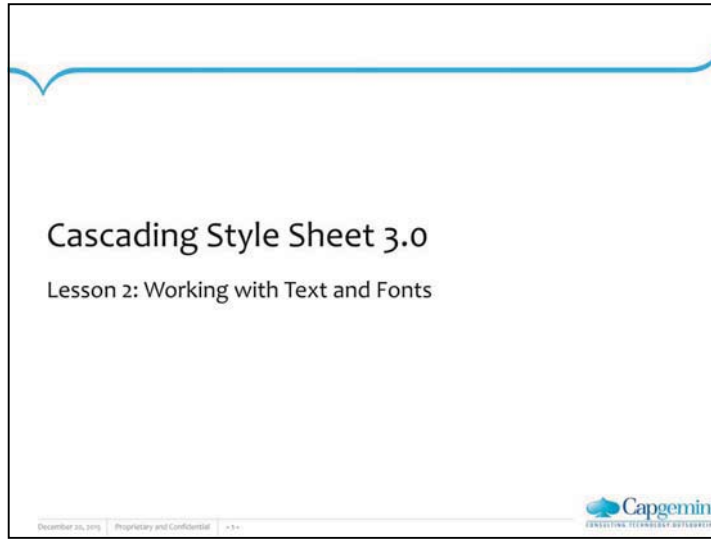
Lesson Summary

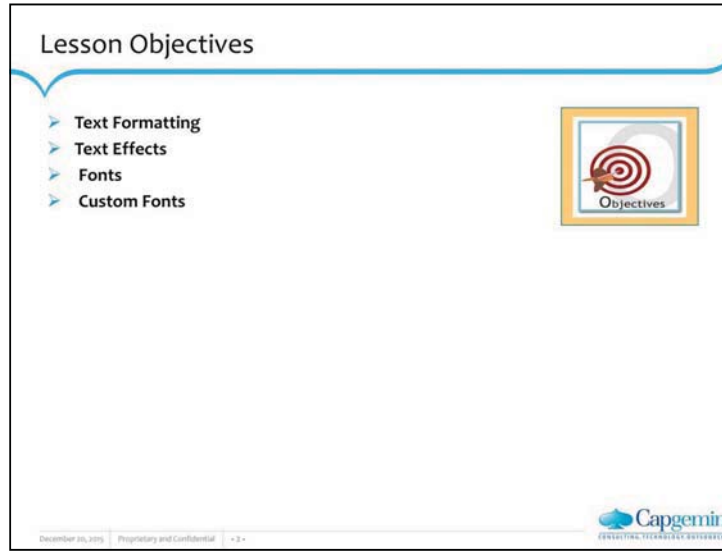
- **In this lesson, you have learnt about:**
  - What is CSS
  - CSS history
  - What CSS can do
  - CSS Syntax
  - Types of CSS
  - Cascading

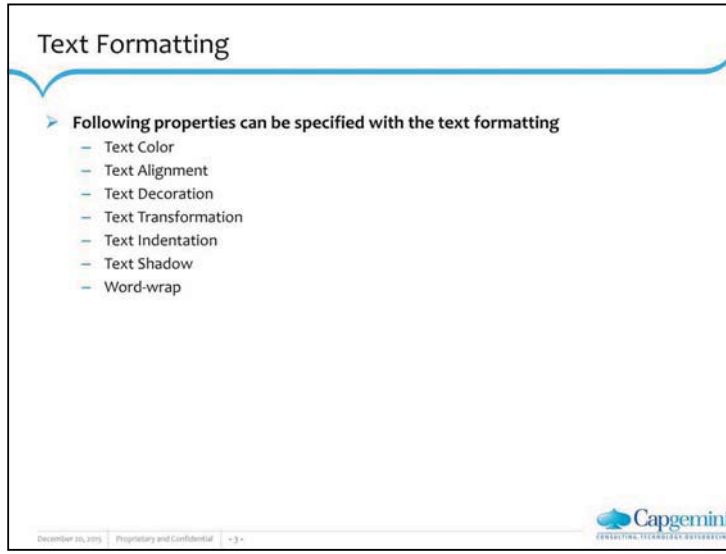
Review Questions

- **Question 1: Which of the following are CSS Types.**
  - Inline
  - Embedded
  - External
  - All the above
- **Question 2: CSS rule has _____ and _____**
  - Selector
  - Declaration
  - Element
  - All the Above

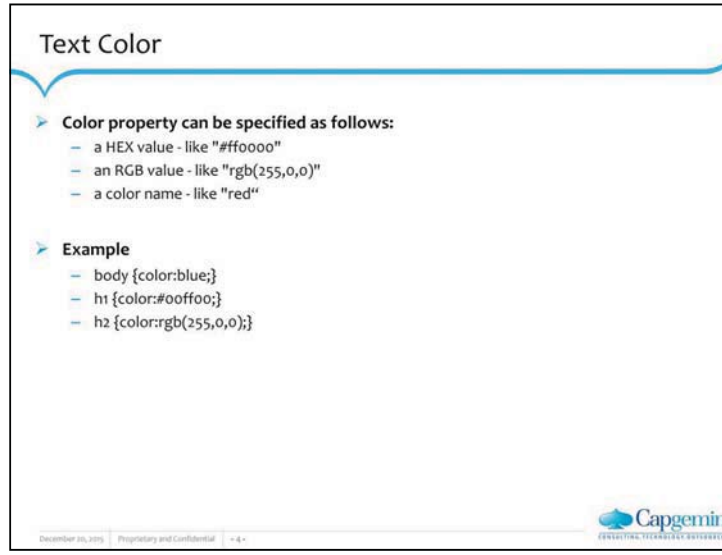
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

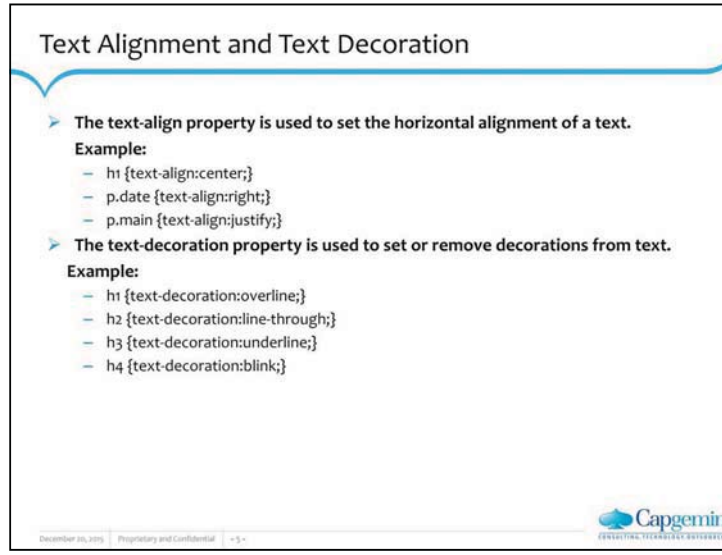

Cascading Style Sheet 3.0
Lesson 2: Working with Text and Fonts
Capgemini

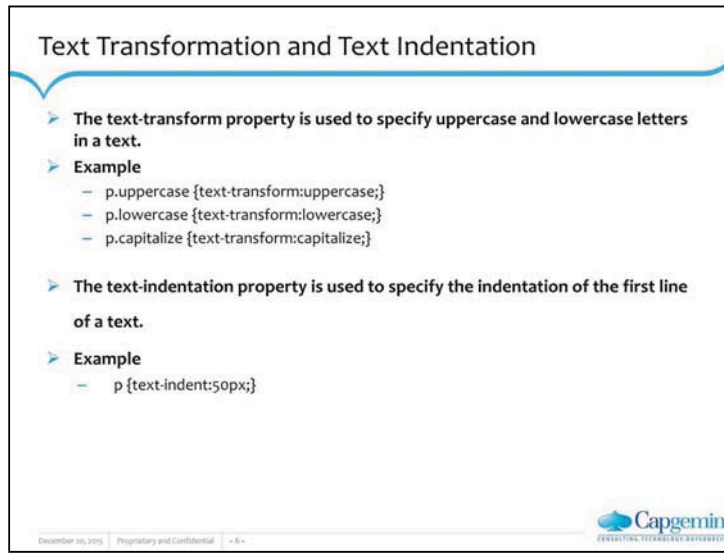

Lesson Objectives
Text Formatting
Text Effects
Fonts
Custom Fonts
Objectives
Capgemini

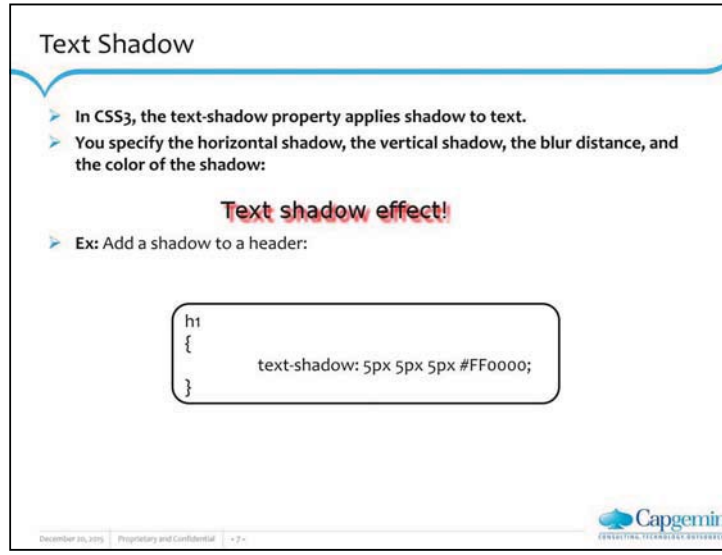


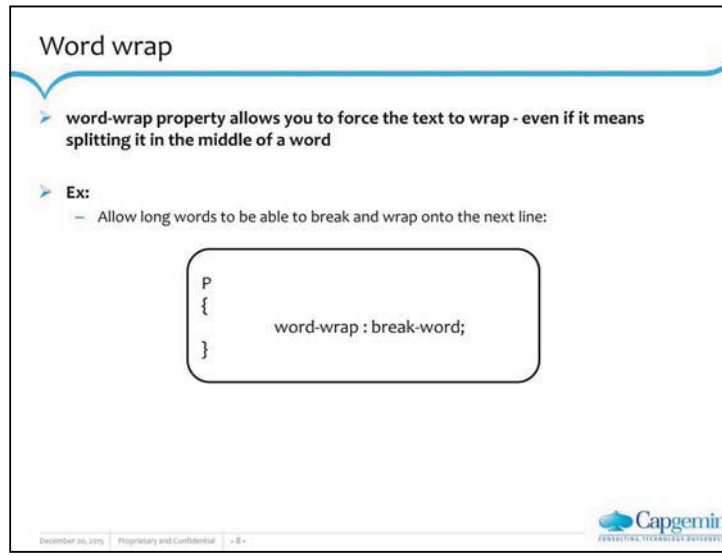
Text Color :The color property is used to set the color of the text.
Text Alignment: The text-align property is used to set the horizontal alignment of a text. Text can be centered, or aligned to the left or right, or justified.
Text Decoration: The text-decoration property is used to set or remove decorations from text.
The text-decoration property is mostly used to remove underlines from links for design purposes:
Text Transformation: The text-transform property is used to specify uppercase and lowercase letters in a text.
It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.
Text Indentation: The text-indentation property is used to specify the indentation of the first line of a text.
Text Shadow: In CSS3, the text-shadow property applies shadow to text
Word Wrapping: In CSS3, the word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word


Text Color
Color property can be specified as follows:
a HEX value - like "#ff0000"
an RGB value - like "rgb(255,0,0)"
a color name - like "red"
Example
body {color:blue;}
h1 {color:#00ff00;}
h2 {color:rgb(255,0,0);}
Capgemini

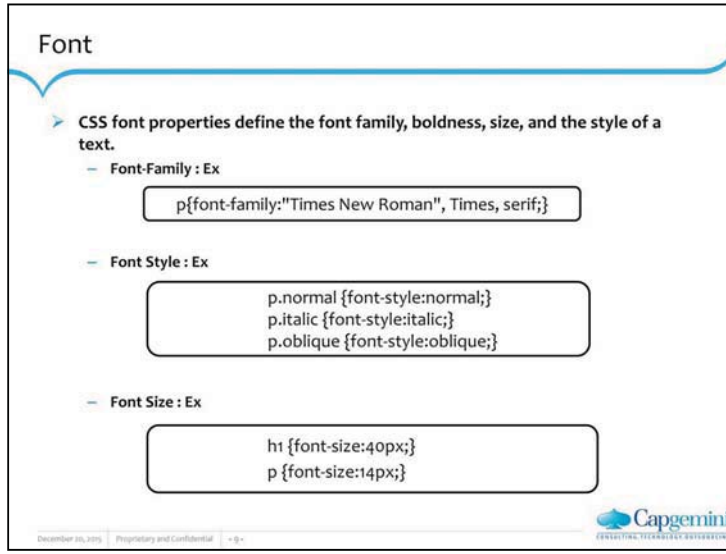

Text Alignment and Text Decoration
The text-align property is used to set the horizontal alignment of a text.
Example:
h1 {text-align:center;}
p.date {text-align:right;}
p.main {text-align:justify;}
The text-decoration property is used to set or remove decorations from text.
Example:
h1 {text-decoration:overline;}
h2 {text-decoration:line-through;}
h3 {text-decoration:underline;}
h4 {text-decoration:blink;}
Capgemini


Text Transformation and Text Indentation
The text-transform property is used to specify uppercase and lowercase letters in a text.
Example
p.uppercase {text-transform:uppercase;}
p.lowercase {text-transform:lowercase;}
p.capitalize {text-transform:capitalize;}
The text-indentation property is used to specify the indentation of the first line of a text.
Example
p {text-indent:50px;}
Capgemini


Text Shadow
In CSS3, the text-shadow property applies shadow to text.
You specify the horizontal shadow, the vertical shadow, the blur distance, and the color of the shadow:
Text shadow effect!
Ex: Add a shadow to a header:
h1
{
text-shadow: 5px 5px 5px #FF0000;
}
Capgemini



New Text Properties:
hanging-punctuation:Specifies whether a punctuation character may be placed outside the line box
punctuation-trim:Specifies whether a punctuation character should be trimmed
text-align-last:Describes how the last line of a block or a line right before a forced line break is aligned when text-align is "justify"
text-emphasis:Applies emphasis marks, and the foreground color of the emphasis marks, to the element's text
text-justify:Specifies the justification method used when text-align is "justify"
text-outline:Specifies a text outline
text-overflow:Specifies what should happen when text overflows the containing element
text-shadow:Adds shadow to text
text-wrap:Specifies line breaking rules for text
word-break:Specifies line breaking rules for non-CJK scripts
word-wrap:Allows long, unbreakable words to be broken and wrap to the next line



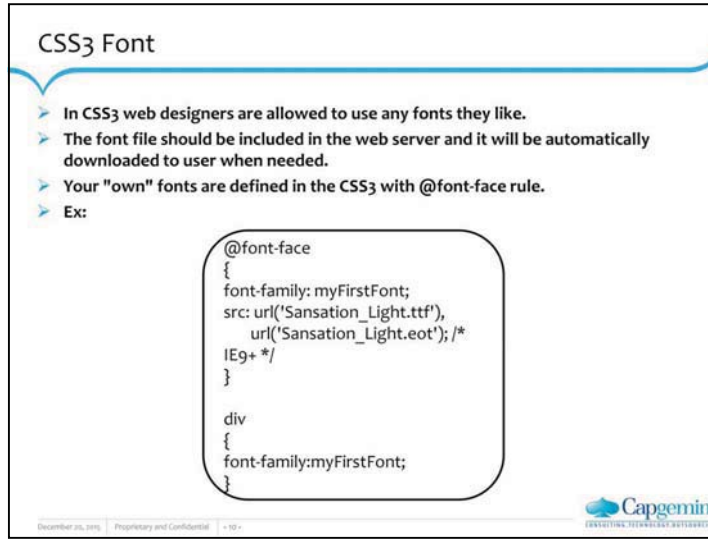
Font Family: The font family of a text is set with the font-family property.
If the name of a font family is more than one word, it must be in quotation marks, like font-family: "Times New Roman".
More than one font family is specified in a comma-separated list

Font Style: The font-style property is mostly used to specify italic text.
This property has three values:
normal - The text is shown normally
italic - The text is shown in italics
oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

Font Size : Setting the text size with pixels gives you full control over the text size:
It can be set either using ***px*** attribute or ***em*** attribute as follows:

```
h1 {font-size:40px;}
h1 {font-size:2.5em;}
```

Note:The em size unit is recommended by the W3C.
1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.
The size can be calculated from pixels to em using this formula: *pixels*/16=*em*



Using Bold text with @font-face:

```
@font-face
{
font-family: myFirstFont;
src: url('Sansation_Bold.ttf'),
    url('Sansation_Bold.eot'); /* IE9+ */
font-weight:bold;
}
```

The file "Sansation_Bold.ttf" is another font file, that contains the bold characters for the Sansation font.
Browsers will use this whenever a piece of text with the font-family "myFirstFont" should render as bold.
This way you can have many @font-face rules for the same font.

Demo : Text and Font

- Lesson02
  - demoFontText.html
  - word_wrap.html

Lesson Summary

- **In this lesson, you have learnt about**
  - Text Formatting
  - Text Effects
  - Fonts
  - Custom Fonts

Review Questions

- **Question 1:Given :**

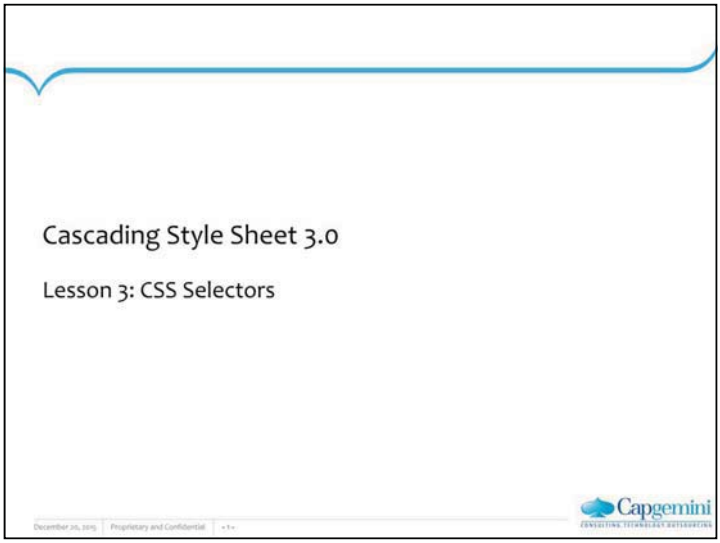
```
h1
 {
 text-shadow: A ,B ,C,D;
 }
```

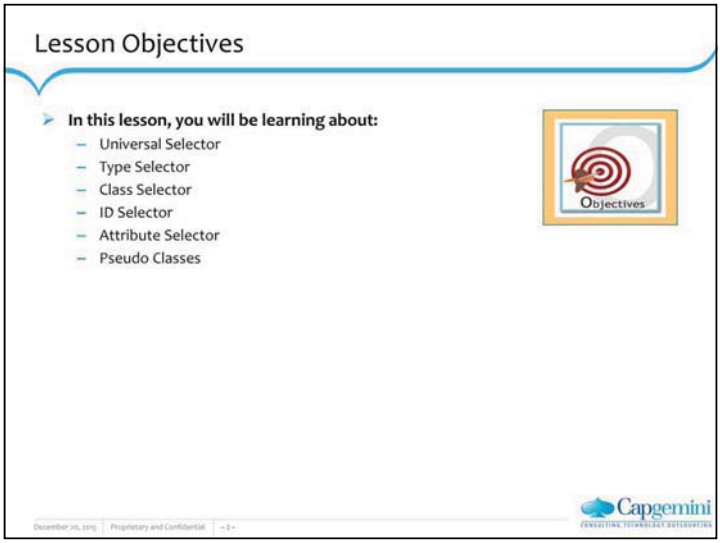
What property does C represents?

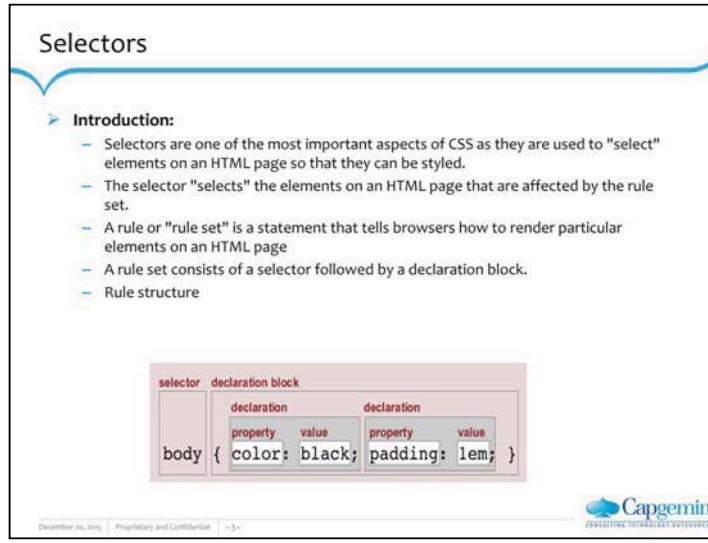
  - Option 1: Colour
  - Option 2: Vertical Shadow
  - Option 3: Blurr
  - Option 4: Horizontal Shadow

- **Question 2: Custom Fonts can be included with CSS 3**
  - Option 1: TRUE
  - Option 2: FALSE

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING


Cascading Style Sheet 3.0
Lesson 3: CSS Selectors
Capgemini


Lesson Objectives
In this lesson, you will be learning about:
Universal Selector
Type Selector
Class Selector
ID Selector
Attribute Selector
Pseudo Classes
Objectives
Capgemini



Text Color :The color property is used to set the color of the text.

Text Alignment:The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

Text Decoration:The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

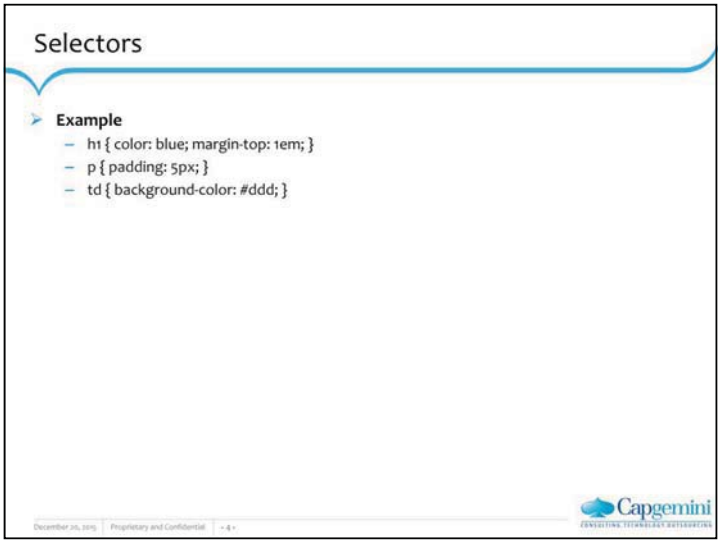
Text Transformation:The text-transform property is used to specify uppercase and lowercase letters in a text.

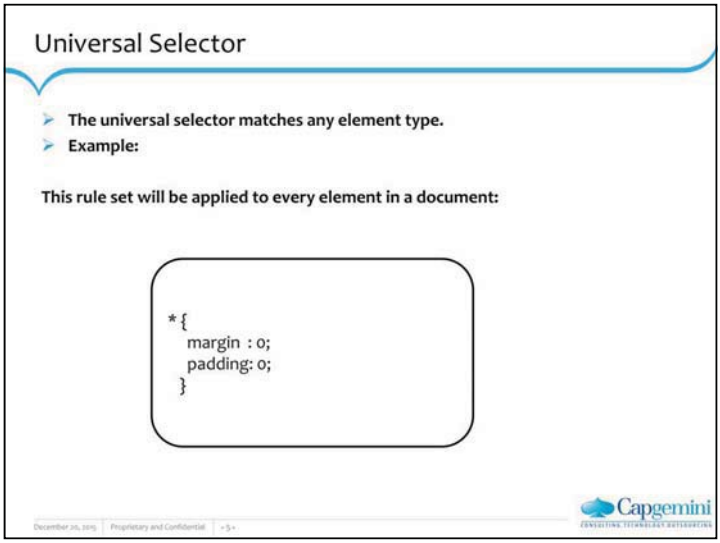
It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Text Indentation:The text-indentation property is used to specify the indentation of the first line of a text.

Text Shadow: In CSS3, the text-shadow property applies shadow to text

Word Wrapping:In CSS3, the word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word


Selectors
Example
h1 { color: blue; margin-top: 1em; }
p { padding: 5px; }
td { background-color: #ddd; }
Capgemini

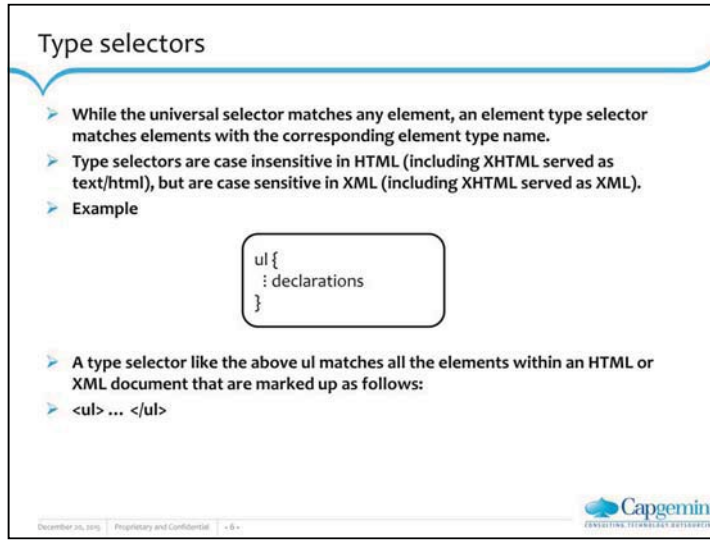


It's important not to confuse the universal selector with a wildcard character—the universal selector doesn't match "zero or more elements." Consider the following HTML fragment:

```
<body>
 <div>
  <h1>The <em>Universal</em> Selector</h1>
  <p>We must <em>emphasize</em> the following:</p>
  <ul>
   <li>It's <em>not</em> a wildcard.</li>
   <li>It matches elements regardless of <em>type</em>.</li>
  </ul>
  This is an <em>immediate</em> child of the division.
 </div>
</body>
```

The selector div * em will match the following em elements:

"Universal" in the h1 element (* matches the <h1>)
"emphasize" in the p element (* matches the <p>)
"not" in the first li element (* matches the <ul> or the <li>)
"type" in the second li element (* matches the <ul> or the <li>)
However, it won't match the immediate element, since that's an immediate child of the div element—there's nothing between and for the * to match.

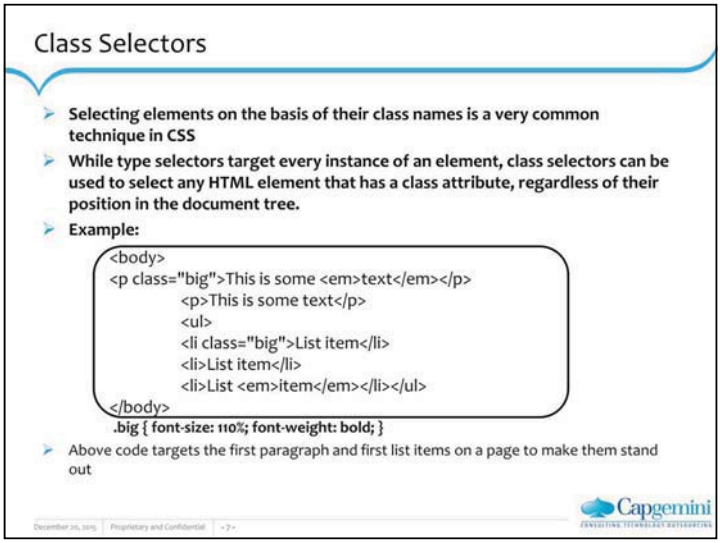


The most common and easy to understand selectors are type selectors. Type selectors will select any HTML element on a page that matches the selector, regardless of their position in the document tree. For example:

em {color: blue; }

This rule will select any element on the page and color it blue. As you can see from the document tree diagram below, all elements will be colored blue, regardless of their position in the document tree

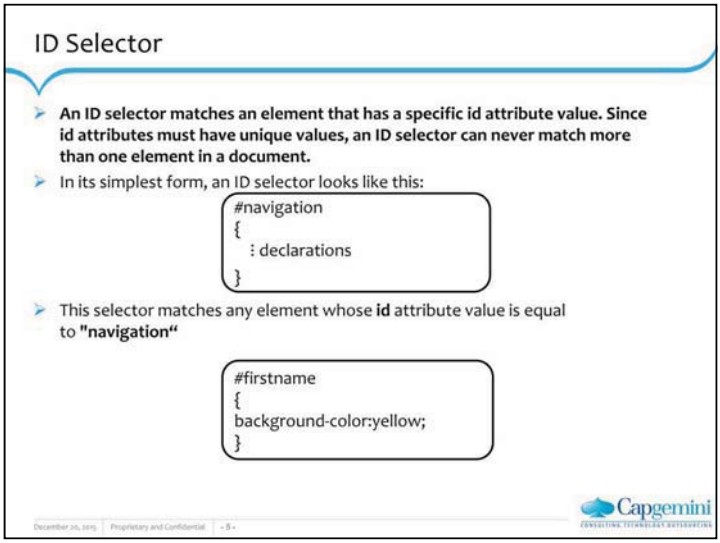
There are a huge range of elements that you can select using type selectors, which means you can change the appearance of any or every element on your page using only type selectors.



Combining class and type selectors:

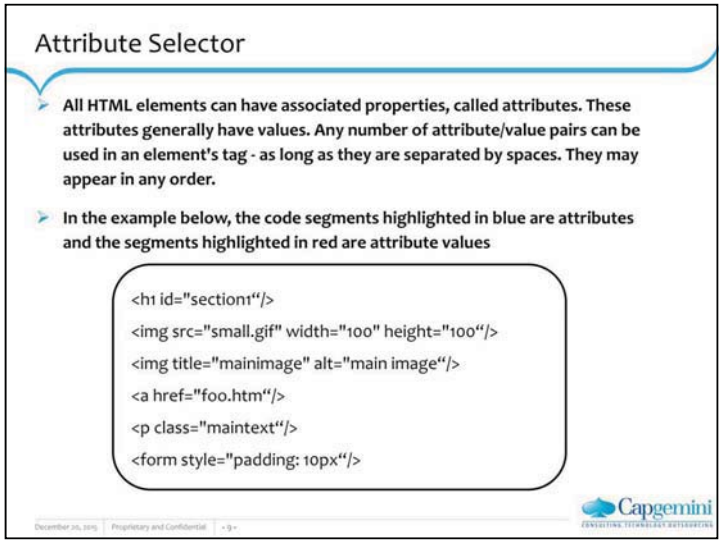
If you want to be more specific, you can use class and type selectors together. Any type selectors can be used.

```
div.big { color: blue; }
td.big { color: yellow; }
label.big { color: green; }
form.big { color: red; }
```

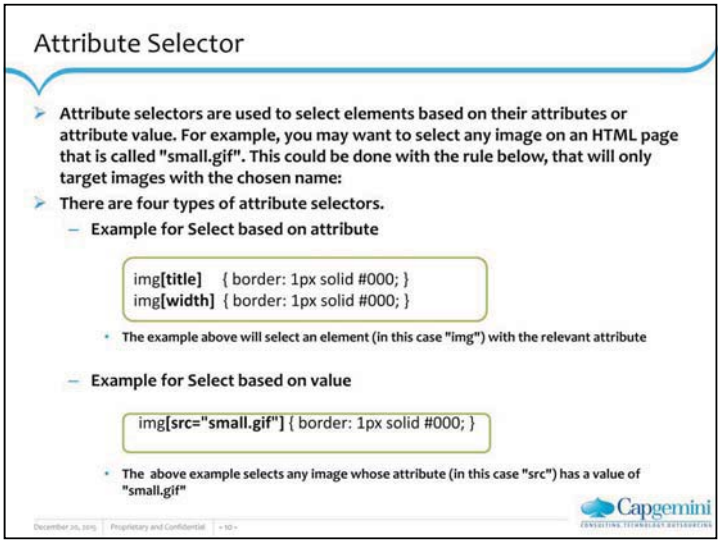


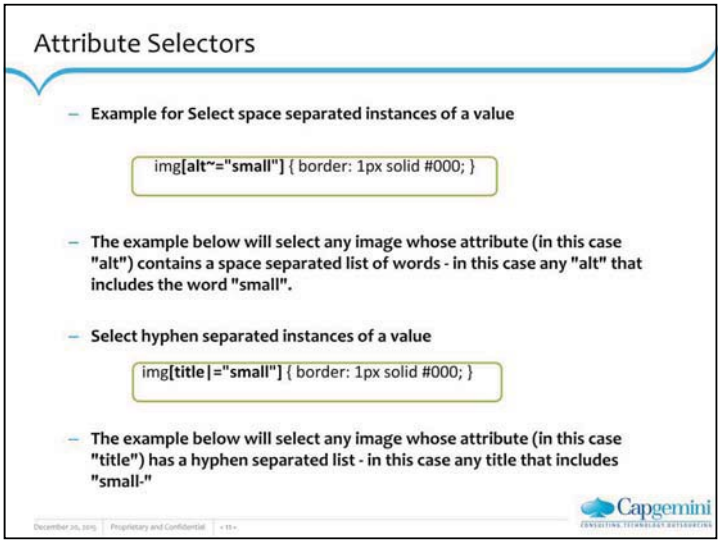
Code:

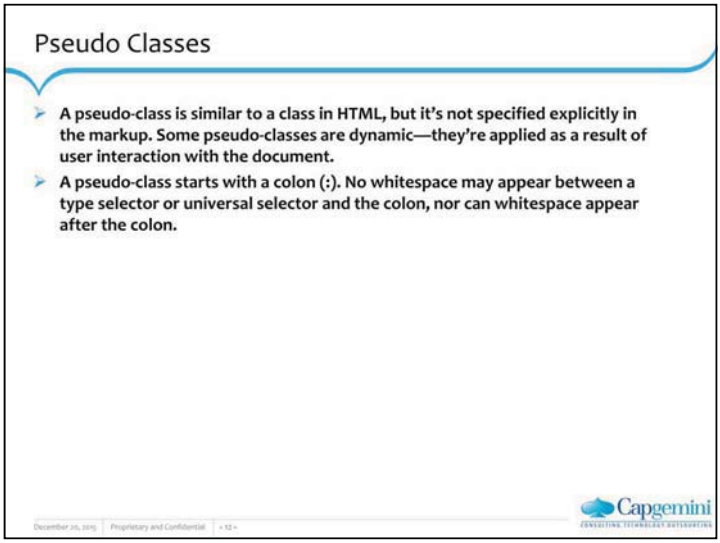
```
<!DOCTYPE html>
<html>
<head>
<style>
#firstname
{
                background-color:yellow;
}
</style>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<div class="intro">
<p id="firstname">My name is iGATE.</p>
<p id="hometown">I live in Bangalore.</p>
</div>

<p>My best friend was Patni.</p>

</body>
</html>
```


Attribute Selector
All HTML elements can have associated properties, called attributes. These attributes generally have values. Any number of attribute/value pairs can be used in an element's tag - as long as they are separated by spaces. They may appear in any order.
In the example below, the code segments highlighted in blue are attributes and the segments highlighted in red are attribute values
<h1 id="section1"/>

<a href="foo.htm"/>
<p class="maintext"/>
<form style="padding: 10px"/>
Capgemini


Attribute Selector
Attribute selectors are used to select elements based on their attributes or attribute value. For example, you may want to select any image on an HTML page that is called "small.gif". This could be done with the rule below, that will only target images with the chosen name:
There are four types of attribute selectors.
Example for Select based on attribute
img[title] { border: 1px solid #000; }
img[width] { border: 1px solid #000; }
The example above will select an element (in this case "img") with the relevant attribute
Example for Select based on value
img[src="small.gif"] { border: 1px solid #000; }
The above example selects any image whose attribute (in this case "src") has a value of "small.gif"
Capgemini


Attribute Selectors
– Example for Select space separated instances of a value
img[alt~="small"] { border: 1px solid #000; }
– The example below will select any image whose attribute (in this case "alt") contains a space separated list of words - in this case any "alt" that includes the word "small".
– Select hyphen separated instances of a value
img[title|="small"] { border: 1px solid #000; }
– The example below will select any image whose attribute (in this case "title") has a hyphen separated list - in this case any title that includes "small-"
Capgemini



CSS1 introduced the :link, :visited, and :active pseudo-classes, but only for the HTML a element. These pseudo-classes represented the state of links—unvisited, visited, or currently being selected—in a web page document. In CSS1, all three pseudo-classes were mutually exclusive.

CSS2 expanded the range of pseudo-classes and ensured that they could be applied to any element. :link and :visited now apply to any element defined as a link in the document language. While they remain mutually exclusive, the :active pseudo-class now joins :hover and :focus in the group of dynamic pseudo-classes. The :hover pseudo-class matches elements that are being designated by a pointing device (for example, elements that the user's hovering the cursor over); :active matches any element that's being activated by the user; and :focus matches any element that is currently in focus (that is, accepting input).

CSS2 also introduced the :lang pseudo-class to allow an element to be matched on the basis of its language, and the :first-childpseudo-class to match an element that's the first child element of its parent.

CSS3 promises an even greater range of powerful pseudo-classes.

Remember that pseudo-classes, like ID selectors and attribute selectors, act like modifiers on type selectors and the universal selector: they specify additional constraints for the selector pattern, but they don't specify other elements. For instance, the selector li:first-child matches a list item that's the first child of its parent; it doesn't match the first child of a list item.

Pseudo Classes

Pseudo class	Description
:link	matches link elements that are unvisited
:visited	matches link elements that have been visited
:active	matches any element that's being activated by the user
:hover	matches elements that are being designated by a pointing device
:focus	matches any element that's currently in focus
:first-child	matches any element that's the first child element of its parent
:lang(C)	allows elements to be matched on the basis of their languages

CSS 3 - Pseudo Classes

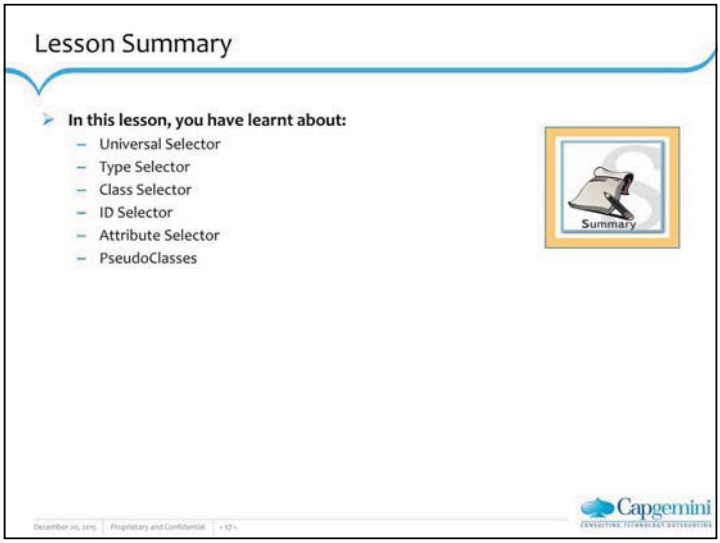
Pseudo class	Description
:nth-child(N)	matches elements on the basis of their positions within a parent element's list of child elements
:nth-last-child(N)	matches elements on the basis of their positions within a parent element's list of child elements
:nth-of-type(N)	matches elements on the basis of their positions within a parent element's list of child elements of the same type
:nth-last-of-type(N)	matches elements on the basis of their positions within a parent element's list of child elements of the same type
:last-child	matches an element that's the last child element of its parent element
:first-of-type	matches the first child element of the specified element type
:last-of-type	matches the last child element of the specified element type

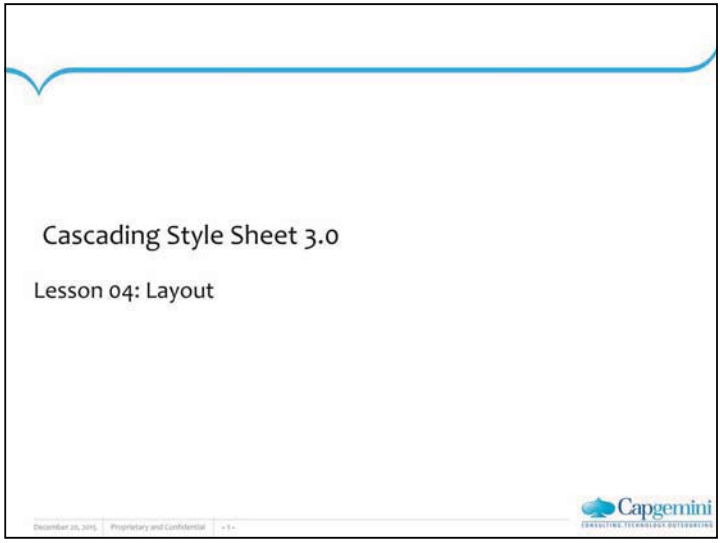
CSS 3 - Pseudo Classes

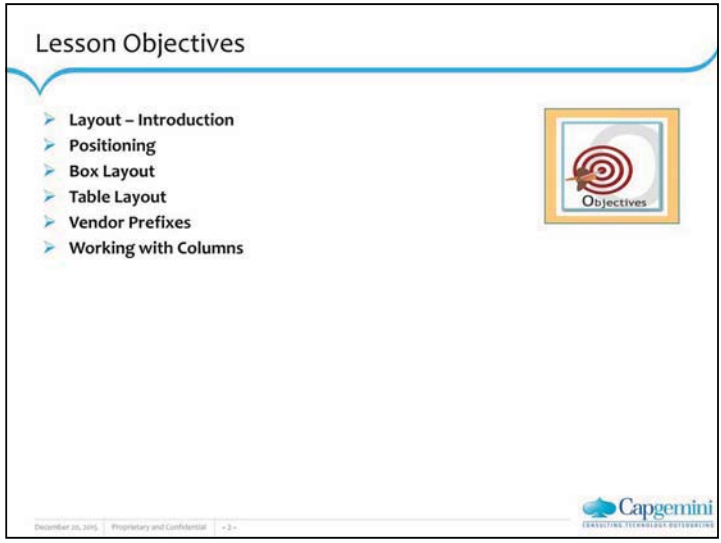
Pseudo class	Description
:only-child	matches an element if it's the only child element of its parent
:only-of-type	matches an element that's the only child element of its type
:root	matches the element that's the root element of the document
:empty	matches elements that have no children
:target	matches an element that's the target of a fragment identifier in the document's URI
:enabled/:disabled	matches user interface elements that are enabled/disabled respectively
:checked Pseudo-class	matches elements like checkboxes or radio buttons that are checked
:not(S)	matches elements that aren't matched by the specified selector

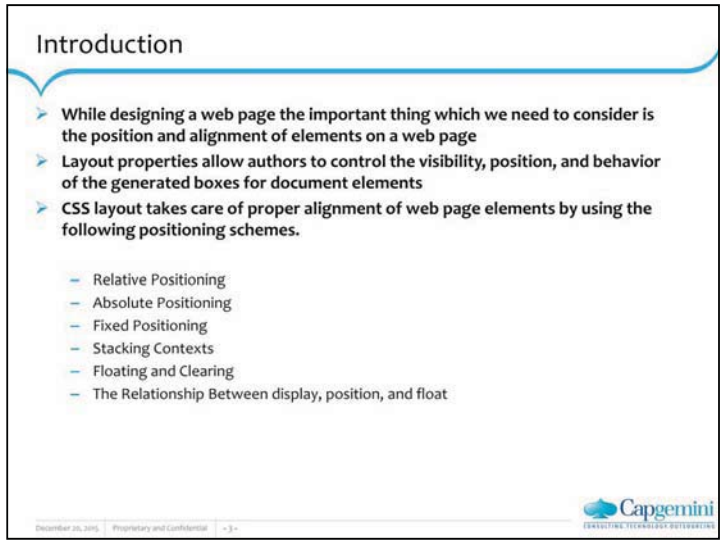
Demo : Selector

- demoType.html
- demoId.html
- demoClass.html
- demoAttributeSelector.html
- demoPseudoClasses.html


Lesson Summary
In this lesson, you have learnt about:
Universal Selector
Type Selector
Class Selector
ID Selector
Attribute Selector
PseudoClasses
Summary
Capgemini

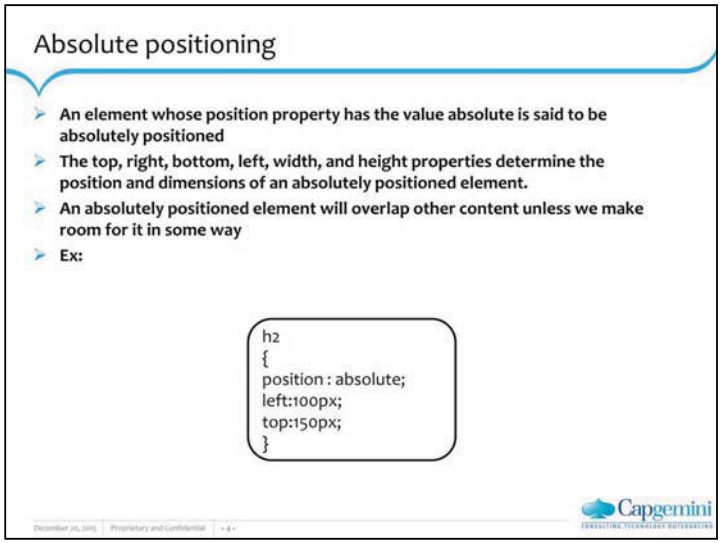

Cascading Style Sheet 3.0
Lesson 04: Layout
Capgemini


Lesson Objectives
Layout – Introduction
Positioning
Box Layout
Table Layout
Vendor Prefixes
Working with Columns
Objectives
Capgemini

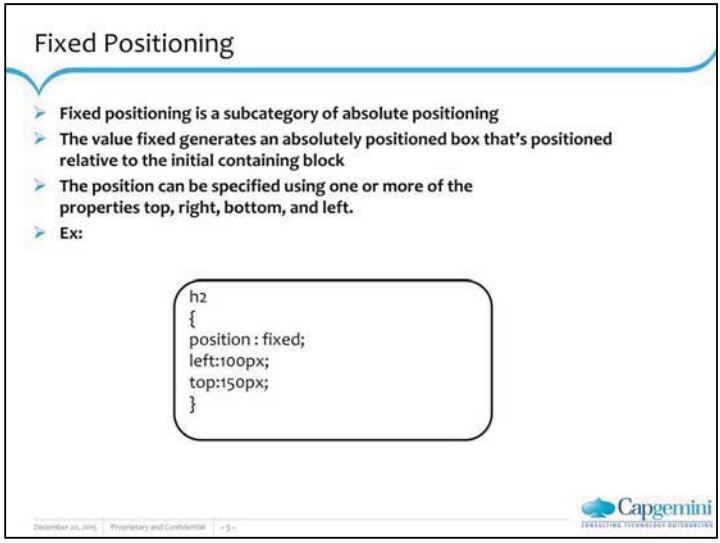


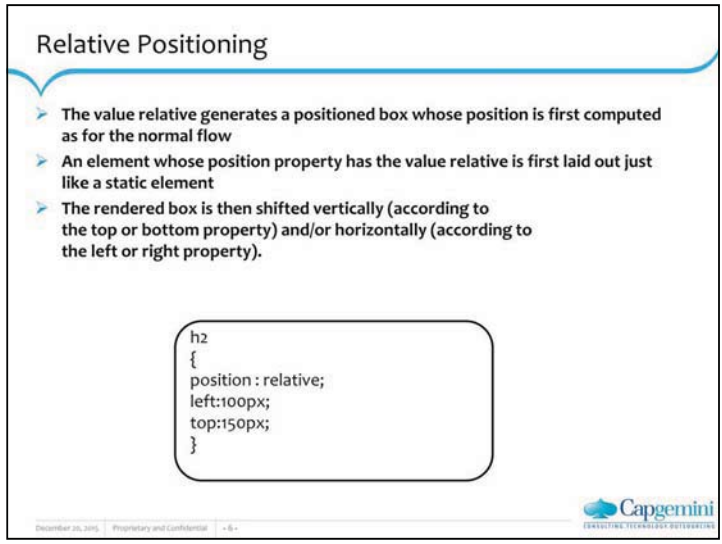
The term "CSS positioning" refers to using CSS to position elements on your HTML page. CSS allows you to position any element precisely where you want it. You can specify whether you want the element positioned*relative* to its natural position in the page or *absolute* based on its parent element.

The position property, together with the float property, controls the way in which the position of the element's generated box is computed.



Both the position and the dimensions can be expressed using all four of the positional properties (top, right, bottom, left). Alternatively, you can specify the position of one corner of the box using top or bottom in combination with left or right, and you can specify the dimensions using width and (optionally) height.

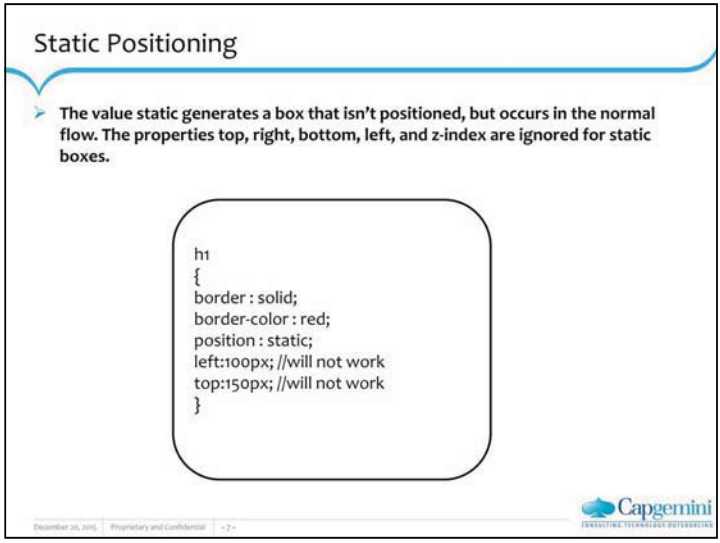

Fixed Positioning
Fixed positioning is a subcategory of absolute positioning
The value fixed generates an absolutely positioned box that's positioned relative to the initial containing block
The position can be specified using one or more of the properties top, right, bottom, and left.
Ex:
h2
{
position : fixed;
left:100px;
top:150px;
}
Capgemini

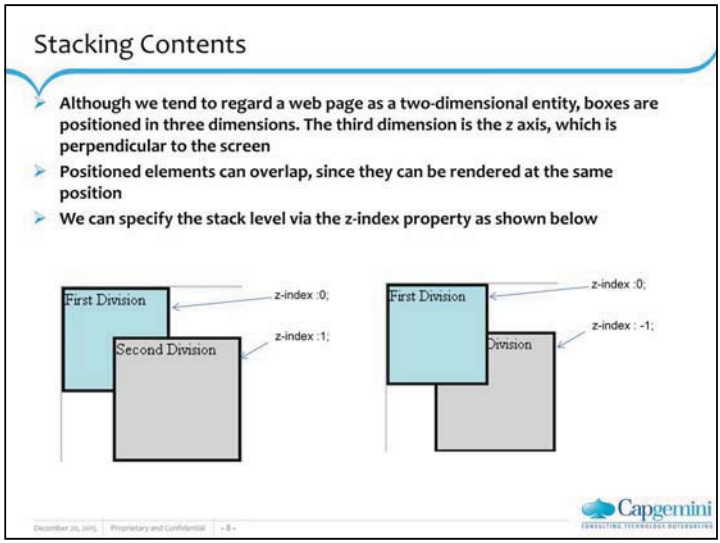


As far as the flow is concerned, the element is still in its original position. If the relative shift is significant, it will leave a "hole" in the flow, in which case the rendered box may overlap other content.

The properties top, right, bottom, and left can be used to specify by how much the rendered box will be shifted. A positive value means the box will be shifted away from that position, towards the opposite side. For instance, a left value of 20px shifts the box 20 pixels to the right of its original position. Applying a negative value to the opposite side will achieve the same effect: a right value of -20px will accomplish the same result as a leftvalue of 20px. The initial value for these properties is auto, which makes the computed value 0 (zero)—that is, no shift occurs

Relative positioning is commonly used when we need to shift a box a few pixels or so, although it can also be useful, in combination with negative margins on floated elements, for some more complex designs


Static Positioning
The value static generates a box that isn't positioned, but occurs in the normal flow. The properties top, right, bottom, left, and z-index are ignored for static boxes.
h1
{
border : solid;
border-color : red;
position : static;
left:100px; //will not work
top:150px; //will not work
}
Capgemini



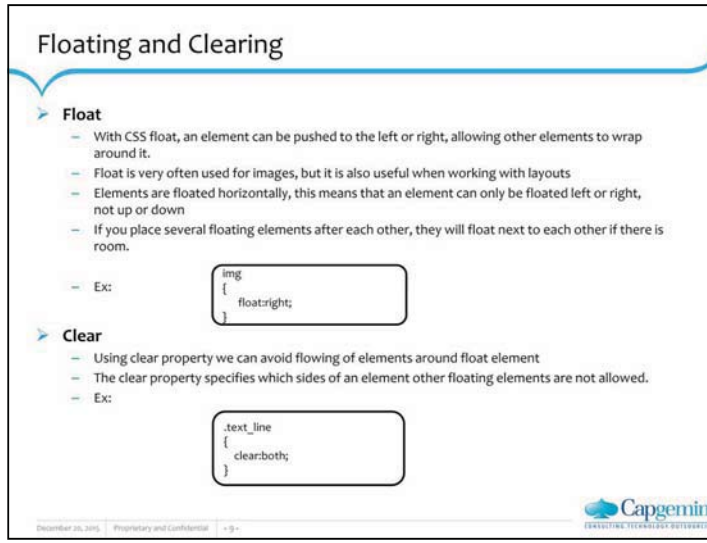
An integer value—which can be negative—sets the stack level of the box in the current stacking context, and also establishes a new stacking context. The box itself has stack level 0 (zero) in the new context.

The value auto gives the box the same stack level as its parent, and doesn't establish a new stacking context

```
<html>
<body>
   <div id="m1">First Division</div>
   <div id="m2">Second Division</div>
</body>
</html>
```

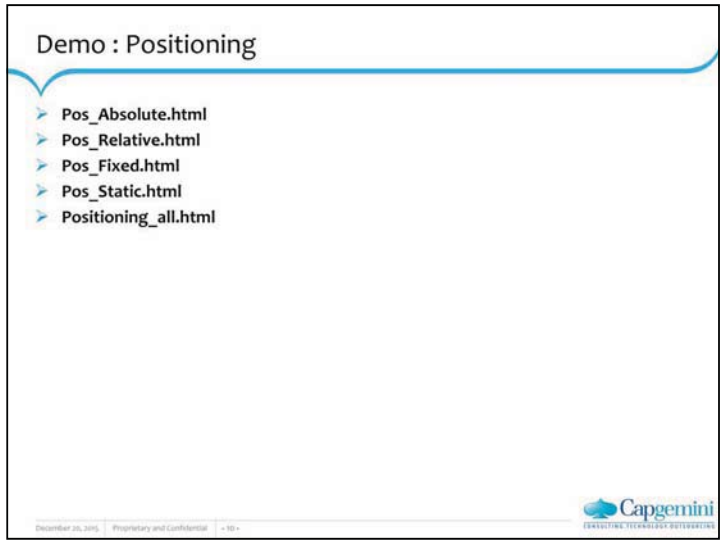
```
#m1
{
 background-olor:powderblue;
 colour:RED;
 width:100px;
 height:100px;
 z-index:0;
}
```

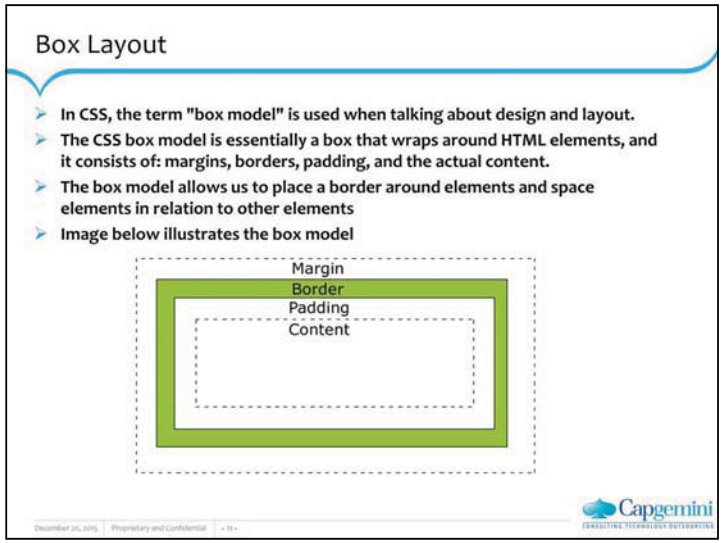
```
#m2
{
 background-color:lightgrey;
 colour:blue;
 top:50px;
 left:50px;
 width:120px;
 height:120px;
 position:absolute;
 z-index:1;
}
```



Float:A floated element is one whose float property has a value other than none. The element can be shifted to the left (using the valueleft) or to the right (using the value right); non-floated content will flow along the side opposite the specified float direction

Clear: To force an element to start below any floated elements, we can use the clear property with a value of left, right, or both. An element whose clear property is set to left will start below all left-floated boxes in the same block formatting context, while aclear value of right will clear all right-floated boxes. If clear is set to both, the element will start below any floated box in that context


Demo : Positioning
Pos_Absolute.html
Pos_Relative.html
Pos_Fixed.html
Pos_Static.html
Positioning_all.html
Capgemini



Your understanding of the box model concept, and how it relates to the way in which an element's final dimensions are determined, will be essential to your understanding of how an element is positioned on a web page. The box model applies to block-level elements.

Explanation of the different parts:

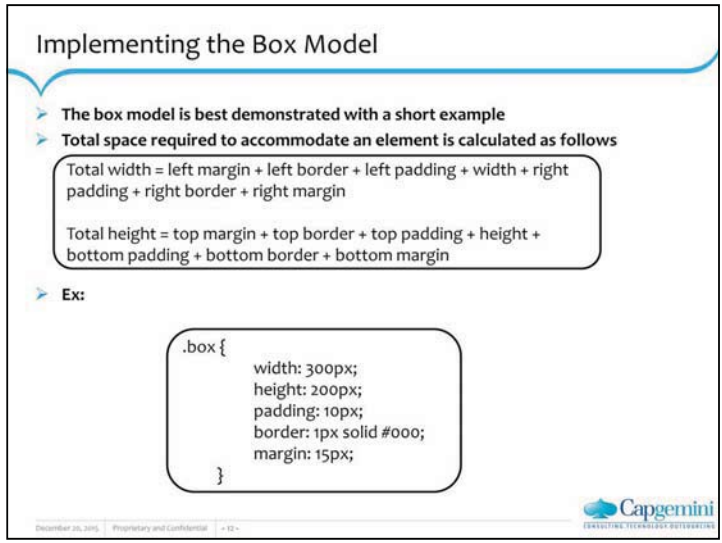
Margin - Clears an area around the border. The margin does not have a background color, it is completely transparent

Border - A border that goes around the padding and content. The border is affected by the background color of the box

Padding - Clears an area around the content. The padding is affected by the background color of the box

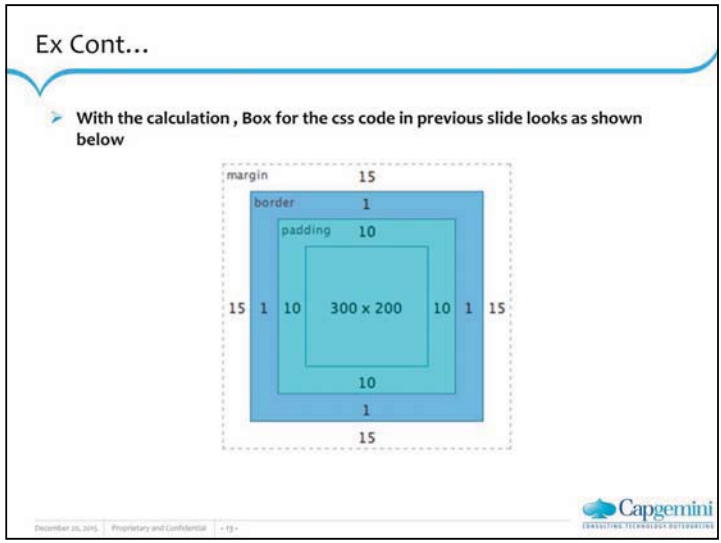
Content - The content of the box, where text and images appear

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.



The total size of the element above will be calculated as follows:

Total width = 15 + 1 + 10 + 300 + 10 + 1 + 15 = 352px
Total height = 15 + 1 + 10 + 200 + 10 + 1 + 15 = 252px



In the above picture , we can clearly see the content area in the center, the padding around the content area, the border area, and the margin area. The outer edge of the content area is called the content edge or inner edge; the outer edge of the padding area is called the padding edge; the outer edge of the border area is called the border edge; and the outer edge of the margin area is called—you guessed it—the margin edge or outer edge.

You can see from this short example that, for this element to fit on the page, we'll need a space that's at least 352px wide and252px high. If the space available is any smaller than this, the element will be misplaced, or will overflow its containing block. Note that Internet Explorer 6 and earlier versions will most likely stretch the containing block to accommodate this extra height, and could severely disrupt the layout. Other browsers will let the element overflow its boundaries, but will ignore the content.

Notes:

An important point to note is that an element that has its widthset to 100% (that is, 100% of the content width of its parent element) shouldn't have any margins, padding, or borders applied

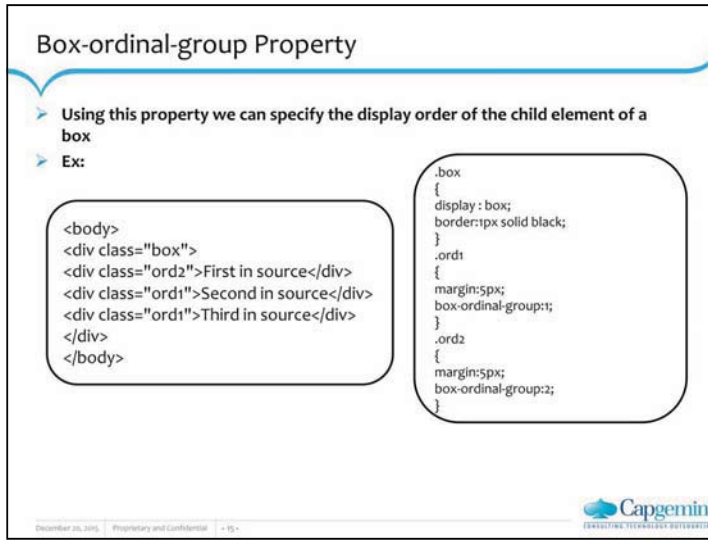
It can severely disrupt a page's layout, as content will either overflow or push elements wider than they should be.

The solution, in most cases, is to avoid adding a value for the property width (other than auto), and to apply the margins, padding, and borders only. The width property of a static element will default to auto, and even with padding, borders, and margins added, it will still assume the full available content width..

This approach may not be feasible in some instances, such as cases where the element is not a static element, and requires the definition of a specific width value (as in the case of a floated element that doesn't automatically expand to fill its parent). In these cases, you have two options.

If the available space is of a fixed width, you can simply add the value of each component together to ensure that it matches the available width. For example, if the available space is 500px wide, and you require an element to have 20px padding, simply set the width to 460px and the padding to 20px for that element (20 + 460 + 20 = 500). This solution assumes that the length values specified for the element's box properties use the same unit of measurement, since you won't be able to add together a mixture of units (200px + 10%, for example, makes no sense in this context).

When the available content space has an unknown width—as in the case of a fluid layout—this method can't be used, as percentages and pixels can't be added together. In this case, the solution is to declare a width of 100% for the element concerned, and to apply the padding, border, and margin values to a nested element instead. That nested element has no width declaration, and can display the required padding, borders, and margins without encroaching on the parent element.



The box-ordinal-group property specifies the display order of the child elements of a box.

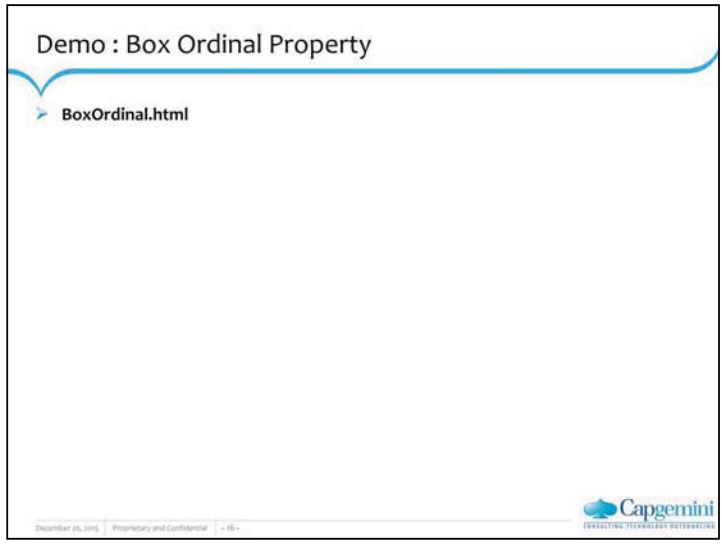
Elements with a lower value are displayed before those with a higher value.

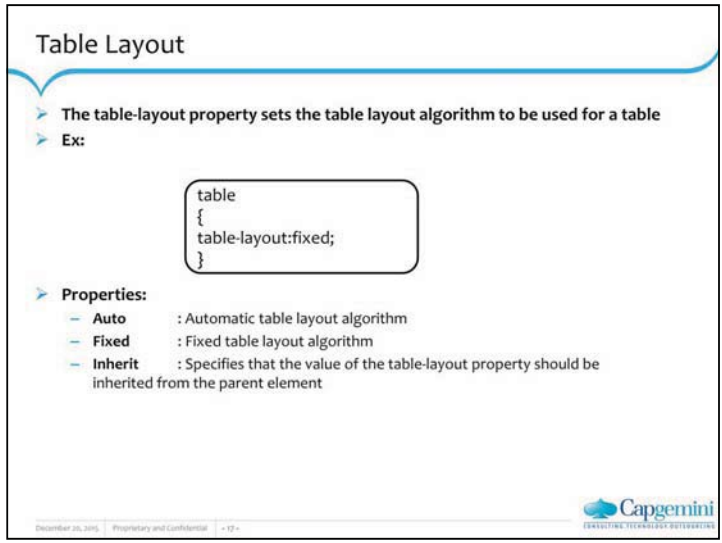
Note:

The display order of the elements with the same group value depend on their source order.

Use vendor prefixes for box-ordinal-group property.

It will not work on internet explorer

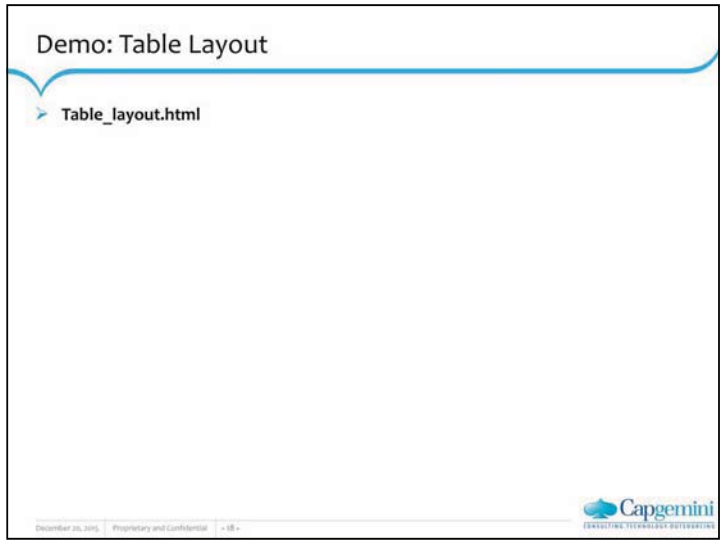

Demo : Box Ordinal Property
BoxOrdinal.html
Capgemini

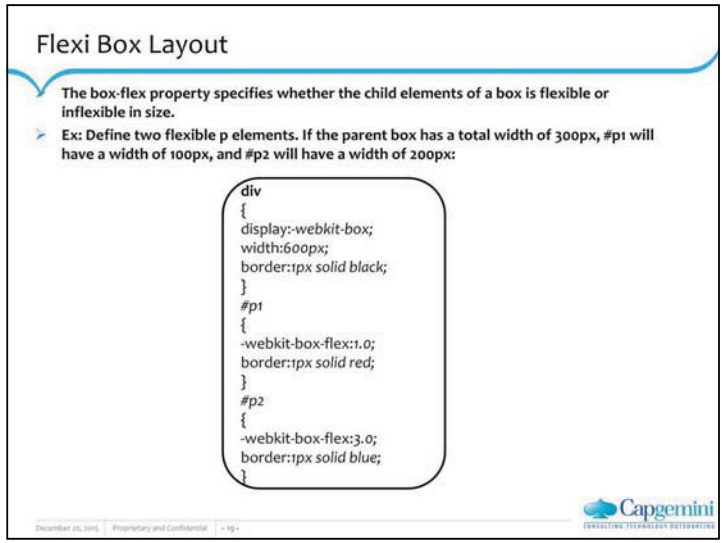


Auto: Automatic table layout algorithm (this is default):
The column width is set by the widest unbreakable content in the cells
Can be slow, since it needs to read through all the content in the table, before determining the final layout

Fixed: Fixed table layout algorithm:
The horizontal layout only depends on the table's width and the width of the columns, not the contents of the cells
Allows a browser to lay out the table faster than the automatic table layout
The browser can begin to display the table once the first row has been received

Inherit: Specifies that the value of the table-layout property should be inherited from the parent element


Demo: Table Layout
Table_layout.html
Capgemini



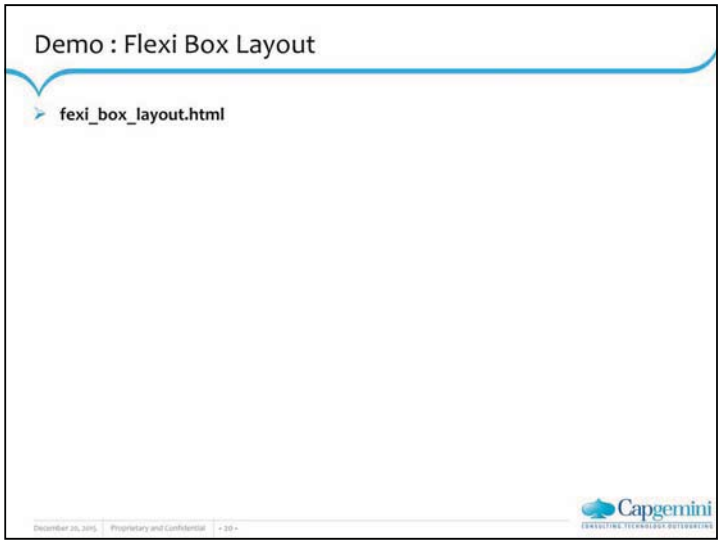
Note:

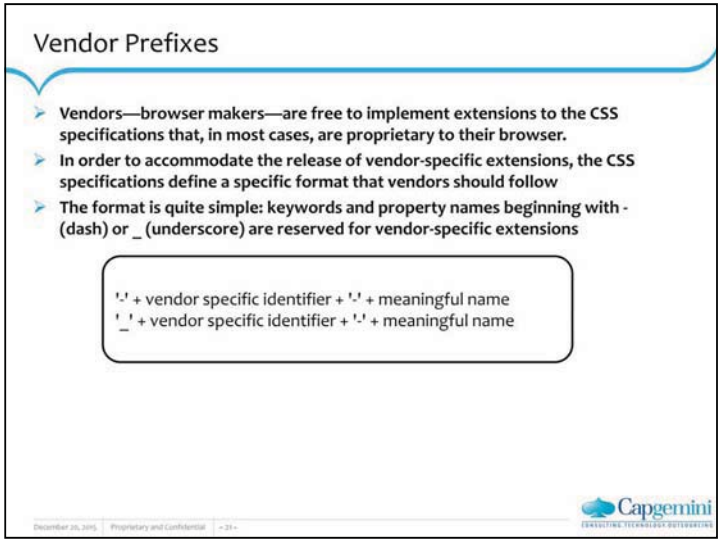
Elements that are flexible can shrink or grow as the box shrinks and grows. Whenever there is extra space in a box, flexible elements are expanded to fill that space.

The box-flex property is only supported by Opera.

Firefox supports an alternative, the -moz-box-flex property.

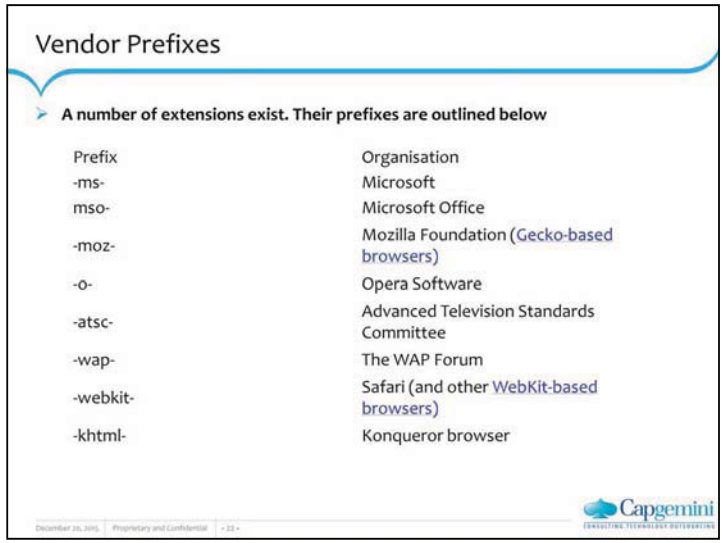
Safari and Chrome support an alternative, the -webkit-box-flex property.


Demo : Flexi Box Layout
fexi_box_layout.html
Capgemini



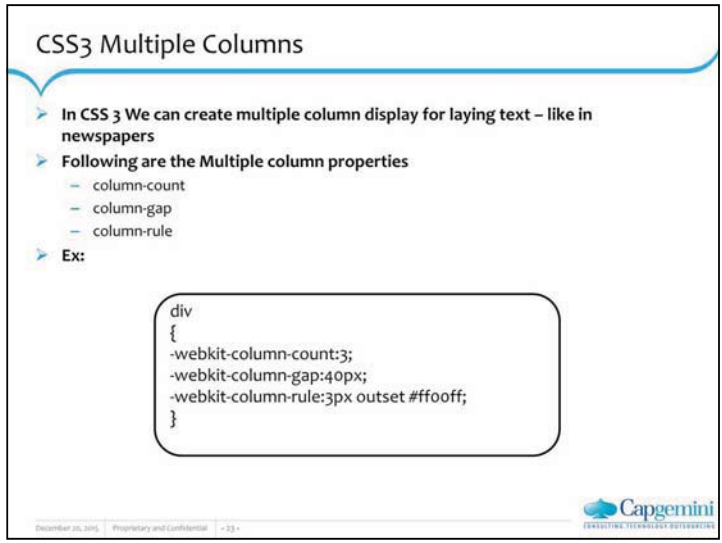
They may do this for a number of reasons, such as adding new features for users, or for experiments and debugging. Most often, though, the extensions are used to release and test browser features that have been developed in the preparation of W3C drafts that have not yet reached Candidate Recommendation status—the extensions allow these new properties to be widely tested before they become available as standard CSS properties.

Current format allows any vendor-specific extension to coexist with any future (or current) CSS properties without causing conflicts because, according to the W3C specifications, a CSS property name will never begin with a dash or an underscore



Even though vendor-specific extensions are guaranteed not to cause conflicts, it should be recognized that these extensions may also be subject to change at the vendor's whim, as they don't form part of the CSS specifications

Although these extensions can be useful at times, it's still recommended that you avoid using them unless it's absolutely necessary



Column Count:

The column-count property specifies the number of columns an element should be divided into:

Column Gap:

The column-gap property specifies the gap between the columns

Column Rule:

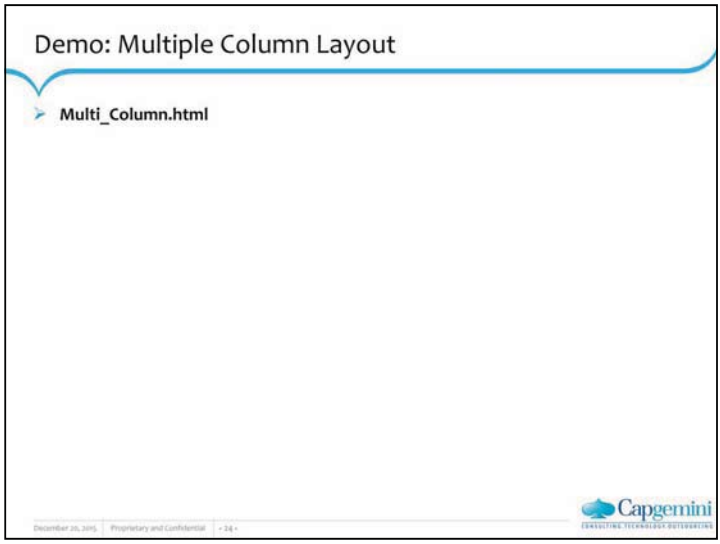
The column-rule property sets the width, style, and color of the rule between columns.

Note:

Internet Explorer does not yet support the multiple columns properties.

Firefox requires the prefix -moz-.

Chrome and Safari require the prefix -webkit-.


Demo: Multiple Column Layout
Multi_Column.html
Capgemini

Lesson Summary

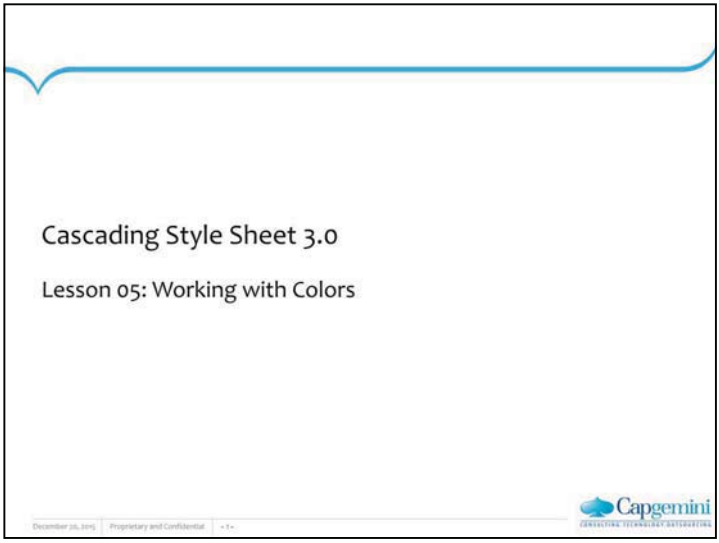
- **We have so far seen:**
  - Layout
  - Positioning
  - Box Layout
  - Table Layout
  - Vendor Prefixes
  - Working with Columns

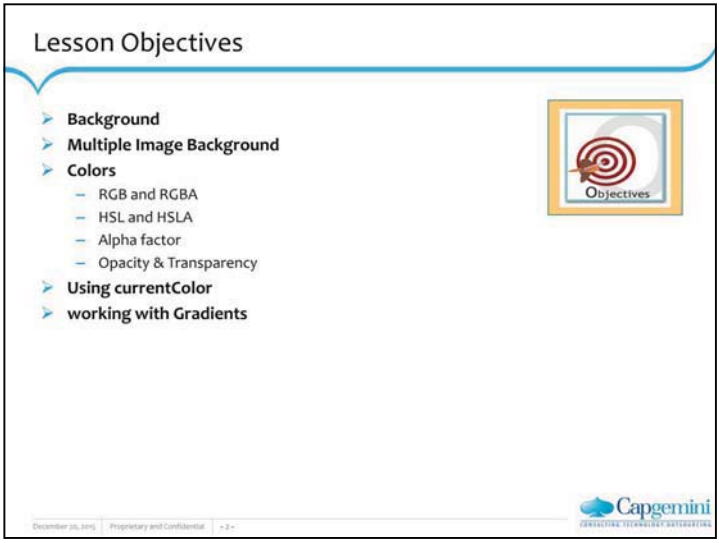
Review Questions

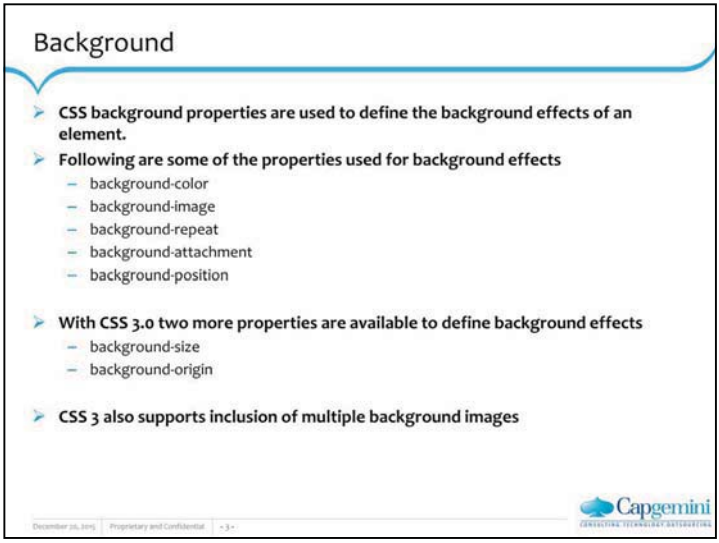
- **Question 1: Which of the following positioning mechanism makes element to be fixed when the web page is resized.**
  - Fixed
  - Absolute
  - Static
  - Relative

- **Question 2: Which of the following is not a property of Multiple Column layout**
  - column-count
  - column-space
  - column-rule
  - None

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

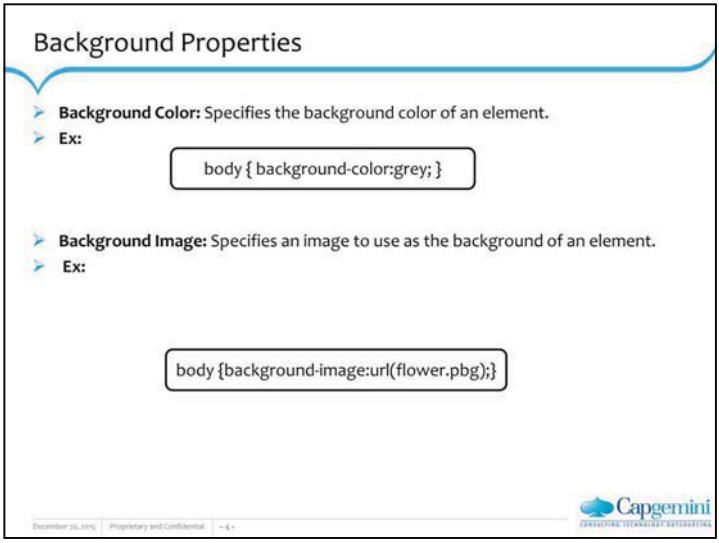

Cascading Style Sheet 3.0
Lesson 05: Working with Colors
Capgemini


Lesson Objectives
Background
Multiple Image Background
Colors
RGB and RGBA
HSL and HSLA
Alpha factor
Opacity & Transparency
Using currentColor
working with Gradients
Objectives
Capgemini



All CSS Background Properties

Property	Description
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated



Background Color: Color can be specified by name like "RED", RGB Value like rgb(255,200,0) or Hex value like #FF00FF

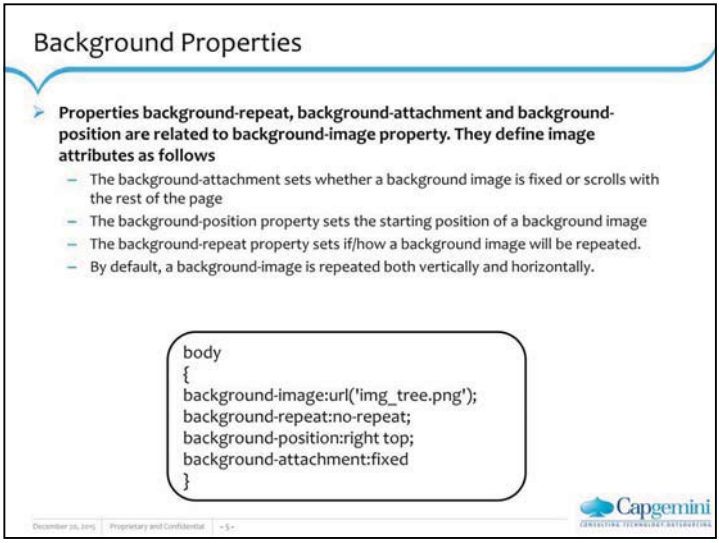
Ex:

```
h1 {background-color:#6495ed;}
p {background-color:rgb(255,200,0) ;}
div {background-color:RED;}
```

In the example above, the h1, p, and div elements have different background colors:

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

By default, the image is repeated so it covers the entire element.



Background Repeat Properties:

No-repeat: Image will not be repeated

Repeat-x : Image repeats horizontally

Repeat-y : Image repeats vertically

By default, a background-image is repeated both vertically and horizontally.

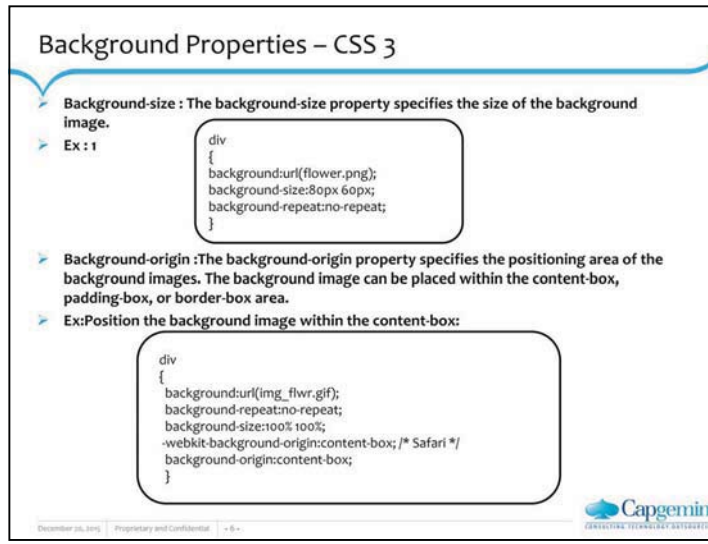
Background Attachment Properties:

Fixed :The value fixed stops the background-image from scrolling with its containing block

Scroll :The value scroll allows the background-image to scroll along with the document. When it's used on an element that has a scrollbar

Background - Shorthand property: It is also possible to specify all the properties in one single property. This is called a shorthand property. The shorthand property for background is simply "background":

Ex: body {background:#ffffff url('img_tree.png') no-repeat right top;}



Background Size:Before CSS3, the background image size was determined by the actual size of the image.

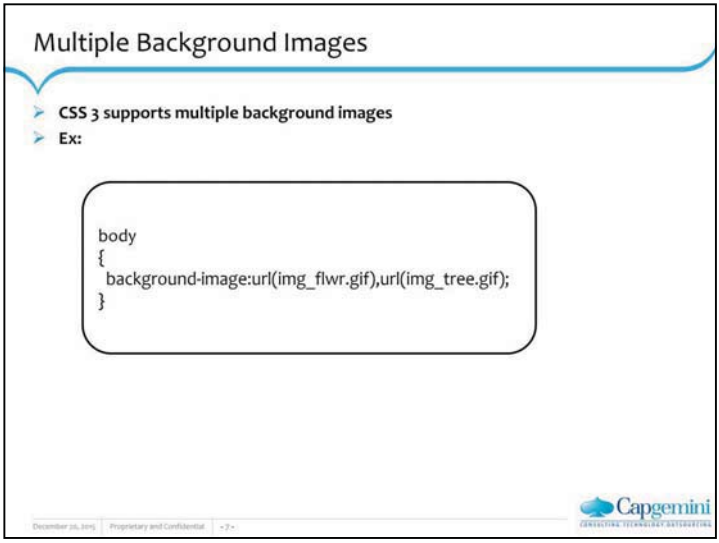
In CSS3 it is possible to specify the size of the background image, which allows us to re-use background images in different contexts.

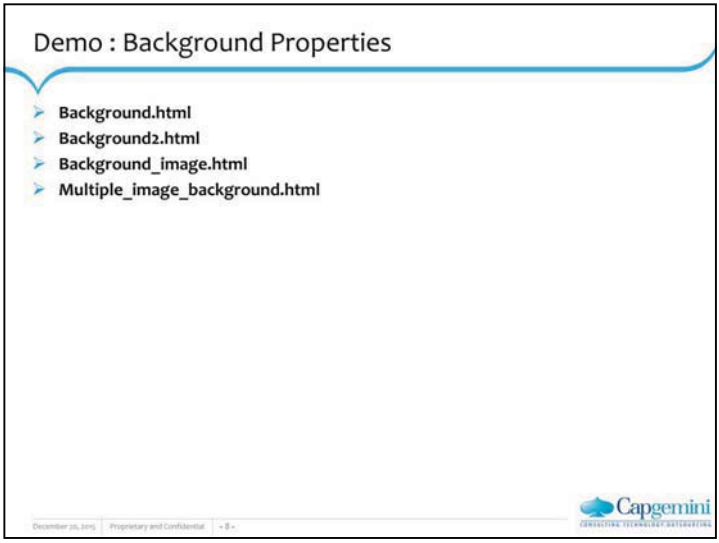
You can specify the size in pixels or in percentages. If you specify the size as a percentage, the size is relative to the width and height of the parent element

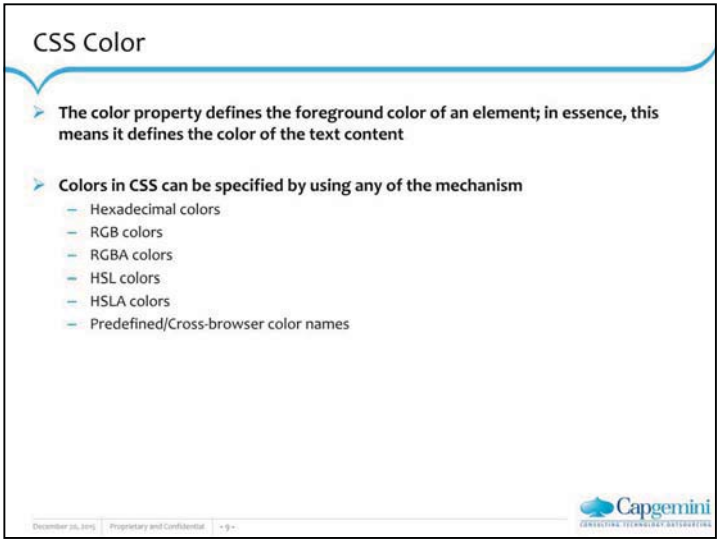
Ex : 2 - Stretch the background image to completely fill the content area:

```
div
{
background:url(flower.png);
background-size:100% 100%;
background-repeat:no-repeat;
}
```

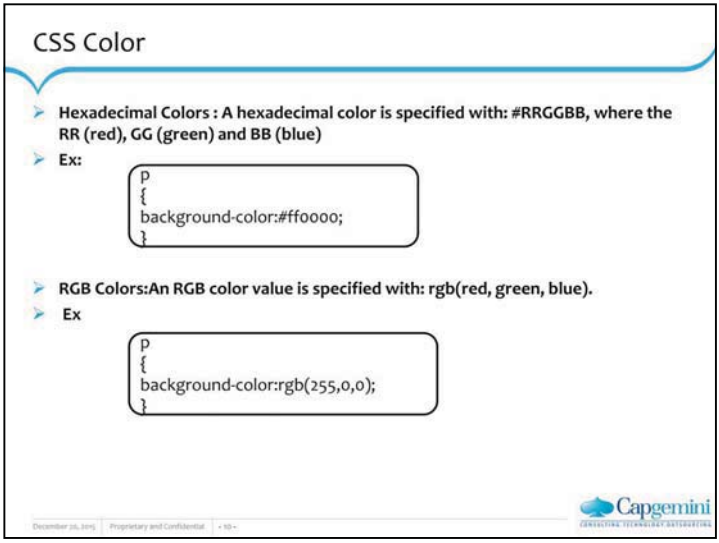
Background Origin:


Multiple Background Images
CSS 3 supports multiple background images
Ex:
body
{
background-image:url(img_flwr.gif),url(img_tree.gif);
}
Capgemini


Demo : Background Properties
Background.html
Background2.html
Background_image.html
Multiple_image_background.html
Capgemini



Note :If a border-color value hasn't been defined explicitly for the element, the color value will be used instead.



Hexadecimal Colors:

Hexadecimal color values are supported in all major browsers.

Hexadecimal integers specify the components of the color. All values must be between 0 and FF.

For example, the #0000ff value is rendered as blue, because the blue component is set to its highest value (ff) and the others are set to 0.

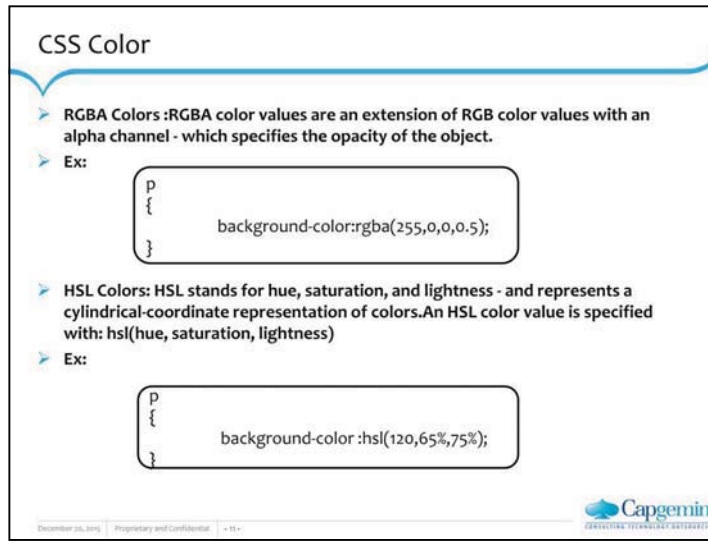
RGB Colors:

RGB color values are supported in all major browsers.

An RGB color notation rgb(red, green, blue), Each parameter (red, green, and blue) defines the intensity of the color and can be an integer between 0 and 255 or a percentage value (from 0% to 100%).

For example, the rgb(0,0,255) value is rendered as blue, because the blue parameter is set to its highest value (255) and the others are set to 0.

Also, the following values define the same color: rgb(0,0,255) and rgb(0%,0%,100%).



RGBA Colors:

RGBA color values are supported in IE9+, Firefox 3+, Chrome, Safari, and in Opera 10+.

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity of the object.

An RGBA color value is specified with: rgba(red, green, blue, alpha). The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

HSL Colors:

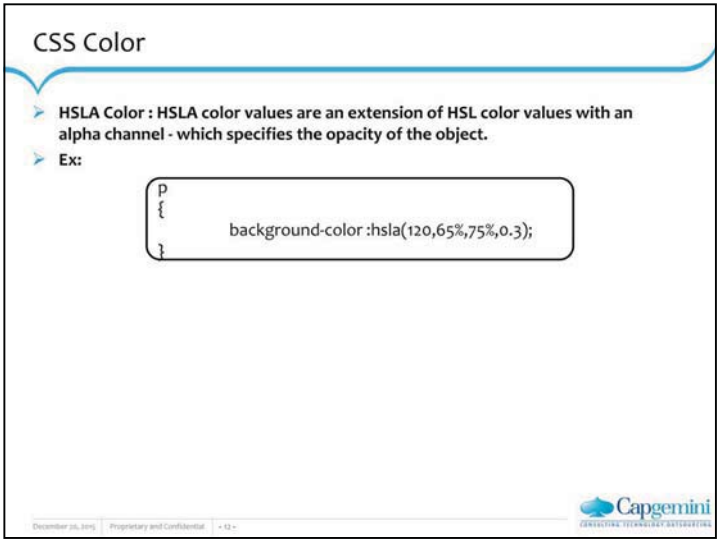
HSL color values are supported in IE9+, Firefox, Chrome, Safari, and in Opera 10+.

HSL stands for hue, saturation, and lightness - and represents a cylindrical-coordinate representation of colors.

Hue is a degree on the color wheel (from 0 to 360) - 0 (or 360) is red, 120 is green, 240 is blue.

Saturation is a percentage value; 0% means a shade of gray and 100% is the full color.

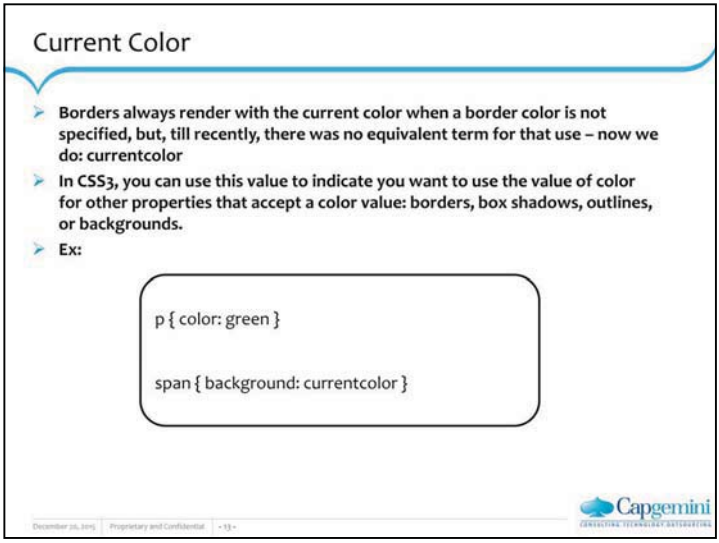
Lightness is also a percentage; 0% is black, 100% is white.



HSLA Colors

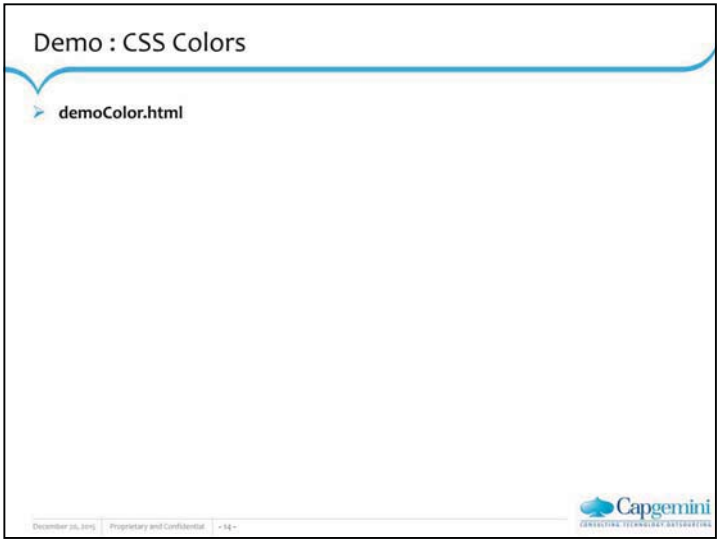
HSLA color values are supported in IE9+, Firefox 3+, Chrome, Safari, and in Opera 10+.

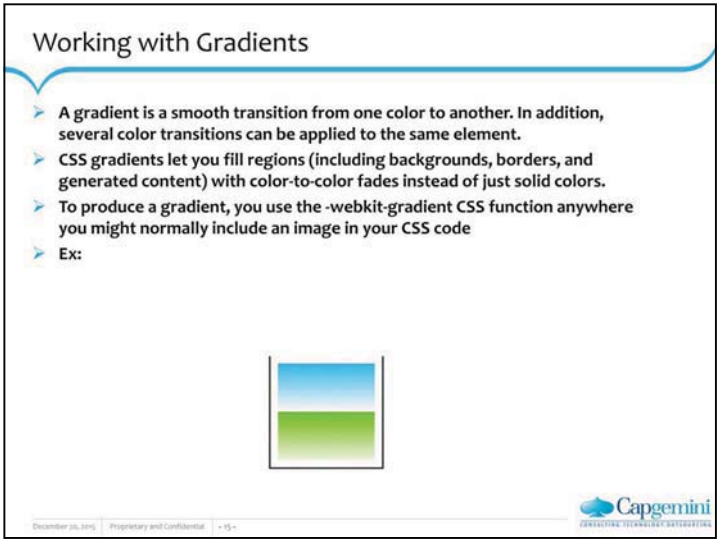
An HSLA color value is specified with: hsla(hue, saturation, lightness, alpha), where the alpha parameter defines the opacity. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).



Note:

This value is supported in Firefox 3.5+, Chrome 1+, and Safari 4+. IE 9 yet does not support this though.


Demo : CSS Colors
demoColor.html
Capgemini



In CSS there are two types of gradients: linear gradients and radial gradients.

Ex:

background: -webkit-gradient(linear, left top, left bottom, from(#00abeb), to(#fff), color-stop(0.5, #fff), color-stop(0.5, #66cc00));

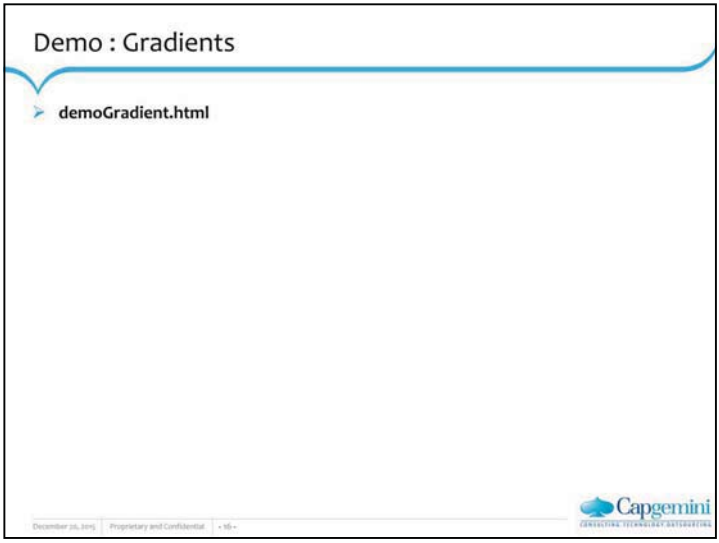
A point is a pair of space-separated values. The syntax supports numbers, percentages or the keywords top, bottom, left and right for point values.

A radius is a number and may only be specified when the gradient type is radial.

A stop is a function, color-stop, that takes two arguments, the stop value (either a percentage or a number between 0 and 1.0), and a color (any valid CSS color).

In addition the shorthand functions from and to are supported.

These functions only require a color argument and are equivalent to color-stop(0, ...) and color-stop(1.0, …) respectively.


Demo : Gradients
demoGradient.html
Capgemini

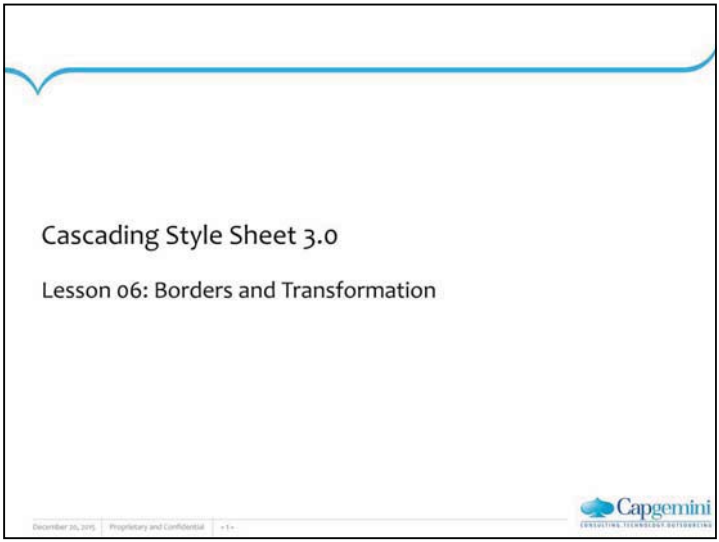
Lesson Summary

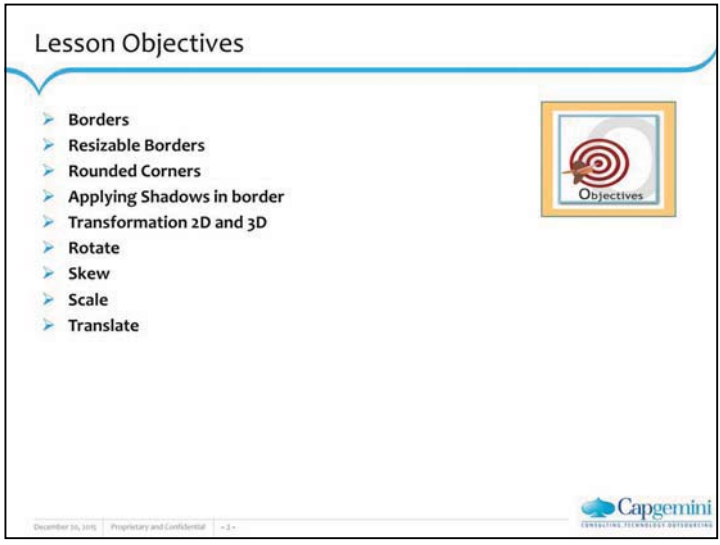
- **In this lesson, you have learn about**
  - Background
  - Multiple Image Background
  - Colors
    - RGB and RGBA
    - HSL and HSLA
    - Alpha factor
    - Opacity & Transparency
  - Using currentColor
  - working with Gradients

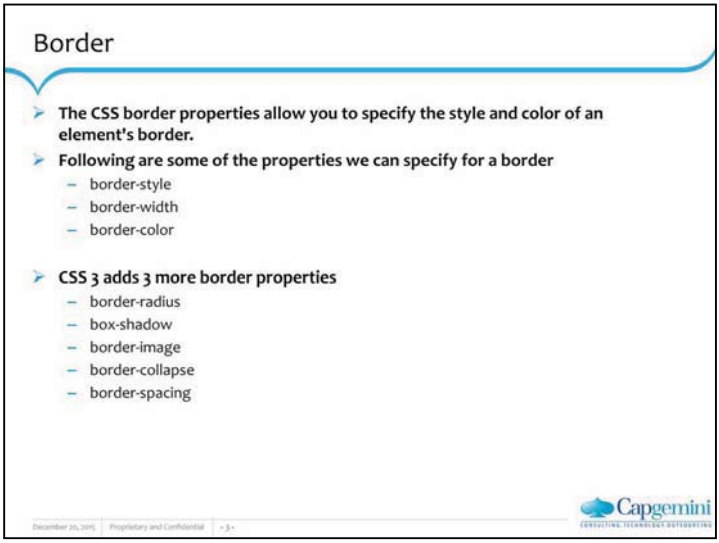
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

Review Questions

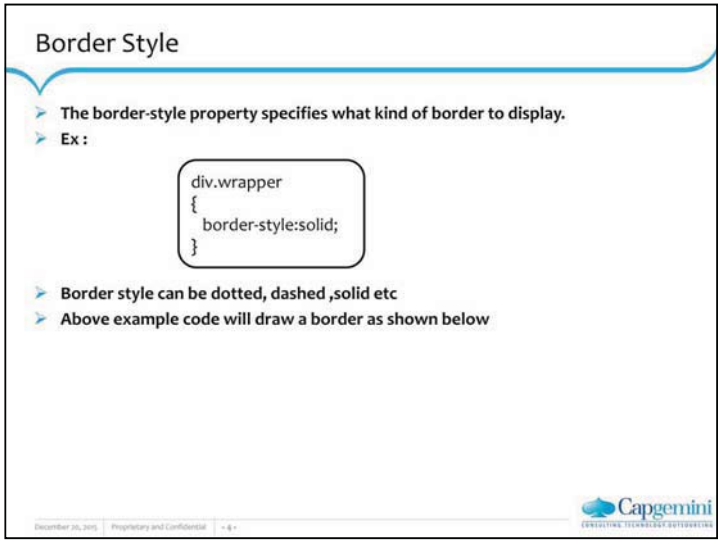
- **Question 1: In HSLA color scheme H,S,L,A stands for _____,_____,_____and _____ respectively**
- **Question 2: With Background Attachment _____ property stops the background-image from scrolling with its containing bloc**
  - Stop
  - No-scroll
  - Static
  - Fixed


Cascading Style Sheet 3.0
Lesson 06: Borders and Transformation
Capgemini


Lesson Objectives
Borders
Resizable Borders
Rounded Corners
Applying Shadows in border
Transformation 2D and 3D
Rotate
Skew
Scale
Translate
Objectives
Capgemini

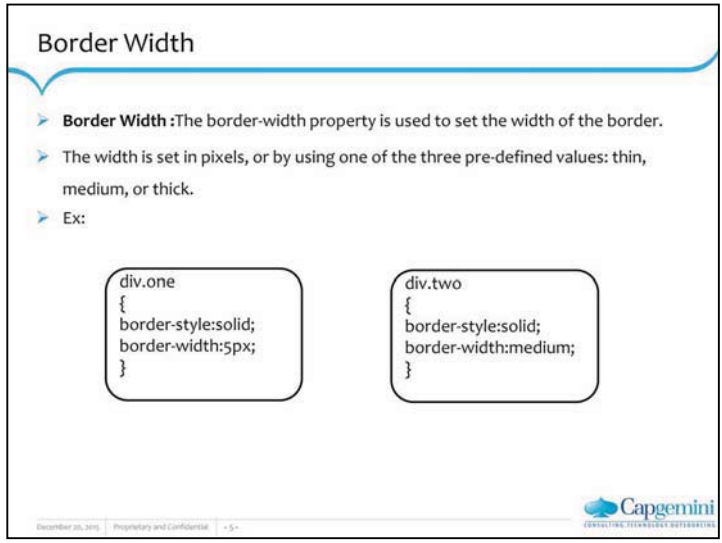


Note: None of the border properties will have ANY effect unless the border-style property is set!

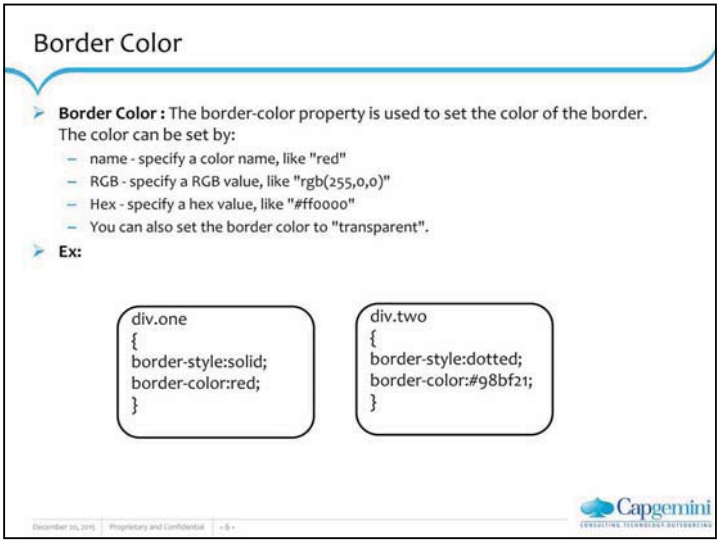


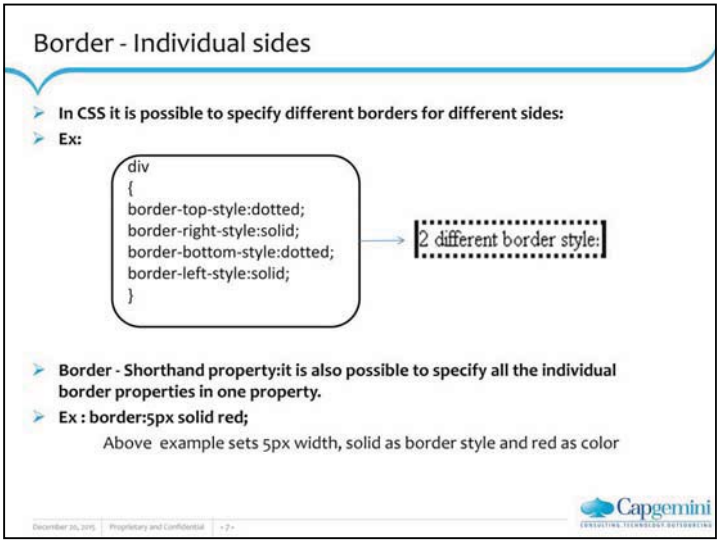
Border-style values:

Style	Description
dotted	Defines a dotted border
dashed	Defines a dashed border
solid	Defines a solid border
double	Defines two borders. The width of the two borders are the same as the border-width value
groove	Defines a 3D grooved border. The effect depends on the border-color value
ridge	Defines a 3D ridged border. The effect depends on the border-color value
inset	Defines a 3D inset border. The effect depends on the border-color value
outset	Defines a 3D outset border. The effect depends on the border-color value



Note: The "border-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.


Border Color
Border Color : The border-color property is used to set the color of the border. The color can be set by:
name - specify a color name, like "red"
RGB - specify a RGB value, like "rgb(255,0,0)"
Hex - specify a hex value, like "#ff0000"
You can also set the border color to "transparent".
Ex:
div.one
{
border-style:solid;
border-color:red;
}
div.two
{
border-style:dotted;
border-color:#98bf21;
}
Capgemini



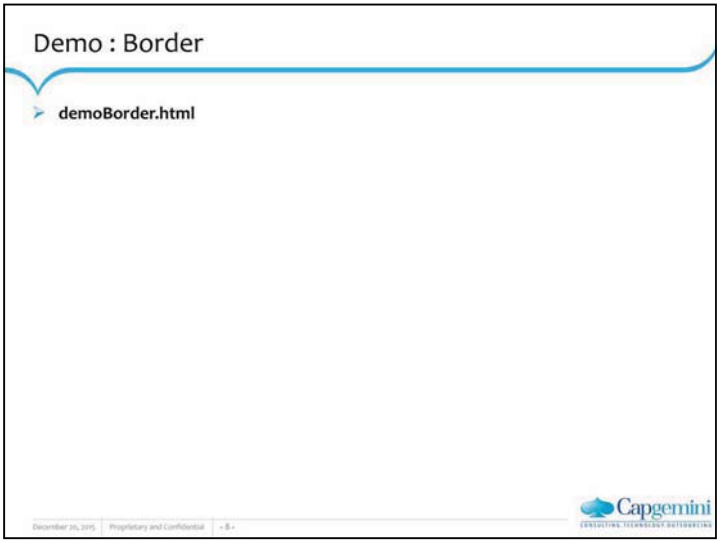
Note :All the four border sides can be of different style
The border-style property can have from one to four values:

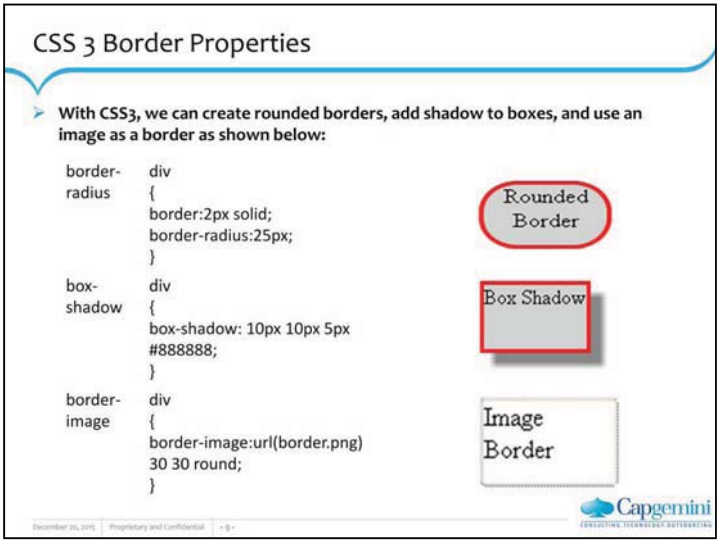
border-style:dotted solid double dashed;
- top border is dotted
- right border is solid
- bottom border is double
- left border is dashed

border-style:dotted solid double;
- top border is dotted
- right and left borders are solid
- bottom border is double

border-style:dotted solid;
- top and bottom borders are dotted
- right and left borders are solid

border-style:dotted;
- all four borders are dotted

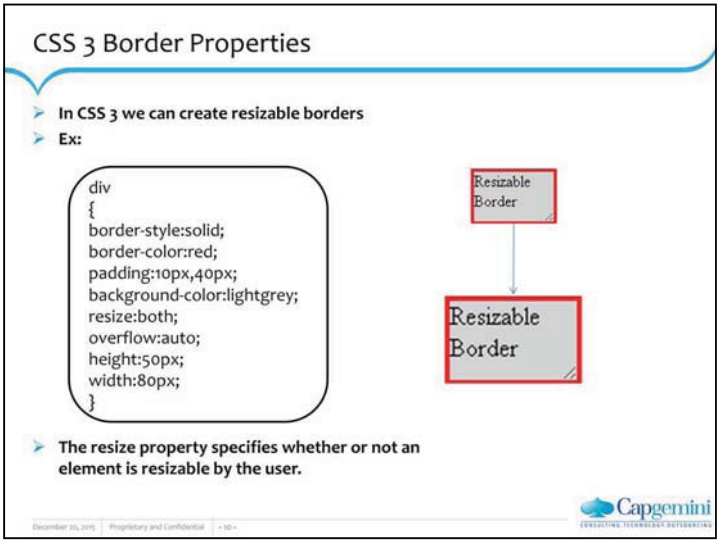

Demo : Border
demoBorder.html
Capgemini



Rounded Corners: In CSS3, creating rounded corners is easy. The border-radius property is used to create rounded corners:
Box Shadow: In CSS3, the box-shadow property is used to add shadow to boxes:
Border Image: With the CSS3 border-image property you can use an image to create a border:

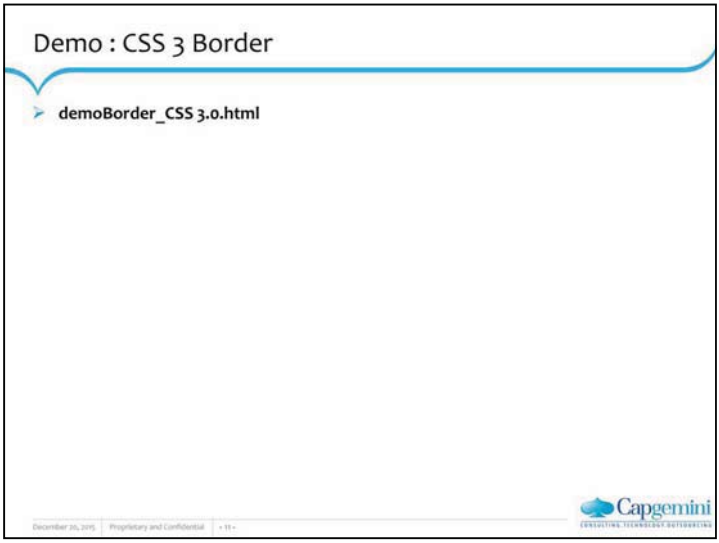
Use following extension for different browser:

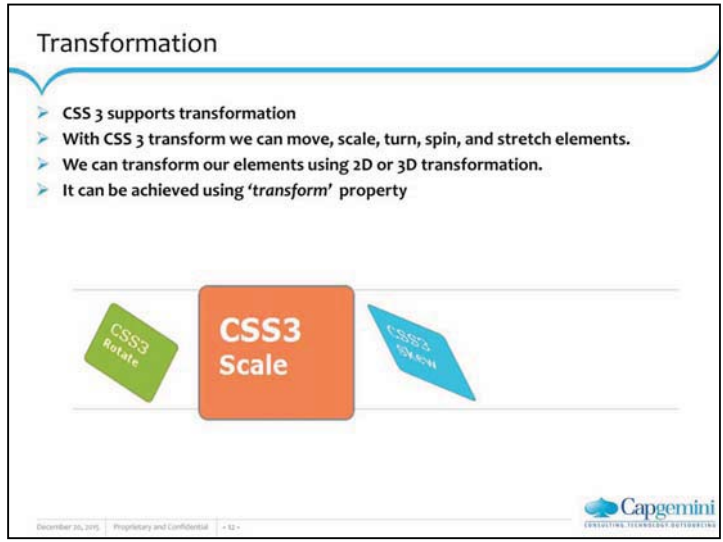
-webkit-	for Safari and Chrome
-moz-	for old firefox
-o-	for opera



Resize property can take following values:

none : The user cannot resize the element
both : The user can adjust both the height and the width of the element
horizontal : The user can adjust the width of the element
vertical: The user can adjust the height of the element


Demo : CSS 3 Border
demoBorder_CSS 3.0.html
Capgemini

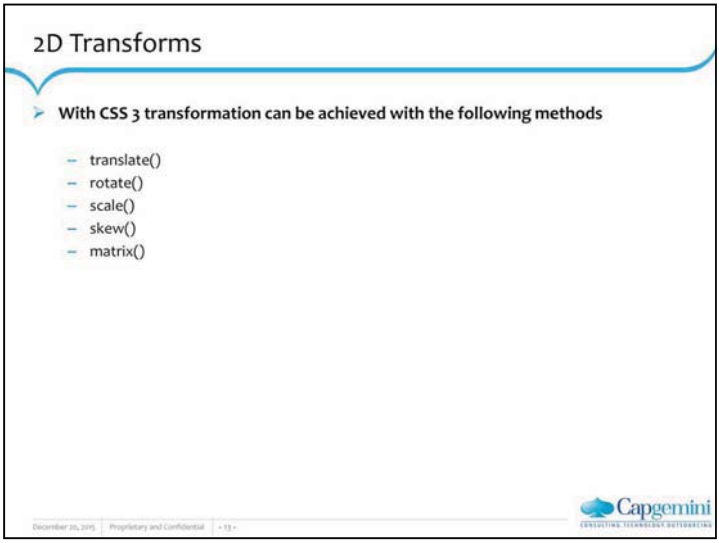


A transform is an effect that lets an element change shape, size and position.

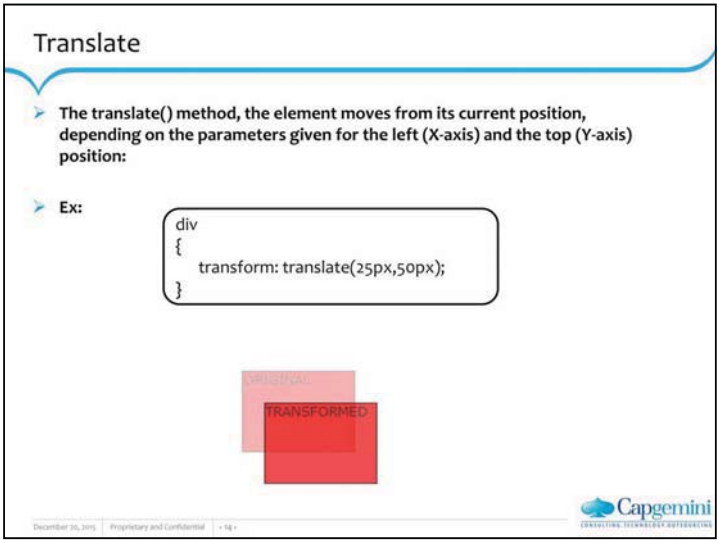
CSS3 transforms can also be done using other web technologies. For example, using JavaScript, SVG, or Canvas

But the overhead and maintainability costs of using those other methods for simple element transformations is a huge drawback, and so the CSS3 Transforms module helps alleviate some of those problems

With simple CSS-based syntax, and (ideally) no unnecessary JavaScript, element transformations become much easier to work with.



Function	Description
matrix(*n,n,n,n,n,n*)	Defines a 2D transformation, using a matrix of six values
translate(*x,y*)	Defines a 2D translation, moving the element along the X- and the Y-axis
translateX(*n*)	Defines a 2D translation, moving the element along the X-axis
translateY(*n*)	Defines a 2D translation, moving the element along the Y-axis
scale(*x,y*)	Defines a 2D scale transformation, changing the elements width and height
scaleX(*n*)	Defines a 2D scale transformation, changing the element's width
scaleY(*n*)	Defines a 2D scale transformation, changing the element's height
rotate(*angle*)	Defines a 2D rotation, the angle is specified in the parameter
skew(*x-angle,y-angle*)	Defines a 2D skew transformation along the X- and the Y-axis
skewX(*angle*)	Defines a 2D skew transformation along the X-axis
skewY(*angle*)	Defines a 2D skew transformation along the Y-axis



In the example above, the translate function is passed two parameters: The first represents positioning along the X axis, while the second parameter (which is optional) represents positioning along the Y axis. So in this case, the element in question will be positioned 25px to the right (or horizontally) and 50px down (or vertically).

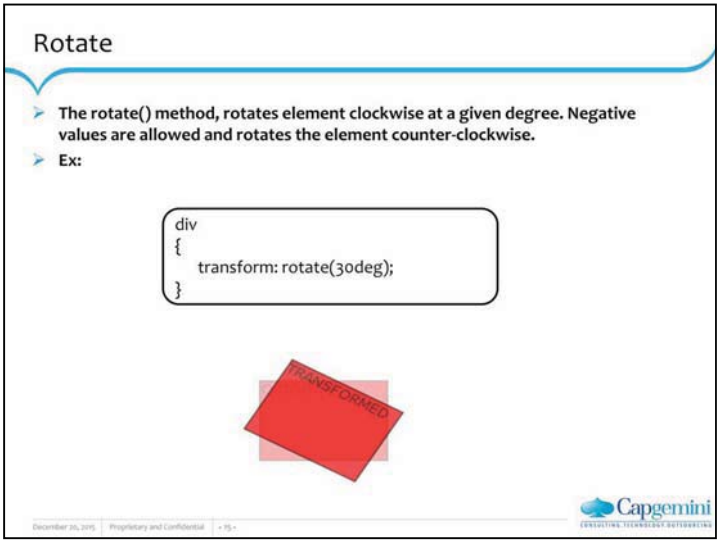
The second parameter (the position along the Y axis) is optional, but if omitted its initial value is zero. You also have the option to use other units (percentages or ems, for example), and negative values are permitted

you can also declare translation functions for the X and Y axes individually using the translateX and translateY functions, each of which accepts a single parameter

Note:

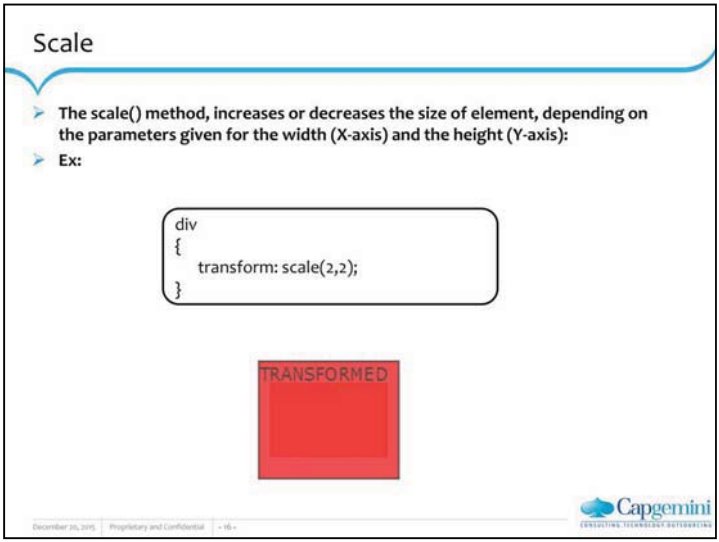
Use the following extension depending on browser:

```
-ms-transform: translate(50px,100px); /* IE 9 */
-webkit-transform: translate(50px,100px); /* Safari and Chrome */
-o-transform: translate(50px,100px); /* Opera */
-moz-transform: translate(50px,100px); /* Firefox */
```



The rotate function takes just one parameter and, as mentioned, positive or negative values will determine in what direction the element is rotated. The unit used for defining the angle of rotation is degrees (expressed as "deg") and you are permitted to use decimal notation

Note : Use browser specific extension



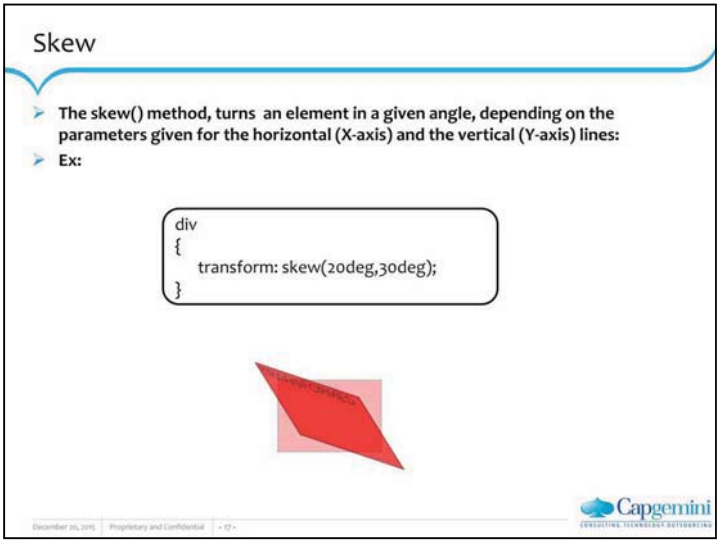
The value scale(2,2) transforms the width and height to be twice its original size

The value of each parameter must be a number without units. Decimal notation is permitted, and if the second parameter is omitted, it will assume the same value as the first parameter (which is different from the translate function that defaults the second missing parameter to zero).

A value of scale(1) will keep the element at its current size (which would be the same as omitting the transform property altogether). A value of "0" would shrink the element down to nothing, which is virtually equivalent to display: none.

As with the translate function, you can use either scaleX or scaleY to scale along only a single axis.

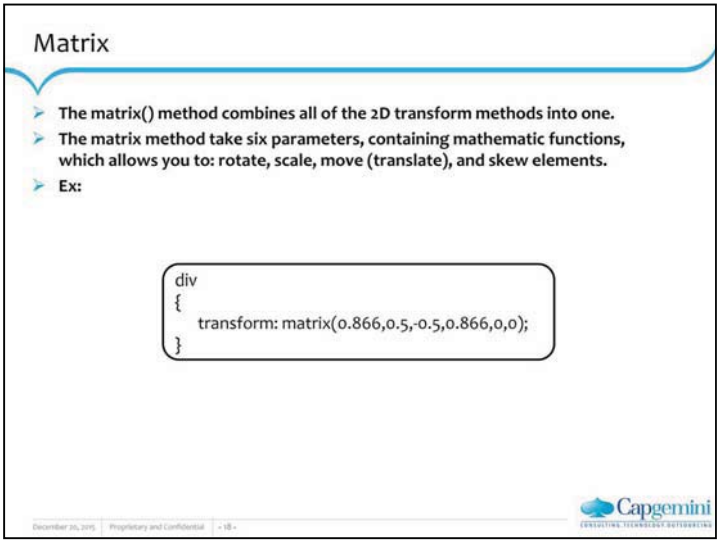
Note : Use browser specific extension



Both parameters are expressed in degrees, and the second parameter is optional. If the second is omitted, a value of "0" is assumed for the skew along the Y axis (which means there is no skew).

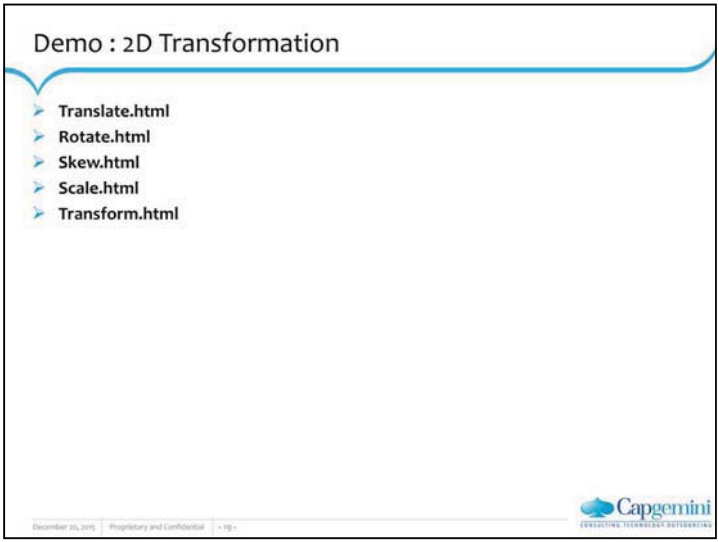
Again, you have the option to use skewX and skewY to target a single axis.

Note : Use browser specific extension

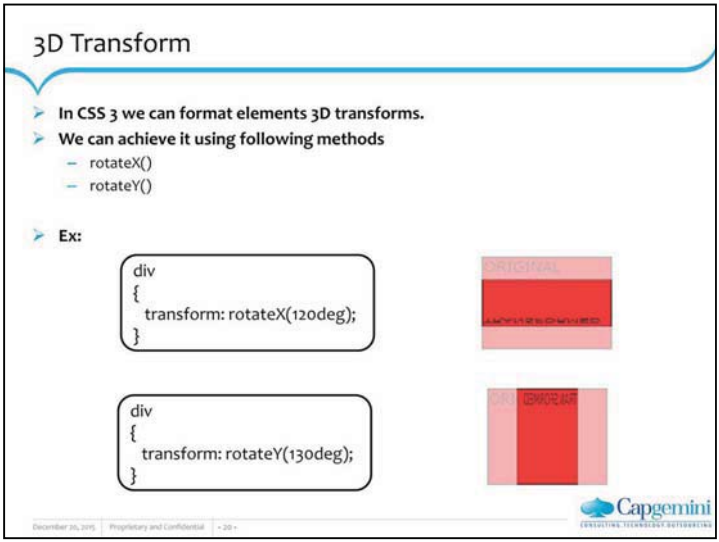


Above example rotates a div element 30 degrees, using the matrix method:

Note : Use browser specific extension

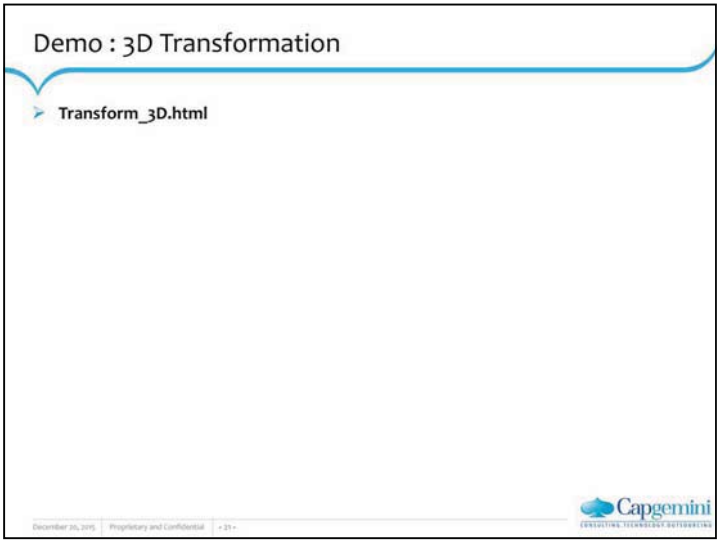


In Lesson06 folder, refer the examples which is available with above mentioned filename.



rotateX(): the element rotates around its X-axis at a given degree.
rotateY(): the element rotates around its Y-axis at a given degree.

Note:
Internet Explorer and Opera does not yet support 3D transforms (They support only 2D transforms).
Firefox requires the prefix -moz-.
Chrome and Safari requires the prefix -webkit-.


Demo : 3D Transformation
Transform_3D.html
Capgemini

Lesson Summary

- **In this lesson, you have learnt about**
  - Borders
  - Resizable Borders
  - Rounded Corners
  - Applying Shadows in border
  - Transformation 2D and 3D
  - Rotate
  - Skew
  - Scale
  - Translate

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

Review Questions

- **Question 1: Which of the following are CSS 3 properties**
  - border-image
  - border-width
  - border-color
  - border-radius
  - box-shadow

- **Question 2: In CSS it is possible to specify different borders for different sides:**
  - True
  - False

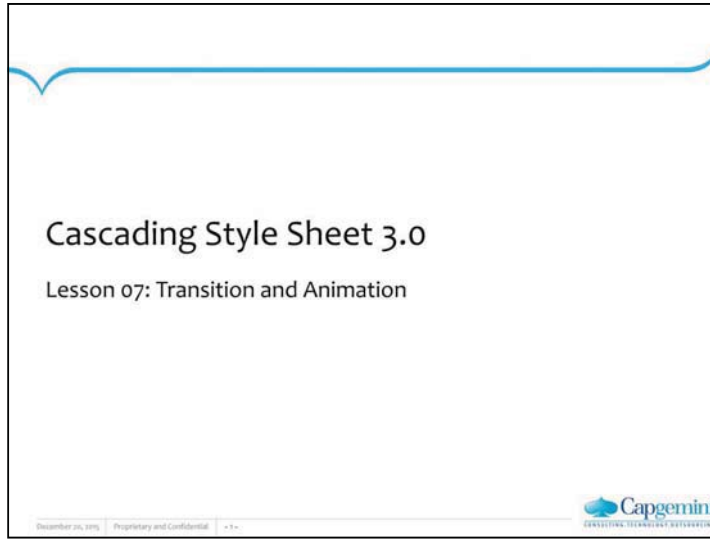
Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

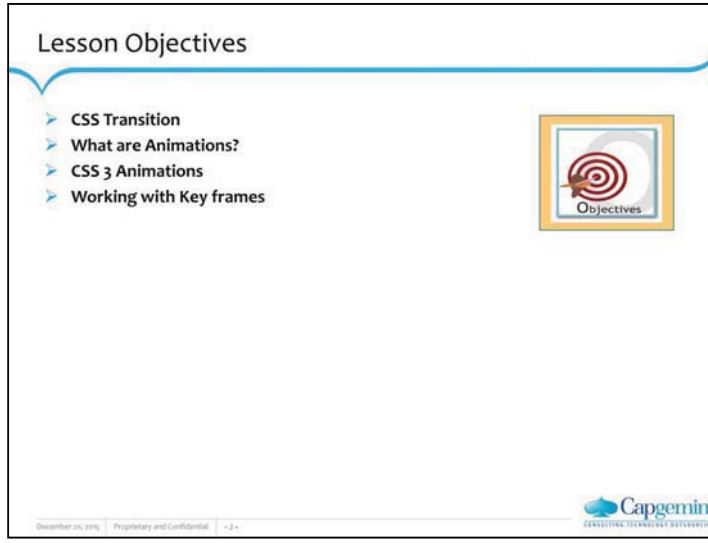
Review Questions

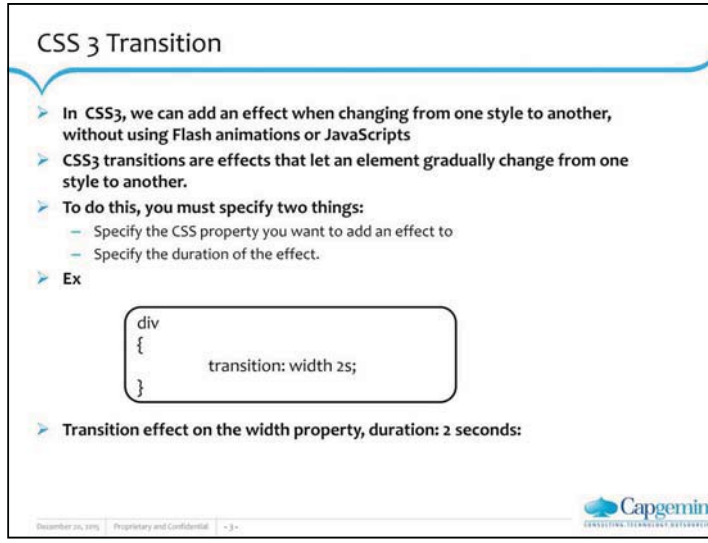
- **Question 3: Given transform: scale(2) , What will be the second parameter**
  - 0
  - 1
  - 2
  - Will not work , 2nd argument must be specified

- **Question 4: _____ method, turns an element in a given angle, depending on the parameters given for the horizontal (X-axis) and the vertical (Y-axis) lines Selector**
  - translate
  - skew
  - scale
  - rotate

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING.


Cascading Style Sheet 3.0
Lesson 07: Transition and Animation
Capgemini


Lesson Objectives
CSS Transition
What are Animations?
CSS 3 Animations
Working with Key frames
Objectives
Capgemini



If the duration is not specified, the transition will have no effect, because default value is 0.
The effect will start when the specified CSS property changes value.
A typical CSS property change would be when a user mouse-over an element:

Multiple changes: To add a transitional effect for more than one style, add more properties, separated by commas:

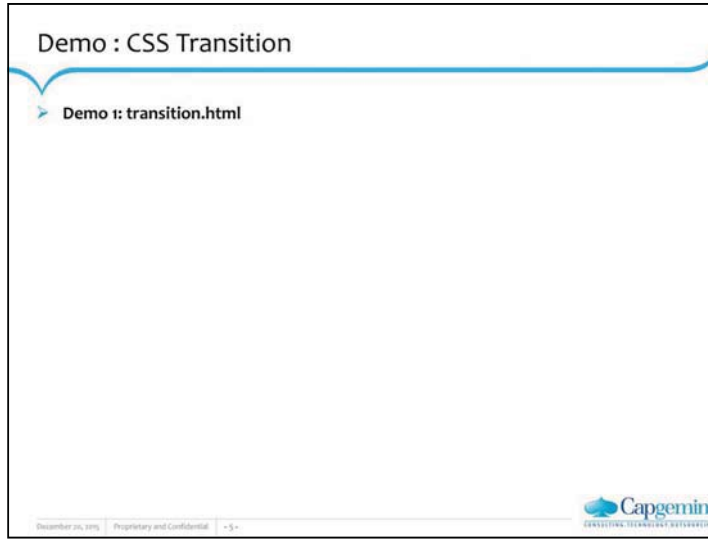
Ex :

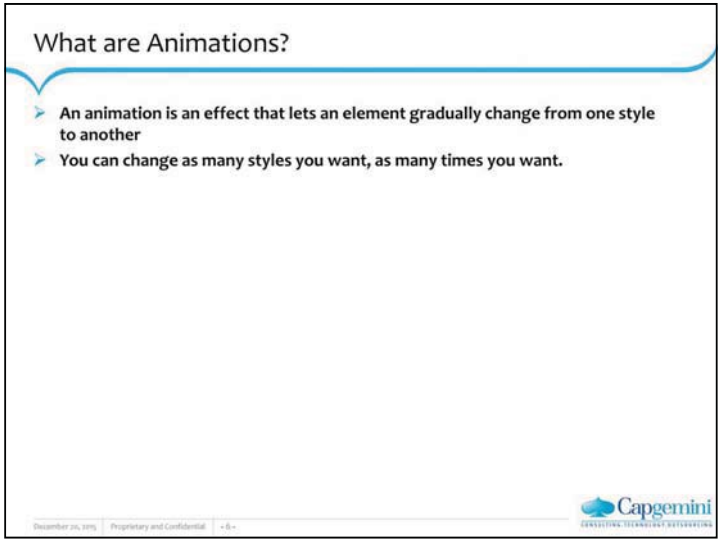
```
div
{
transition: width 2s, height 2s, transform 2s;
}
```

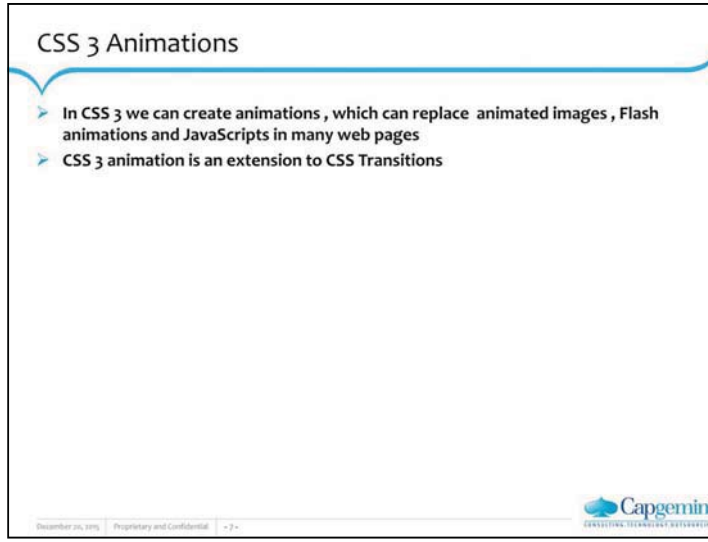
Note:
Internet Explorer does not yet support the transition property.
Firefox 4 requires the prefix -moz-.
Chrome and Safari requires the prefix -webkit-.
Opera requires the prefix -o-.

Transition Properties:

Property	Description	CSS
transition	A shorthand property for setting the four transition properties into a single property	3
transition-property	Specifies the name of the CSS property to which the transition is applied	3
transition-duration	Defines the length of time that a transition takes. Default 0	3
transition-timing-function	Describes how the speed during a transition will be calculated. Default "ease"	3
transition-delay	Defines when the transition will start. Default 0	

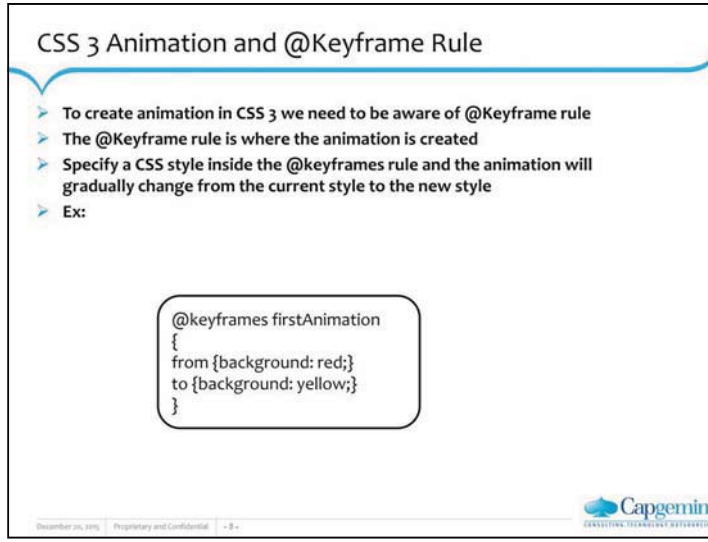

Demo : CSS Transition
Demo 1: transition.html
Capgemini


What are Animations?
An animation is an effect that lets an element gradually change from one style to another
You can change as many styles you want, as many times you want.
Capgemini



Today's HTML5 applications can provide awesome experiences thanks to the new CSS3 specifications. One of them is CSS3 Animations. It can help you building rich animations on HTML elements. This can provide interesting feedbacks to the users and enables fast & fluid UIs. As those new animations are most of the time hardware accelerated by the GPU, they definitely raise the quality bar of the new generation of HTML5 applications.

CSS3 Animations introduces defined animations, which specify the values that CSS properties will take over a given time interval. This specification is an extension to CSS Transitions.



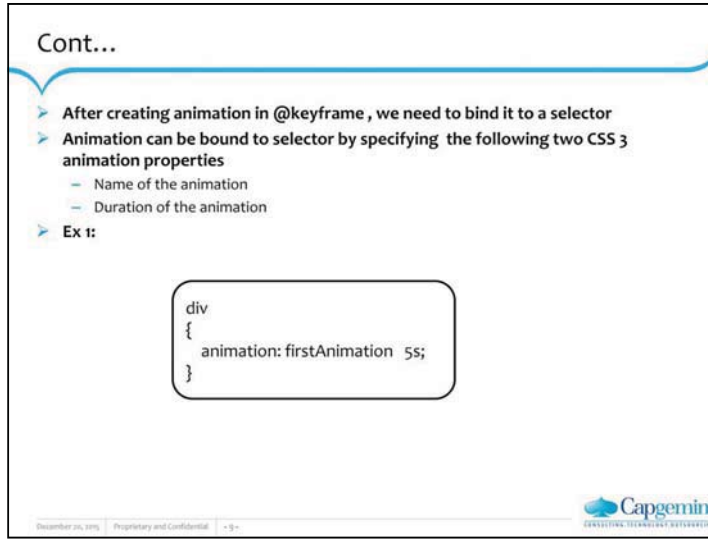
Note:

Internet Explorer does not yet support the @keyframes rule or the animation property.

Firefox requires the prefix -moz-,

Chrome and Safari require the prefix -webkit-, and

Opera require the prefix -o-



Note:

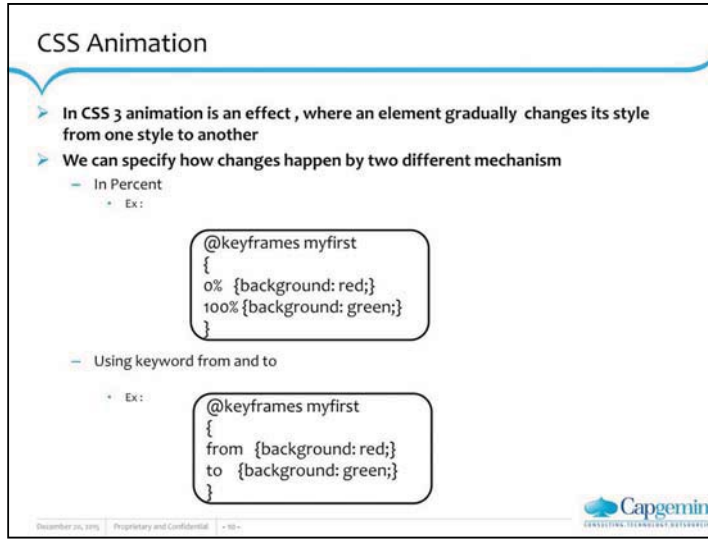
If animation is not bind to a selector it will not have any effect

We must define the name and the duration of the animation. If duration is omitted, the animation will not run, because the default value is 0.

Use browser specifc extensions

List of CSS 3 Animation Properties:

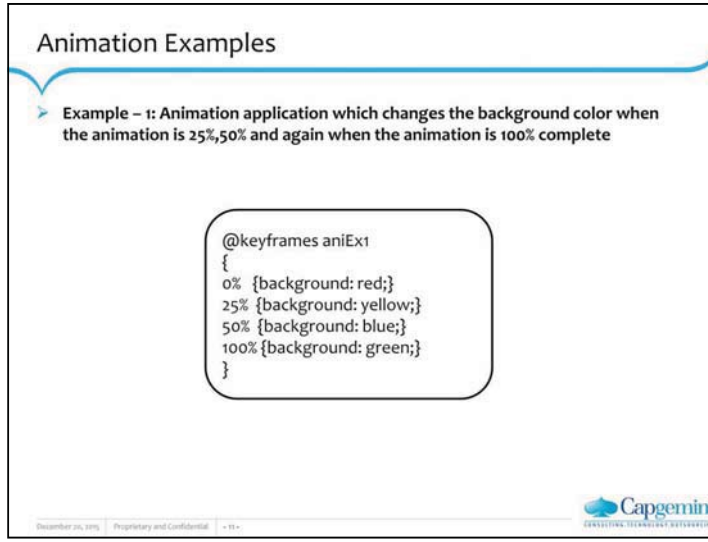
Property	Description	CSS
@keyframes	Specifies the animation	3
animation	A shorthand property for all the the animation properties, except the animation-play-state property	3
animation-name	Specifies the name of the @keyframes animation	3
animation-duration	Specifies how many seconds or milliseconds an animation takes to complete one cycle. Default 0	3
animation-timing-function	Describes how the animation will progress over one cycle of its duration. Default "ease"	3
animation-delay	Specifies when the animation will start. Default 0	3
animation-iteration-count	Specifies the number of times an animation is played. Default 1	3
animation-direction	Specifies whether or not the animation should play in reverse on alternate cycles. Default "normal"	3
animation-play-state	Specifies whether the animation is running or paused. Default "running"	3



Note:

0% is the beginning of the animation, 100% is when the animation is complete.

For best browser support, you should always define both the 0% and the 100% selectors.

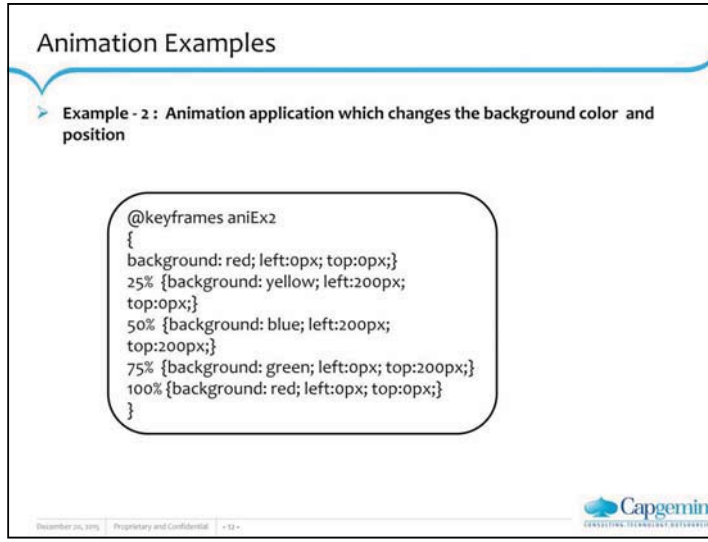


Associate the animation with a selector by specifying Name and Duration of animation as followes:

```
<body>
  <div id="anidiv">

    Animation works here!!!

  </div >
</body>
```

CSS Code:

```
#anidiv
{
  border-style:solid:
  border-color:black;
  height:100px;
  width:100px;
  animation: aniEx1 5s;
}
```

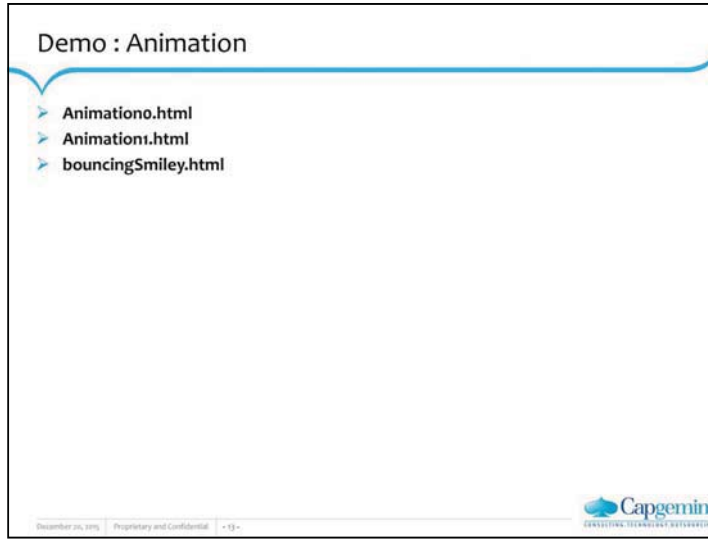


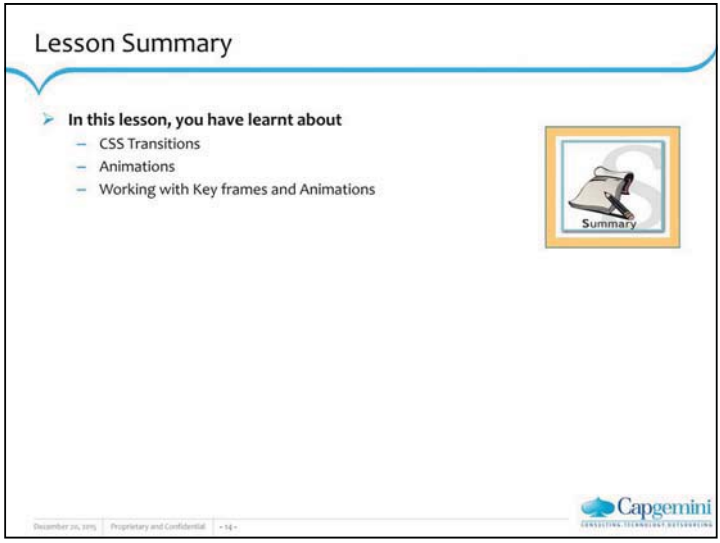
Associate the animation with a selector by specifying Name and Duration of animation as followes:

```
<body>
  <div id="anidiv">

    Animation works here!!!

  </div >
</body>
```

CSS Code:

```
#anidiv
{
  border-style:solid:
  border-color:black;
  height:100px;
  width:100px;
  animation: aniEx2 5s;
}
```


Demo : Animation
Animation0.html
Animation1.html
bouncingSmiley.html
Capgemini


Lesson Summary
In this lesson, you have learnt about
CSS Transitions
Animations
Working with Key frames and Animations
Summary
Capgemini

Web Basics – CSS 3.0

Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
31-Dec-2012	1.0	Mohan Chinnaiah	Initial Draft
21-Apr-2015	2.0	Rathnajothi P	Revamp/Refinement as per revised TOC

Table of Contents

Getting Started

Overview

This lab book is a guided tour for learning CSS 3.0. It comprises solved examples and 'To Do' assignments. Follow the steps provided in the solved examples and work out the 'ToDo' assignments given.

Setup Checklist for CSS 3.0

Here is what is expected on your machine in order to work with the lab assignments.

Minimum System Requirements

- Hardware: Networked PCs with minimum 64 MB RAM and 60 MB HDD.
- Software:
  - Window based Operating System having the latest version of Browser (Google Chrome 20.0, Mozilla Firefox 13.0, Opera 12.0 and Internet Explorer 9.0 or higher) is installed.
  - Eclipse Luna or Visual Studio 2008 or an editor like Notepad, Edit Plus etc is installed.

Instructions

- For all coding standards refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory css_assgn. For each lab exercise create a directory as lab<lab number>.
- You may also look up the on-line help provided in the MSDN library.
- The faculty will introduce you to the editor to be used.

Learning More (Bibliography)

www.w3Schools.com
www.csssitepoint.com

Lab 1. Introduction to CSS

Goals	• Learn and understand the process of: o Using inline, embedded and external CSS
Time	10 minutes

1.0 Create an HTML file as shown below

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

      <h1> Introduction to CSS</h1>

      <p> A CSS (cascading style sheet) file allows you to separate your web sites
      (X)HTML content from it's style. As always you use your (X)HTML file to arrange
      the content, but all of the presentation (fonts, colors, background, borders, text
      formatting, link effects & so on...) are accomplished within a CSS. </p>

  </body>
</html>
```

Example 1: Introduction.html

1.1 add the following styles to above html file using inline style sheet

Header Font Color : Red
Paragraph Font Color : Blue
Background Color : Light Grey

1.2 add the above styles using embedded style sheet

1.3 add the above styles using external style sheet

Lab 2. Working with Text and Fonts

Goals	• At the end of this lab session, you will be able to: ○ Text formatting ○ Text effects ○ Fonts and custom fonts
Time	30 min

2.1: Use introduction.html from above exercise and apply following styles using external style sheet

```
h1
{
        font-family: calibri;
        font-size: 30px;
        font-style: normal;
        font-weight: bold;
        color: grey;
        text-decoration: underline;
        text-align: center;
}

p
{
        font-family: verdana;
        font-size: 20px;
        font-style: italic;
}
```

Example 2: Introduction.css

2.2 Use introduction.css and apply the following properties

- **Text Transformation**
- **Text Indentation**
- **Specify text color using Hex value**

2.3 Use introduction.css and apply text-shadow property

2.4 Use introduction.css and apply custom fonts with @font-face to header and paragraph:

Note: Instructor will provide custom fonts.

Lab 3. CSS Selectors

Goals	• At the end of this lab session, you will be able to understand: ○ **Universal Selector** ○ **Type Selector** ○ **Class Selector** ○ **ID Selector** ○ **Attribute Selector** ○ **Pseudo-classes**
Time	30 min

3.0: Given

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

  <h1>Learn CSS 3.0 for better web design</h1>

  <div>Do not go where the path may lead, go instead where there is no path and leave a trail.
</div>

  <div>It is always the simple that produces the marvelous. </div>

  <div>As knowledge increases, wonder deepens.</div>

  <p>For beautiful eyes, look for the good in others; for beautiful lips, speak only words of
kindness; and for poise, walk with the knowledge that you are never alone.</p>

  <p>The best and most beautiful things in the world cannot be seen or even touched - they must
be felt with the heart.</p>

  <p>It is during our darkest moments that we must focus to see the light. </p>

  <p>Happiness is not something you postpone for the future; it is something you design for the
present.</p>

  <p>Be faithful to that which exists within yourself. </p>

</body>
</html>
```

Example 3: Selector1.html

3.1 Use selector.html and apply color attribute using universal selector

3.2 Use selector.html apply different colors to text based on type selectors - H1, Div and P.

3.3 Use selector.html apply id attribute by specifying different values to every paragraph and division (use p1, p2, p3 … for paragraph and d1, d2, d3 … for div).
Apply text and font properties using external style sheet

3.4 Use selector.html apply class attribute for paragraphs and divisions. All the paragraphs under para_class class and all divisions under div_class class. Apply different CSS font and text styles.

3.5 Create an html file as shown

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

  <h1>Learn CSS 3.0 for better web design</h1>

  <img  src="image1.jpg" title=”title-1”  width="100" height="100“/>

  <img  src="image2.jpg" title=”title-2”  width="150" height="150“/>

  <img  src="image3.jpg" title=”title-3”  width="200" height="200“/>

  <img  src="picture1.jpg"  width="100" height="100“/>

  <img  src="picture2.jpg"  width="150" height="150“/>

  <img  src="picture3.jpg"  width="200" height="200“/>

</body>
</html>
```

Example 4: Selector2.html

Note: Use your own images and pictures of proper size

Create an external CSS file and apply following styles using attribute selector.

a. **img[title] { border: 1px solid #000; }**
b. **img[width] { border: 1px solid #000; }**
c. **img[title|="title"] { border: 1px solid #000; }**

3.6 Create an html file as show below.

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

<h2>CSS Pseudo Classes or Links</h2>
<p>This is a <a href="">link with Pseudo Classes</a> ! </p>

</body>
</html>
```

Example 5: Selector3.html

Apply below css to given html file using pseudo classes and understand how pseudo classes works

```
a:link{
      text-decoration: none;
        color: gray;
}

a:visited{
       text-decoration: none;
        color: gray;
}

a:hover{
       text-decoration: none;
        color: green;
        font-weight: bolder;
        letter-spacing: 2px;
}
```

Example 6: Selector4.css

Lab 4. Layout

Goals	• At the end of this lab session, you will be able to understand: ○ Positioning ○ Box Layout ○ Table Layout ○ Vendor Prefixes ○ Working with Columns
Time	1hr

4.0 Using CSS layout technique design a web page which looks as shown below having 3 divisions, Header, Content and Footer

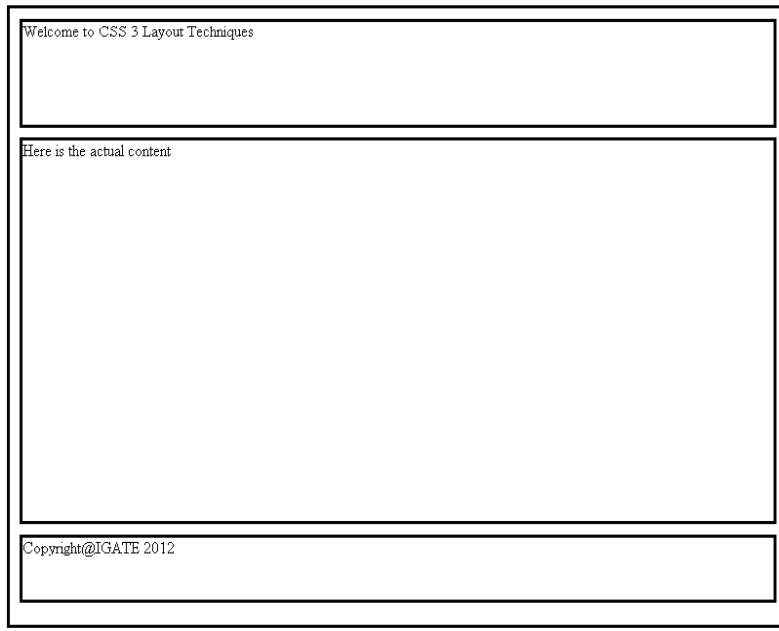


Figure 1: Layout-1

4.1. Refer the page you designed in problem 4.0, make the following changes
a. Set appropriate background and foreground color
b. Change header title to ***'University for Learning"***
c. Align title to center of the division and set appropriate font style

4.2 Refer the layout designed in problem 4.1 then make changes to middle division so that webpage looks as shown below. Set appropriate background and foreground colors to all the divisions

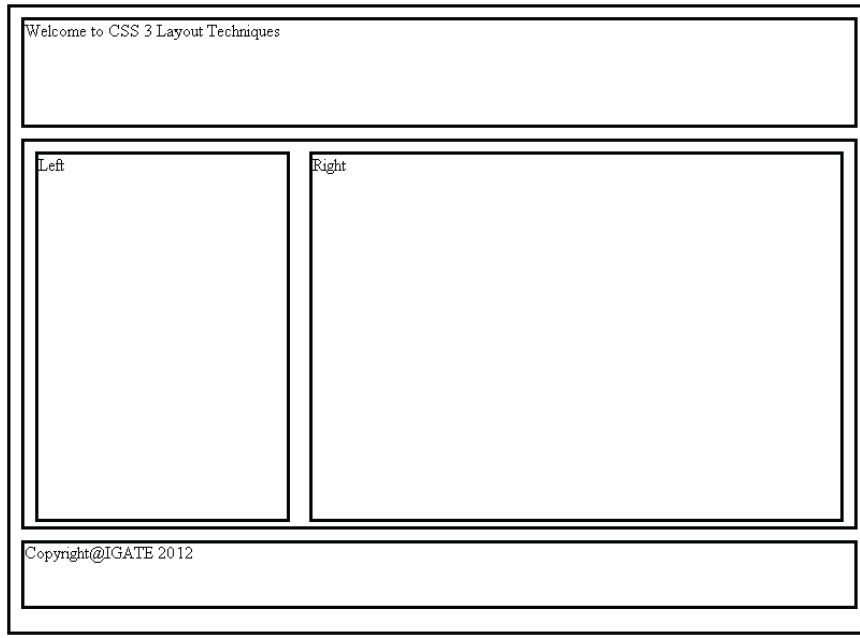


Figure 2: Layout-2

4.3. Refer the layout designed in 4.2, then make the changes so that web page looks as shown below.

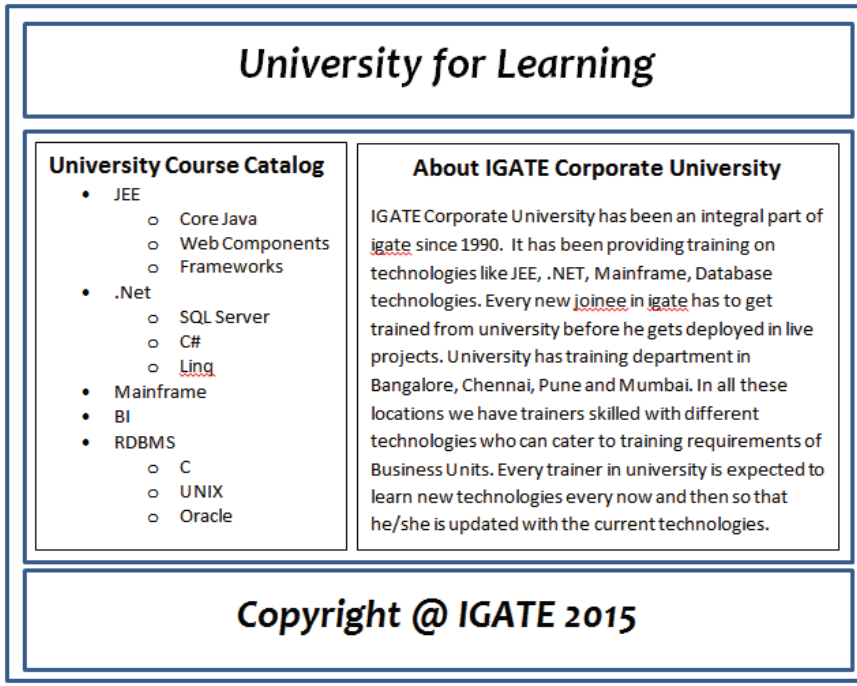


Figure 3: Layout-3

Lab 5. Working with Colors

Goals	• **At the end of this lab session, you will be able to understand:** ○ **Background** ○ **Multiple Image Background** ○ **Applying Colors** ○ **Opacity & Transparency** ○ **Using currentColor** ○ **Working with Gradients**
Time	30 minutes

5.0. Refer problem 4.3's solution and set appropriate background and foreground colors using following mechanism.

- **Hexadecimal colors**
- **RGB colors**
- **RGBA colors**
- **HSL colors**
- **HSLA colors**

5.1. While setting color for a particular division use currentColor attribute.

5.2. Apply CSS 3 gradient function, to background color of header division as follows.

background: repeating-linear-gradient(90deg,rgb(255,0,0),rgb(100,0,0) 20px, rgb(255,0,0) 40px);

Apply same functionality to other divisions with different color combinations.

Lab 6. Borders and Transformation

Goals	• **At the end of this lab session, you will be able to learn how to apply :** ○ **Borders** ○ **Resizable Borders** ○ **Rounded Corners** ○ **Applying Shadows in border** ○ **Transformation 2D and 3D using rotate, skew, scale and translate**
Time	1 hr

6.0 Refer problem 4.3 and make changes so that it looks as shown below,(replace mid-right division by login division)

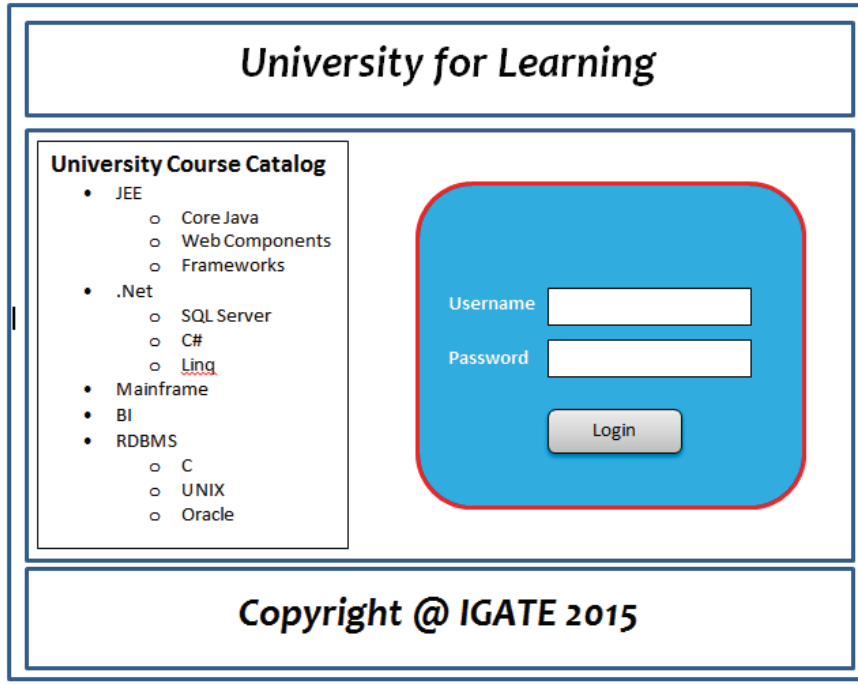


Figure 4: Layout-4

6.1: Refer 6.0 designs; apply the following transformation on login division.

A. Rotate by 30 degree
B. Scale by 2 units
C. Translate it by 20px along y axis and -20 along x axis
D. Apply Skew function with 20deg.
E. Apply all the above with matrix function and rotate it by 20 deg.

Lab 7. Animation

Goals	• At the end of this lab session, you will be able to understand: o CSS 3 Animations o Working with Key frames
Time	30 minutes

7.0 Refer 6.1 designs and include IGATE logo on the right side of header division as shown below.

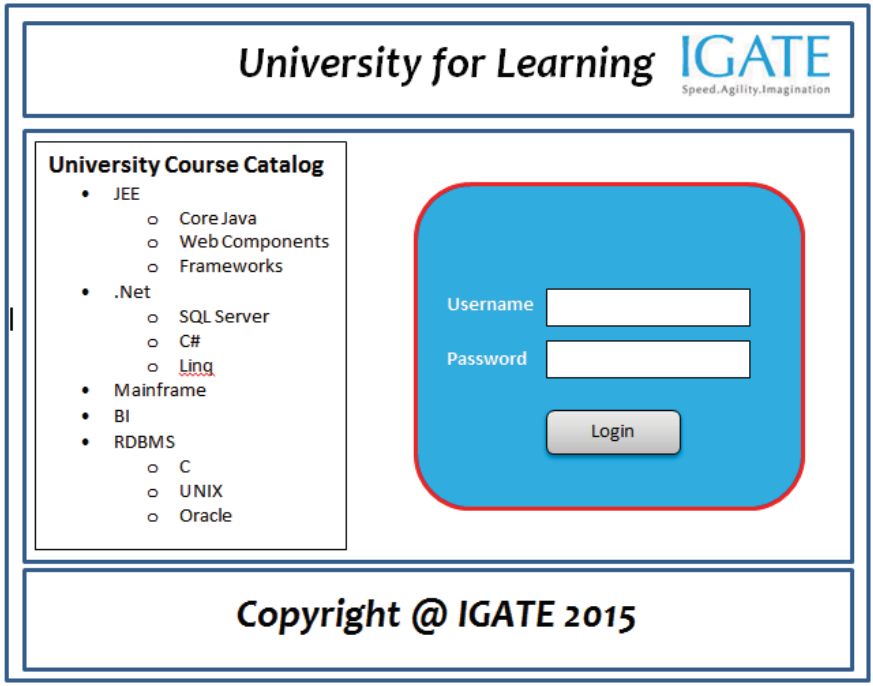


Figure 5: Layout-5

Apply animation to IGATE log and rotate it along y-axis.

Appendix A: Table of Figures

Appendix B: Table of Examples