CSS3 Introduction

Unit: Programming Languages

Problem Area: Incorporating CSS (Cascading Style Sheets)

Lesson: CSS3 Introduction

Student Learning Objectives.

Instruction in this lesson should result in students achieving the following objectives:

1. **Summarize new features of CSS3.**
2. **Create animations using CSS3.**

Resources.

The following resources may be useful in teaching this lesson:

E-unit(s) corresponding to this lesson plan. CAERT, Inc. http://www.mycaert.com.

Castro, Elizabeth, and Bruce Hyslop. *HTML and CSS: Visual Quickstart Guide*, 8th ed. Peachpit, 2014.

"CSS Specifications," *W3C*. Accessed Feb. 13, 2017. https://www.w3.org/Style/CSS/specs.

"CSS3: The Mega Cheat Sheet Infographic," *makeawebsitehub.com*. Accessed Feb. 13, 2017. http://makeawebsitehub.com/css3-mega-cheat-sheet/.

Equipment, Tools, Supplies, and Facilities

- ✓ Overhead or PowerPoint projector
- ✓ Visual(s) from accompanying master(s)
- ✓ Copies of sample test, lab sheet(s), and/or other items designed for duplication
- ✓ Materials listed on duplicated items
- ✓ Computers with printers and Internet access
- ✓ Classroom resource and reference materials

Key Terms. The following terms are presented in this lesson (shown in bold italics):

- @font-face
- animations
- CSS3 module
- Google fonts
- keyframe
- linear gradients
- radial gradients
- rotate function
- scale function
- skew function
- text shadows
- transformations
- transitions
- translate function
- webfont

Interest Approach. Use an interest approach that will prepare the students for the lesson. Teachers often develop approaches for their unique class and student situations. A possible approach is included here.

> *CSS3 introduces new features that can be used to create exciting Web designs. Learning these new features allows you to become a more skilled Web developer professional. Make a list of some of the ways you think this will make you a more valuable employee.*

CONTENT SUMMARY AND TEACHING STRATEGIES

Objective 1: Summarize new features of CSS3.

Anticipated Problem: What are new features of CSS3?

I. New CSS3 features

A. CSS3

1. The introduction of CSS in 1996 was a significant event in the history of Web development. It solidified the idea of separating content from its formatting. As

a result, HTML documents maintained content, and styling/formatting information was provided via CSS styles.

2. Since the introduction of CSS, its specifications have been extended. Currently, CSS2 is widely used. The World Wide Web Consortium (W3C) no longer maintains the CSS2 recommendation and has started work on CSS3 specifications. CSS3 specifications are not maintained in a single specification. Instead, the specification is maintained in a number of separate modules. A ***CSS3 module*** is a tool that extends CSS2 features or adds new features, and individual modules are at different levels of completion and stability. Some modules are referred to as level 4 modules and will eventually become part of CSS4. [See VM–A.]

B. CSS3 browser compliance

1. CSS3 specifications have evolved over time. Browser software vendors have kept up with these changes and provide support for new enhancements. However, there is a time lag between when a CSS3 module's development is started and when it is finalized. In the interim, when an enhancement is not stable, vendors may choose to implement the feature to provide its users with access to the feature. [See VM–B.]
   a. Code beneath the VM–B table shows an example of using vendor prefixes for the <p> element. The actual meaning of the code is explained later on in this lesson. For now, the focus is on the notation used to specify browser implementations.
   b. Lines highlighted in yellow on the VM–B table show the CSS property prefixed with the browser prefixes. The line highlighted in green shows the property specified without any vendor prefix.
2. As CSS3 support grows in browsers, properties can be set without vendor prefixes. However, as new enhancements are developed in CSS technology—and browsers provide support before the enhancement specification is finalized—developers need to use vendor prefixes for them.

C. CSS3 font enhancements

1. CSS has always allowed developers to specify font using the font-family property. Two types of font-family names are generic and specific.
   a. The ones built into the browser software are generic family names and have the values "serif," "sans-serif," "cursive," "fantasy," and "monospace." When a generic family name is specified, fonts installed on the client machine that match the family are used.
   b. Specific font-family values can be specified, such as "Times" or "Arial." However, if a specific family name is specified and the font is not found on the client machine, it cannot be used.
2. CSS3 allows developers to use webfonts. A ***webfont*** is a typeface installed on a Web browser that is often designed for screen use. It is treated like an image or CSS file. The font file associated with a webfont is referenced using a URL in the HTML document. When a user navigates to an HTML document, a webfont is retrieved from the URL instead of using a font from the client machine. The

URL may point to the font file on another server, or it may reside on the same site as the HTML document that uses it.

a. Where are webfonts found? Some are free, and some of them need to be purchased before they can be used. Webfonts can be used from another server, or the file can be downloaded and saved in the same location as the HTML document. [See VM–C and VM–D.]

b. ***Google fonts*** are free webfonts that may be used in a webpage. These fonts cannot be downloaded to the same location as the HTML document and must be connected using a URL. [See VM–E. Navigate to https://fonts.google.com/ to locate a font you wish to use.]

   (1) See VM–F. The display that pops up when the block box for a font is clicked is shown. It displays the two methods to use this font. The first method uses a <link> element, and the second method uses the @import statement.

   (2) See VM–G. It shows two HTML documents. Example 1 uses the <link> element shown in the Google site. A style is created as per the instructions at Google. The second example uses the @import statement to link to the Google font.

   (3) Both examples produce the same output. The <link> element method is faster than using the @import statement.

c. Downloading a webfont: Font files are provided in multiple formats. Fonts used on a desktop machine are maintained in files with the ".ttf" extension. Webfonts are maintained in files with the ".otf" or "woff" extension. Webfonts may be downloaded from a website. For this lesson, the author used https://www.fontsquirrel.com/. Some of the fonts at this site are used for desktop display or print publishing. REMINDER: A developer must select a font that provides a webfont version.

   (1) See VM–H. See examples of fonts with and without webfont files. The first font has no link indicating that it provides webfont files. The second font, "1942 REPORT," (https://www.fontsquirrel.com/fonts/1942-report) contains a link called "Webfont Kit," indicating that it contains the files for the webfont version of the file.

   (2) See VM–I. It shows the screen displayed when the link to download webfonts is clicked. Clicking on the link downloads a compressed file that contains a folder called "webfonts." Inside the folder structure is a file called "1942-webfont.woff." This file contains the webfont and can be used in an HTML document. The folder structure also contains a demo HTML file that shows code to make use of the font. The "1942-webfont.woff" file is placed in the same folder as the HTML document that will use it.

d. See VM–J. It shows code to use the downloaded webfont file in a webpage. Font information is provided using the @font-face element. ***@font-face*** is an element that allows authors to specify online fonts to display text online. Code highlighted in yellow provides two pieces of information: the name of the font and the location of the file. Since this author placed the ".woff" file

in the same folder as the HTML document, no path information is provided. If the font file is in a different location, relative path locations may be provided.

(1) The code highlighted in orange shows the usage of the webfont for a <p> element. The name of the font is the name specified in the @font-face statement and is not the one maintained in the FONTsquirrel website.

(2) Using webfonts increases the variety of fonts that may be used on a webpage.

D. CSS3 background enhancements

1. Using the new features of CSS3, block elements can be created with a gradient background. [See VM–K. It shows the two types of gradients that can be used, along with their attributes.]

   a. ***Linear gradients*** are gradients set up with a minimum of two colors and a direction. The direction specification can be a text constant or a number indicating the degree of the gradient. The browser handles the blending of the colors in the gradient. [See VM–L. It shows examples of linear gradients using text constants highlighted in yellow. The third example specifies the first argument as a number followed by "deg" to denote the degree of the gradient. Only relevant portions of the code are shown. The rendering of the code in a browser is also shown. See VM–M. It lists the valid values for the first argument to the linear-gradient statement, along with the degree values for them. Therefore, the developer may use "to right" or "90deg" for the same blending pattern.]

   b. ***Radial gradients*** are gradients that allow colors to be blended along a circle or an ellipse. [See VM–N for the four parameters that can be specified for a radial gradient: shape, size, position, and color.] The first parameter specifies the gradient shape, which can be circle or ellipse. The shape may be omitted since it can be inferred by the value for the second parameter that specifies the gradient size. The second parameter specifies the size of the shape. If it is omitted, the shape is determined by the system based upon the dimensions of the block element to which it is applied. If one number is specified for the size parameter, the shape is assumed to be a circle. If two numbers are specified for size, it is assumed to be an ellipse. The third parameter indicates the position of the circle or ellipse. Lastly, the gradient colors are specified.

      (1) See VM–O for examples of radial gradients. Only relevant portions of the file are shown. In the first example, all four parameters are specified: shape, size, position, and colors. In the second example, only the colors are specified. Default values are used for the first three parameters. The browser automatically determines the default value for size. The default value of position "at center" is used. This centers the shape in the block within the element. In the third example, the shape and size parameters are omitted, and the position parameter is specified along with the color set.

(2) Radial backgrounds can be applied to an entire webpage by specifying the background property for the <body> element.

E. CSS3 border enhancements

1. Block elements can be placed on a webpage with colored and gradient backgrounds and are rendered on a webpage as rectangles. By using CSS3 features, developers can render block elements with rounded edges.
   a. See VM–P. To render a rounded corner, the radius of the circle that will create the curved border is specified. The larger the radius of the circle, the greater the roundedness of the edge.
   b. See VM–Q for examples of elements with rounded edges. In the first example, the radius is set to 100px. The text-align property of text was set to center in order to center text within the div. Without this property setting, text will appear at the upper left side of the element and not honor the rounded corners. In the second example, the radius is specified as a percent. If the height and width of an element are equal, setting the border-radius to 50 percent will render a perfect circle. In the third example, two numeric values are specified for the border-radius property. When two values are provided, the first numeric value is used to set up the border radius of the top-left and bottom-right corners. The second number is used to set up the radius for the top-right and bottom-left edges.
2. Border radius can be set for each of the four corners of an element using the border-top-left-radius, border-top-right-radius, border-bottom-right-radius, and border-bottom-left-radius properties.

F. ***Text shadows*** are the parameters that must be specified to create a shadow effect. CSS3 can be used to render shadows for text content and elements. [See VM–R to visualize the text shadow parameters.] The x-offset, y-offset, and blur-radius are specified in length units (px or em). Multiple shadows may be applied to the same text.

1. Shadows for text content are created using the text-shadow property. Multiple shadows can be applied to a single set of text content. See VM–S for examples of creating text shadows.
   a. In the first example, all four parameters are specified.
   b. In the second example, two shadows are created for the shadow2 class. The first is red in color, and the second is orange in color. They are blended to provide a composite shadow.
   c. In the third example, only two numeric values are provided. These provide the x-offset and the y-offset for the shadow effect. The blur parameter is omitted, and the shadow is rendered without any blurring.
2. Shadows can be applied to elements, such as the element, using the box-shadow property. This property also requires that the x-offset, y-offset, blur-radius, and shadow color be specified.
3. Shadows for a container element, such as the element, can be applied in conjunction with rounded borders. When the borders of an element are

rounded, the shadow is also created with rounded corners. See VM–T for examples of setting shadows for container elements.

a. In the first example, the element is set up to have rounded corners using the border-radius property. It has a box-border property setting. This is shown in the browser view at the right of the code. The element holds a <p> element that provides a shadow for its content.
b. In the second example, the element is rendered with a shadow. The <p> element contained inside the element does not have a text-shadow property setting, and this leads to the text in the <p> element getting rendered without a property. This also illustrates the fact that the shadow property is not an inherited property.

Teaching Strategy: *Many techniques can be used to help students master this objective. Use VM–A through VM–T to summarize the new features of CSS3. Assign LS–A.*

Objective 2: Create animations using CSS3.

Anticipated Problem: How is CSS3 used to create animations?

II. CSS3 animation-related features

A. Transformations

1. ***Transformations*** are operations that change an element's appearance and the position of the element on the page. CSS3 provides powerful new features to transform the appearance of elements in a webpage. Many types of transformation can be affected using CSS3. [See VM–U. It shows the various types of transformations that are allowed in CSS3. These transformations are applied to block elements.] When an element is transformed, it changes its appearance and position on the page. However, other elements on the page are not correspondingly transformed to maintain pre-transformation design. This may lead to the overlap of elements.
   a. The ***rotate function*** is a feature that repositions an element on a page. Rotation is provided as a degree value. Positive degree values cause clockwise rotation, and negative values lead to counter-clockwise transformations. [See VM–V. The code shows three examples of the transform property and the rotate function.] The first element class definition sets up a element class that has no transformation. The element is rendered horizontally, with 0 rotation. The second example shows a element with a rotation of 45 degrees. Note the use of "deg" after the number that specifies the rotation angle. The third example shows a element that has been rotated by –90 degrees, causing it to be rotated counter-clockwise. This causes the element to overlap the previous element.

b. The ***scale function*** is a feature used in the transform property to modify the size of an element by changing the element's height and/or width. [See VM–W.] The scale function may be invoked with one numeric argument or with two numeric arguments. When only one argument is specified, it is used as the scaling factor to modify the height and width. When two arguments are specified, the first number is used as the horizontal/width scaling factor, and the second argument is used to scale its height/ vertically. [See VM–X. It shows scale function examples.]
   (1) The first example is a control case where no scaling factor has been applied to the element.
   (2) In the second example, the scale function is used with one argument used as the scaling factor for height and width. This causes the element to shrink by 50 percent, and this scaling is applied to the <p> element inside the element.
   (3) In the third example, the scale function is invoked with two arguments, where the first argument scales the width by 0.75, making the narrower to 75 percent of its original width. The second argument scales it vertically and makes it taller by a factor of two. The text inside the element appears narrower and taller.
   (4) The scale function can be invoked with a negative number, such as –1. This causes the element to be mirrored horizontally and/or vertically.

c. The ***translate function*** is a feature that moves an element from its original position to a new position within the page. Movement is specified in length units vertically and horizontally.
   (1) See VM–Y. There are three ways for translate functions to be used in a page. In the first format, it is invoked with two arguments, where the first argument denotes the amount to be moved horizontally, and the second argument denotes the amount to move vertically. However, if movement is to be accomplished only along one axis, the translateX or translate functions can be used.
   (2) See VM–Z. Three classes are set up. The first one has no translation property settings. The second element uses the translate function with two arguments, causing it to move vertically and horizontally. The third example uses the translateX function to move an element horizontally.

d. The ***skew function*** is a feature that slants elements along its x- and y-axis, providing a 3D-feel to the transformation. [See VM–AA. The skew functions are similar to the translate functions.] The skew function may be invoked with two arguments, indicating the degree to which the element must be skewed along the x- and y-axis. Alternatively, specialized functions may be invoked to skew along only one axis.
   (1) See VM–BB for examples of the skew function. The first example serves as a test case with no skew transformations. The second

example skews along the x- and y-axis, while the third example skews horizontally.

(2) See VM–CC. It is possible to apply multiple transformations to an element. The highlighted line shows two transform actions that are part of the style. Any two-transform specifications are separated by a space. When the style is applied, both operations are performed.

B. CSS3 transitions

1. ***Transitions*** are a mechanism for an element's style to morph from a starting style to an ending style. The beginning and ending styles must be defined before applying the transition. Based on transition settings, the actual mode of transition is implemented by the browser.
2. CSS3 transitions are implemented using four properties: the end value of the property to be modified, duration, timing, and any delay. [See VM–DD.]
   a. Step 1–Property: It is necessary to select a property that is to be transitioned, such as the background color. The transition will modify the current value of the property to the one specified in the transition.
   b. Step 2–Duration, timing, and delay: It is important to specify the duration of the transition in seconds or milliseconds. The timing property can be specified using pre-defined keywords or by setting up a custom timing function. The delay setting is also a time-related setting and can be specified as a single number or as a set of comma-separated values.
3. See VM–EE. It shows a simple transition, in which the color of a element changes when the mouse hovers above it, as specified by the hover selector. The style for the hover selector shows that the background color must be changed to gray when the mouse hovers over the element.
   a. It is essential to consider the line highlighted in coral. It depicts a transition property that specifies the background color should change over a period of 5 seconds.
   b. Next, it is necessary to consider the line highlighted in yellow. This states that the color of the element should become gray over a period of 5 seconds. This action is executed when the mouse moves away from the element, and the color transitions from gray back to beige.
4. See VM–FF. It shows the application of multiple transitions across a single element. In the example, a box shadow is applied to the element, along with a background color change on a hover. The box shadow is erased when the mouse moves away from the element, along with reverting to its background color.

C. CSS3 animations

1. ***Animations*** are CSS3 techniques that provide interactivity in a webpage. Transitions are triggered by actions (e.g., a hover action) and allow developers to change properties based upon actions. Animations can be used to produce the same effect but allow for greater control over the process. For example, animations can be created in a page that loops continuously without user interaction.

2. Another difference between transitions and animations is that transitions require only definitions for the start and end phases. Then the system decides on the process of moving from the start to the end phase. In an animation, the user has control over the intermediate phases between the start and end phases.
3. The different phases of an animation are specified in keyframes. For example, an animation may have three keyframes. Each ***keyframe*** is a tool that contains property specifications. Animation is affected by cycling though the property settings maintained in each keyframe. Implementing only two keyframes results in an animation that works like a transition.
4. See VM–GG. It shows the various properties that can be set for an animation. An animation, identified by a name, can be played out over duration, with or without a delay. It may be performed once or multiple times.
5. See VM–HH. The code highlighted in yellow sets up an animation, which is composed of keyframes, via @keyframes selector. Inside this selector, a collection of keyframes is defined at various time intervals. When this selector is used, all the keyframes inside it are played, providing an impression of movement.
    a. VM–HH also shows a keyframes selector called "example." It contains three keyframes that alter the width of an element. The first frame sets the width to 100px; the second keyframe sets it to 50px; and the third keyframe sets it back to 100px.
    b. This animation is used across the hover selector for a element. The name of the keyframes selector is specified, along with the time that the animation should be played, which is 500 milliseconds in this case. The animation should continue indefinitely and will cease only when the mouse moves away from the element.
    c. Playing this animation causes the element to contract and expand since the keyframes alter the width property of the element to which the animation is applied.
6. See VM–II. It contains a keyframes selector called opacity, and it contains six keyframes. Each keyframe sets the opacity, width, and padding properties of an element. When this animation is applied to an element, such as a element, it appears animated when the mouse hovers above it. In addition, it appears to shrink and fade, and then it enlarges and regains opacity. The animation runs constantly since the iteration count is set to infinite number of times.
7. Learning to apply transitions and animations helps create websites that implement interactivity using CSS.

Teaching Strategy: *Many techniques can be used to help students master this objective. Use VM–U through VM–II. Assign LS–B.*

- **Review/Summary.** Use the student learning objectives to summarize the lesson. Have students explain the content associated with each objective. Student responses can be used in determining which objectives need to be reviewed or taught from a different angle. If a textbook is being used, questions at the ends of chapters may be included in the Review/Summary.

- **Application.** Use the included visual master(s) and lab sheet(s) to apply the information presented in the lesson.

- **Evaluation.** Evaluation should focus on student achievement of the objectives for the lesson. Various techniques can be used, such as student performance on the application activities. A sample written test is provided.

- **Answers to Sample Test:**

Part One: Completion

1. webfont
2. linear gradients
3. blur-radius
4. translate
5. scale function
6. @keyframes

Part Two: True/False

1. T
2. T
3. F
4. F
5. F
6. T

Part Three: Short Answer

1. -moz-border-radius:5px;
2. <link href="https://fonts.googleapis.com/css?family=Indie+Flower" rel="stylesheet">
3. <style>
   p{

   font-family: 'Indie Flower';
   }
   </style>

Sample Test

Name _____

CSS3 Introduction

Part One: Completion

Instructions: **Provide the word or words to complete the following statements.**

1. A typeface installed on a Web browser often designed for screen use is a/an _____.
2. Gradients set up with a minimum of two colors and a direction are _____.
3. The setting for a box shadow that determines whether the edges of the shadow are sharp or not are _____.
4. The feature that allows an element to be moved in a webpage is _____.
5. The keyword that refers to a transform function that modifies the size of an element is a/an _____
6. The selector used to define a CSS3 animation is _____.

Part Two: True/False

Instructions: **Write *T* for true or *F* for false.**

_____1. Elements can have gradient backgrounds.
_____2. A element may have multiple shadows defined for it.
_____3. A selector is allowed to have only one transform operation defined for it.
_____4. Transition duration is specified in seconds only.
_____5. Animation is affected by cycling though the property settings maintained in each keyframe.
_____6. A transition is similar to an animation with only two keyframes.

Part Three: Short Answer

Instructions: **Answer the following.**

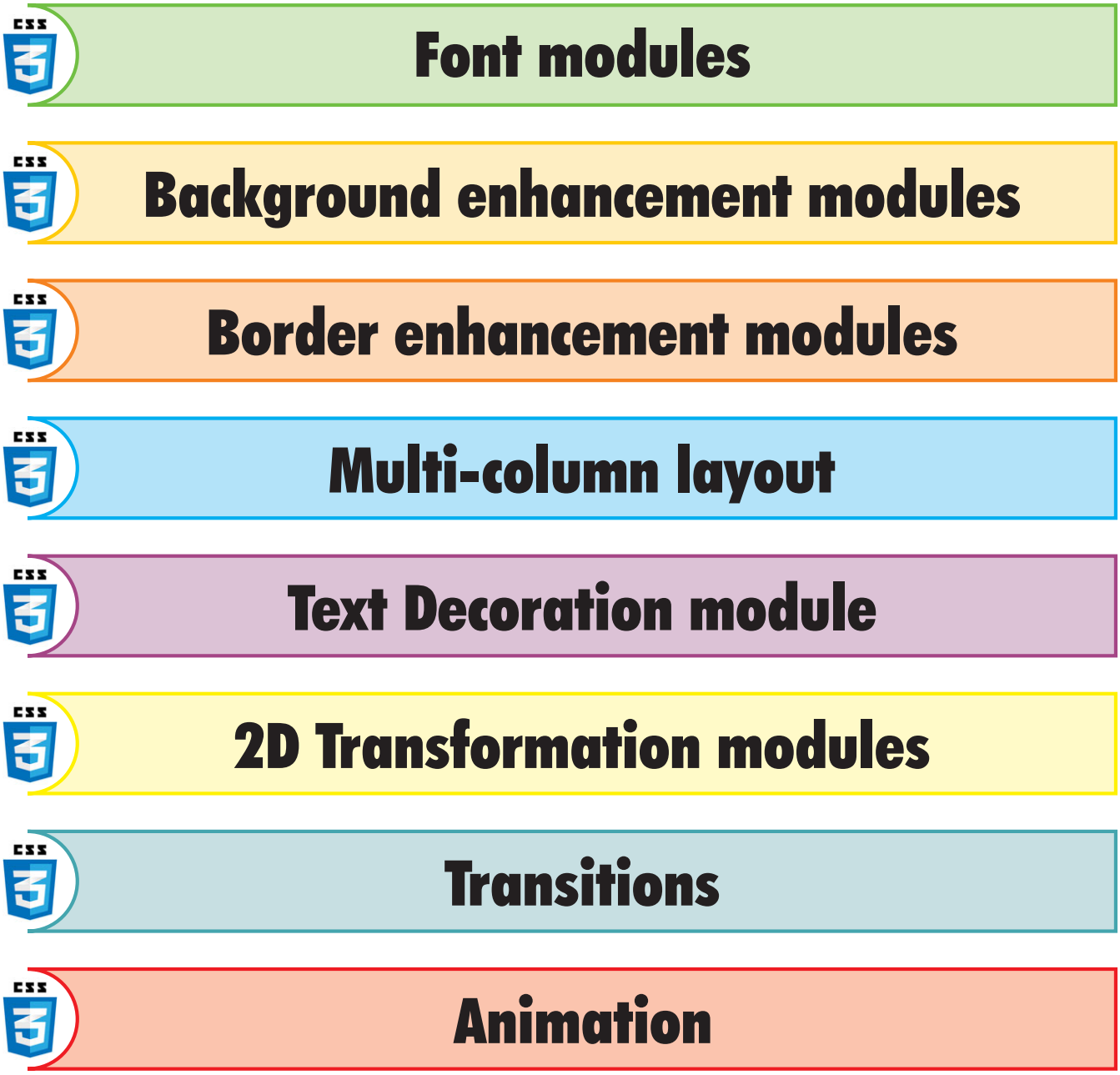
1. Write the statements to set up a rounded border with a 5px radius using the browser prefix for Firefox.

2. Suppose that there is a Googlefont with this URL: https://fonts.googleapis.com/css?family=Indie+Flower. The Googlefont site indicates that this font is defined as the font-family "Indie Flower." Code the link statement to use this Googflefont.

3. Use the font defined in Question 2 in the <p> class selector.

VM–A

IMPORTANT CSS3 MODULES



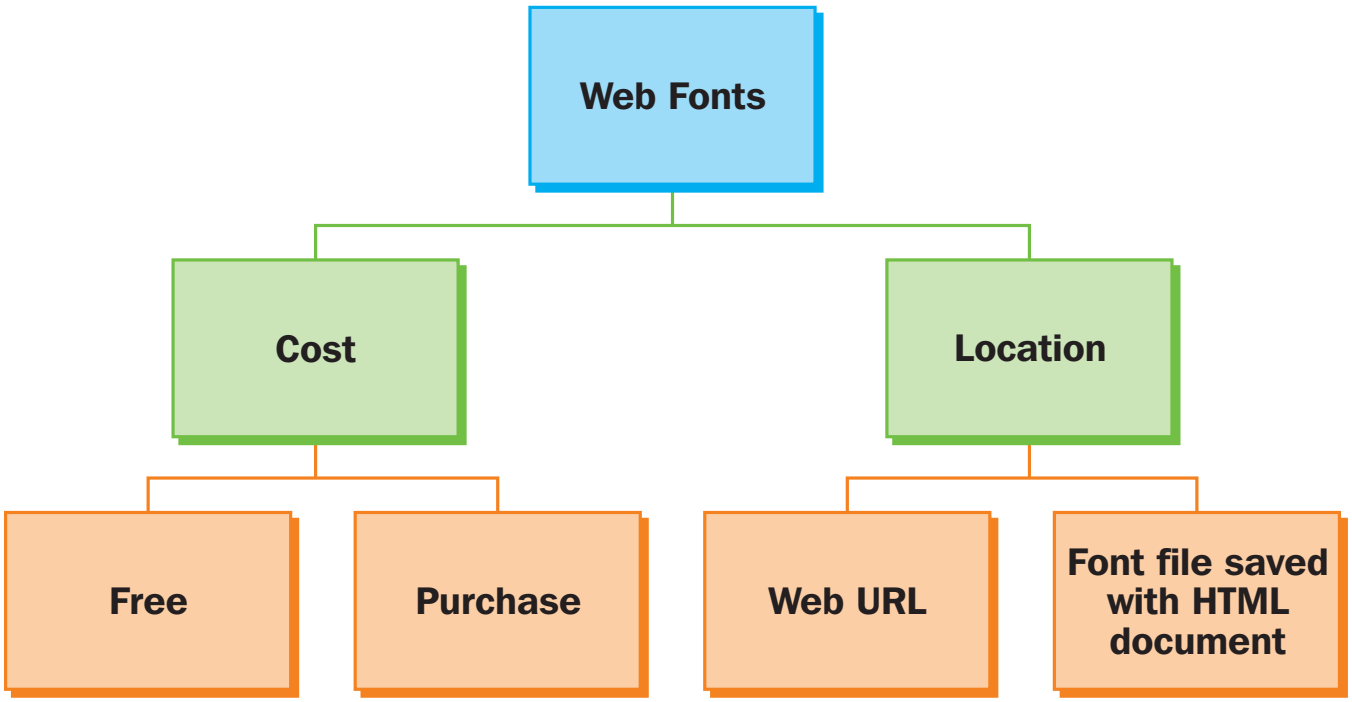
VENDOR PREFIXES

Browser Vendor	Prefix
Webkit, Safari, older versions of Chrome	-webkit-
Firefox	-moz-
Interner Explorer	-ms-
Opera	-o-

Example: Using a Vendor Prefix

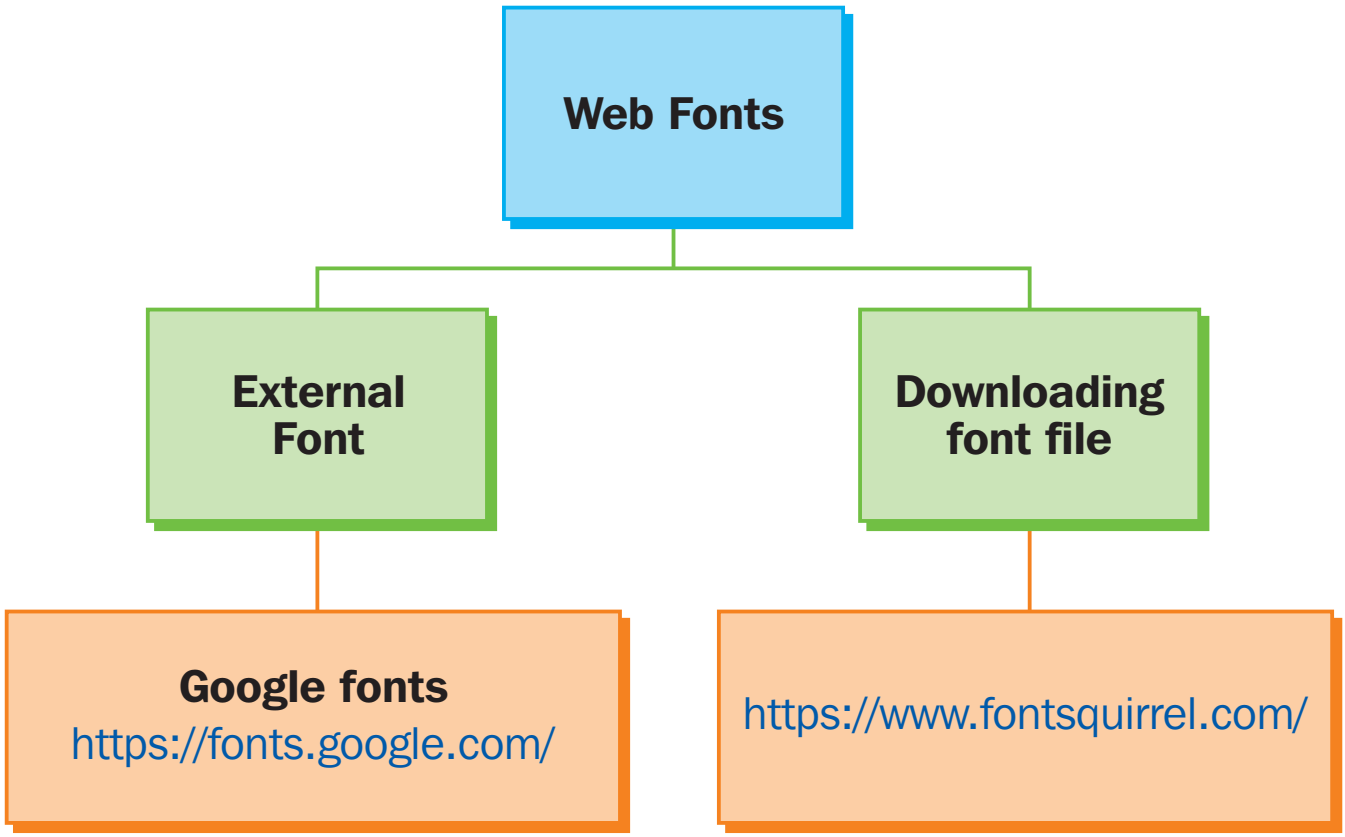
```
<!DOCTYPE html>
<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
    <style>
        p{
        -moz-border-radius: 10px;
        -webkit-border-radius: 10px;
        -o-border-radius: 10px;
        border-radius: 10px;
        border-style: solid;
        border-color: green;
        }
    </style>
</head>
<body>
    <p>This is a test paragraph,
        followed by a list</p>
</body>
</html>
```

WEBFONT DETAILS



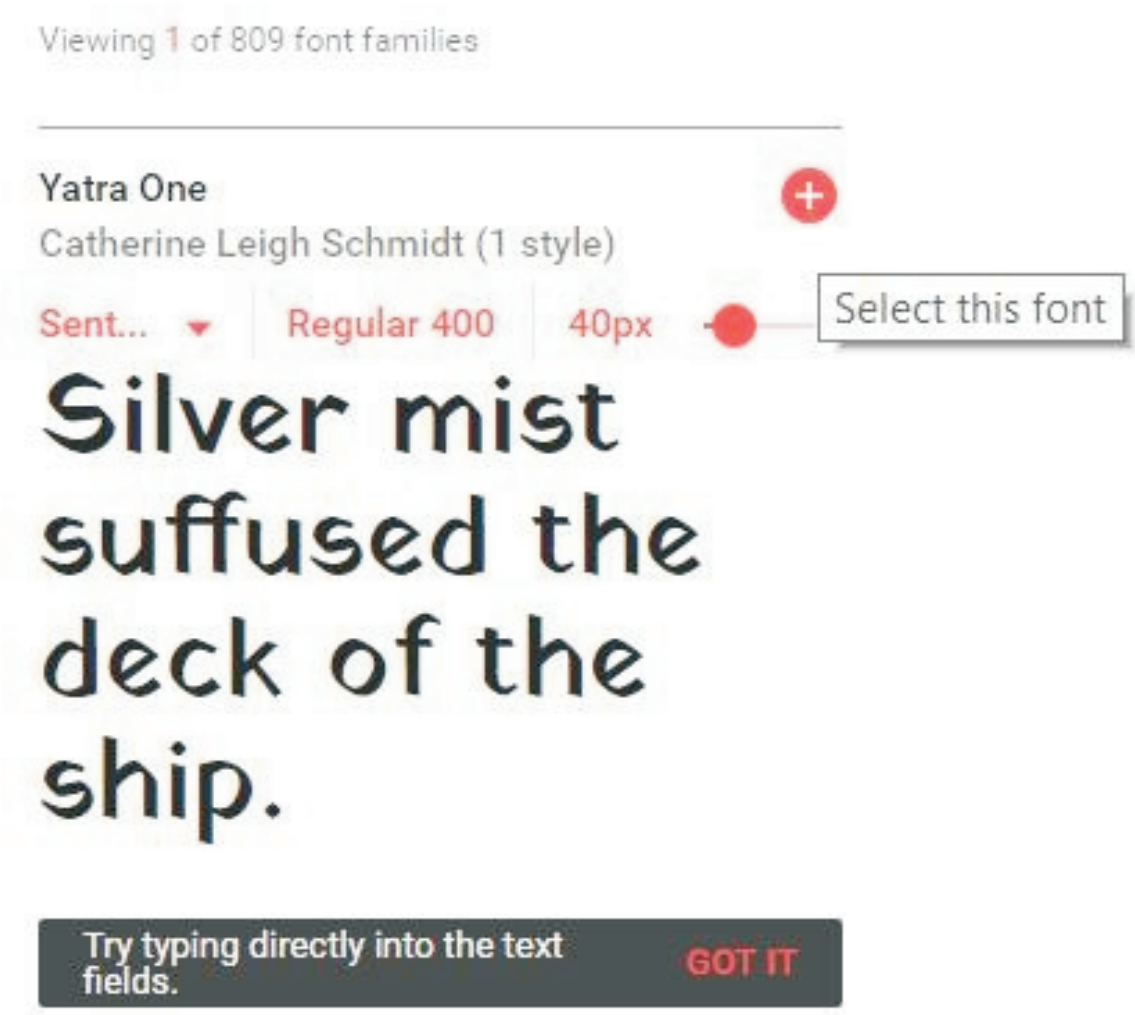
VM–D

FINDING A FREE WEBFONT

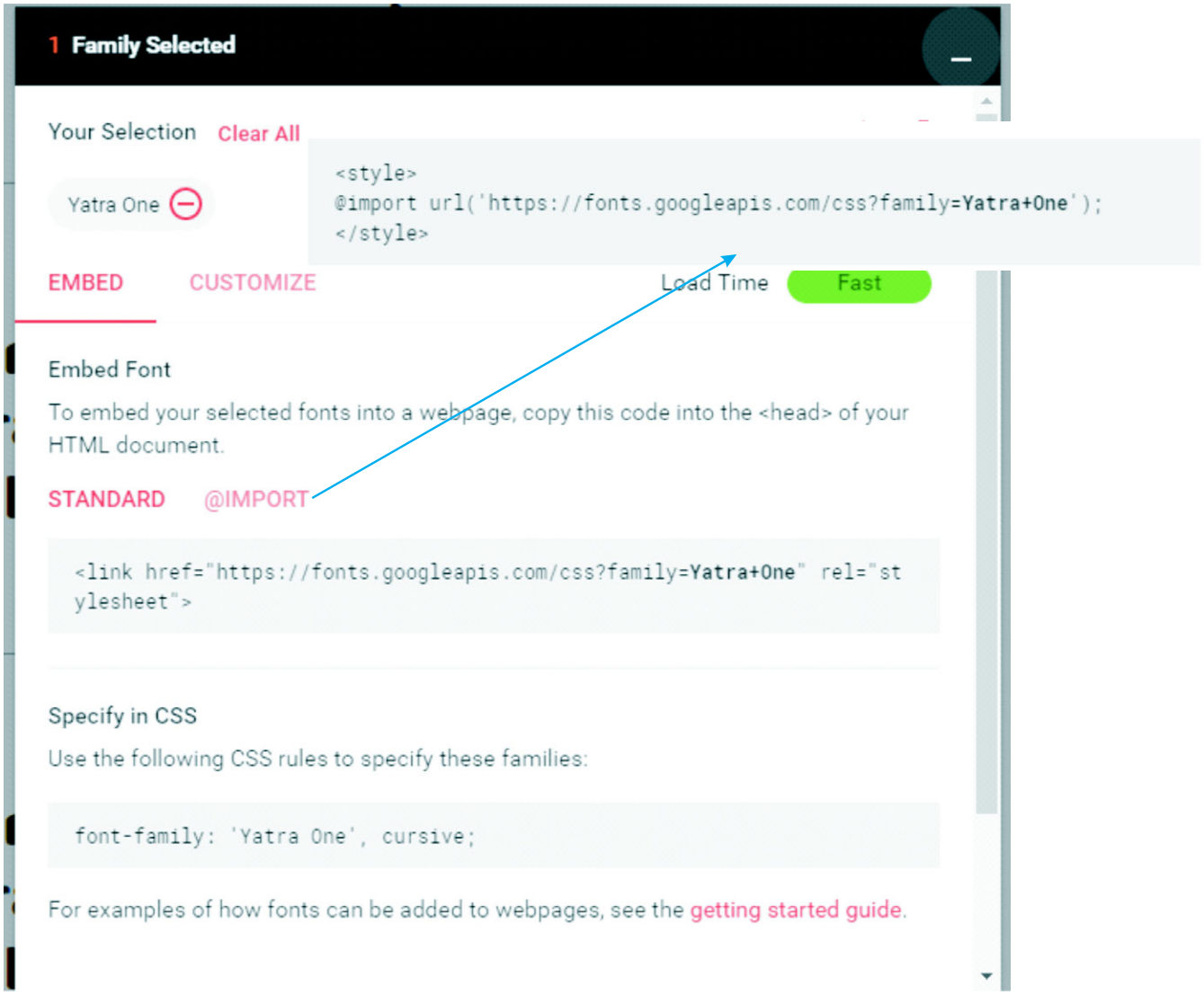


VM–E

WORKING WITH A GOOGLE FONT



USING A GOOGLE FONT: 1



USING A GOOGLE FONT: 2 EXAMPLES

Example 1

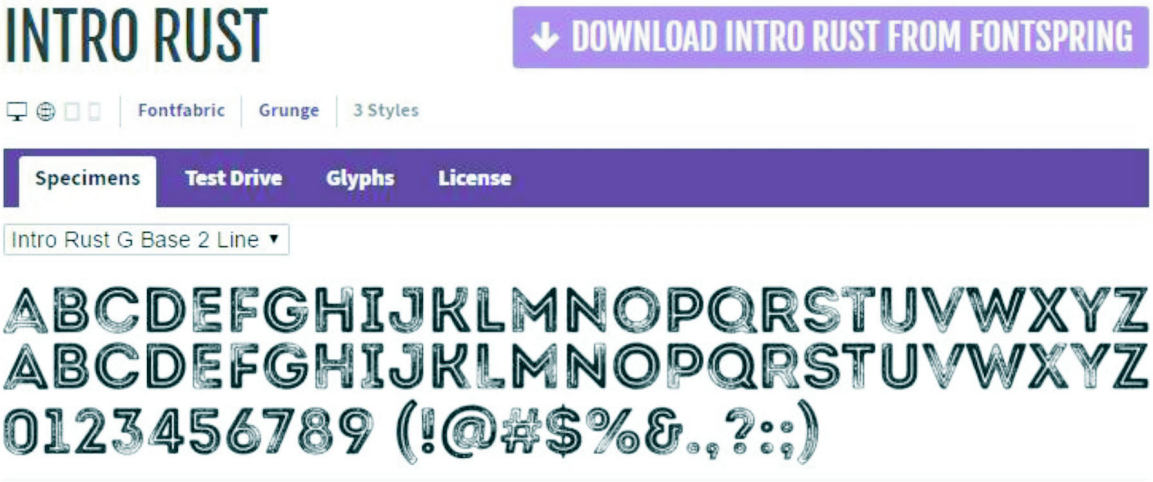
```
<!DOCTYPE html>
<head>
    <meta charset = "utf-8" />
    <title>Myfirst HTML5 page</title>
    <link href="https://fonts.googleapis.com/css?family=Yatra+One"
        rel="stylesheet">
    </link>
    <style>
        p{
            font-family: 'Yatra One', cursive;
        }
    </style>
</head>
<body>
    <p>This is a test paragraph. It tests
    google fonts</p>
</body>
</html>
```

Example 2

```
<!DOCTYPE html>
<head>
    <meta charset = "utf-8" />
    <title>Myfirst HTML5 page</title>
    <style>
        @import 'https://fonts.googleapis.com/css?family=Yatra+One';
    </style>
    <style>
        p{
            font-family: 'Yatra One', cursive;
        }
    </style>
</head>
<body>
    <p>This is a test paragraph. It tests
    google fonts</p>
</body>
</html>
```

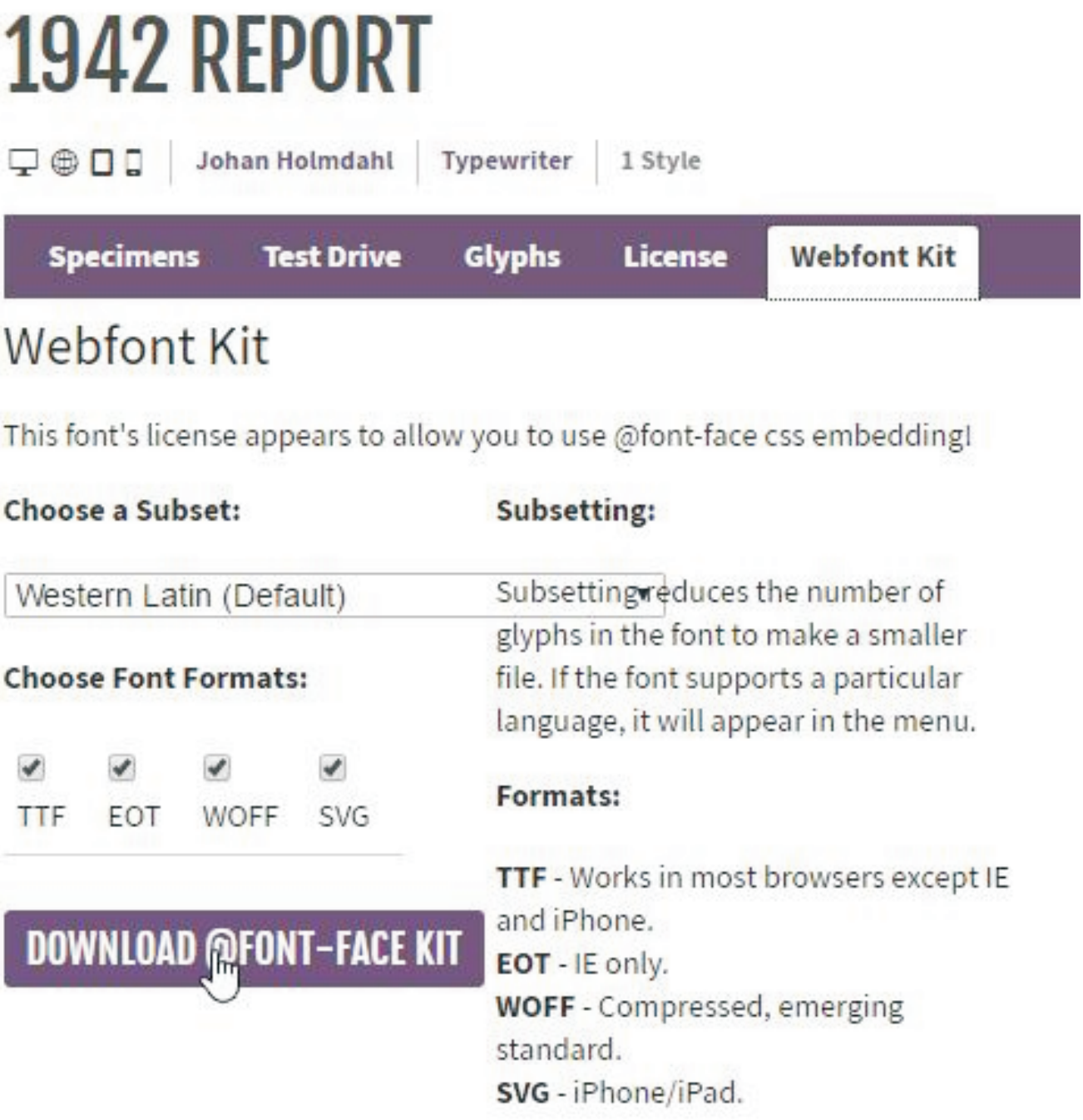
USING A WEBFONT FROM FONTSQUIRREL

Font Without Webfont Files



Font With Webfont Files

DOWNLOADING A WEBFONT FROM FONTSQUIRREL



USING A DOWNLOADED WEBFONT

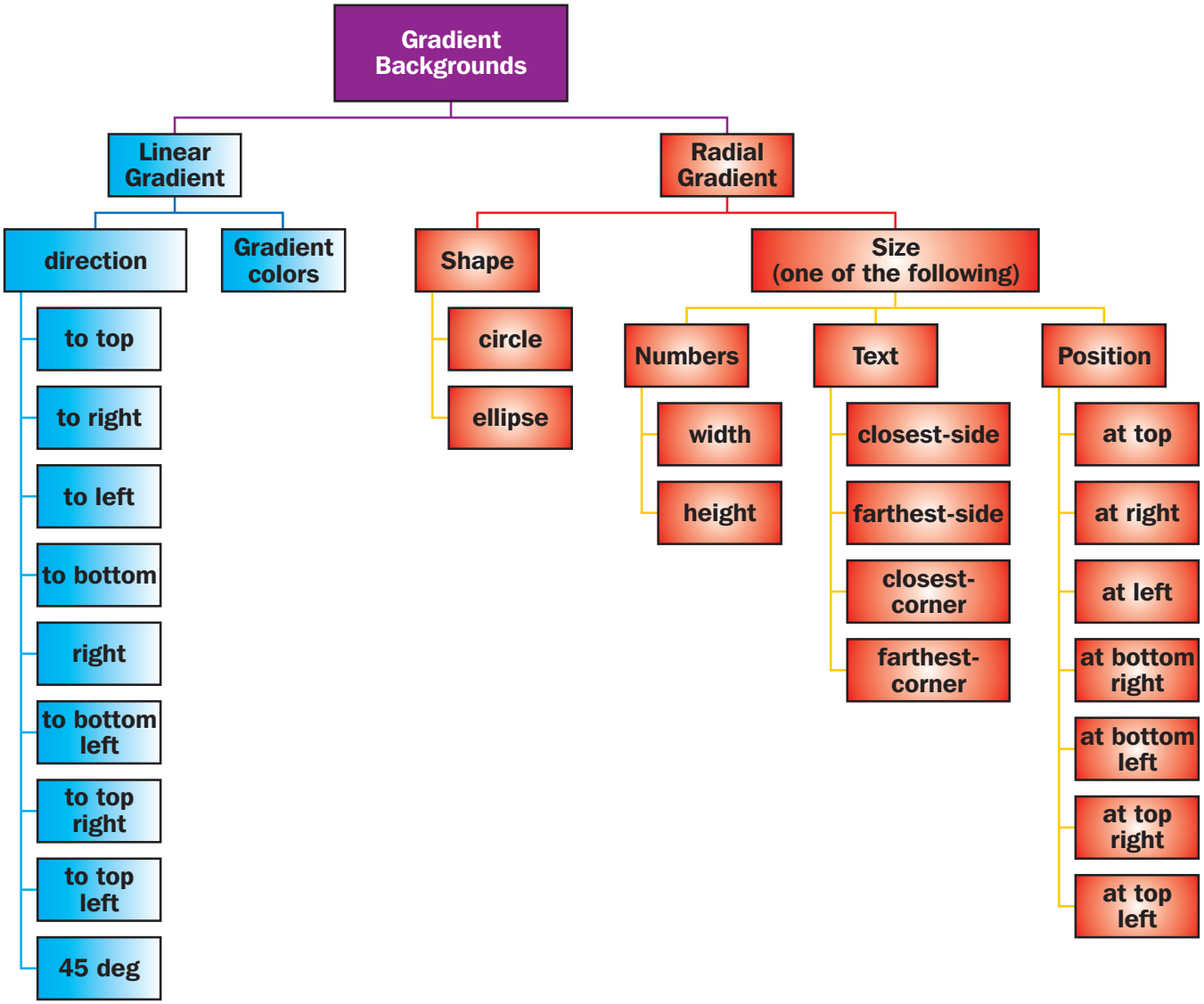
HTML Code

```
<!DOCTYPE html>
<head>
    <meta charset="utf-8" />
    <title>Myfirst HTML5 page</title>

    <style>
    @font-face {
      font-family: 'My_web_font';
      src: url('1942-webfont.woff');
    }

    p{
       font-family: 'My_web_font', 'Georgia', serif;
       font-size:large;
    }
    </style>
</head>
<body>
    <p>This is a test paragraph, followed by a list.</p>
</body>
</html>
```

Browser View

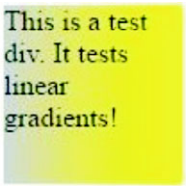
This is a test paragraph, followed by a list

CSS3 GRADIENTS

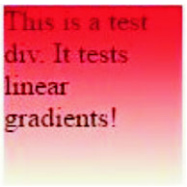


EXAMPLES OF CSS3 LINEAR GRADIENTS

Code	Browser

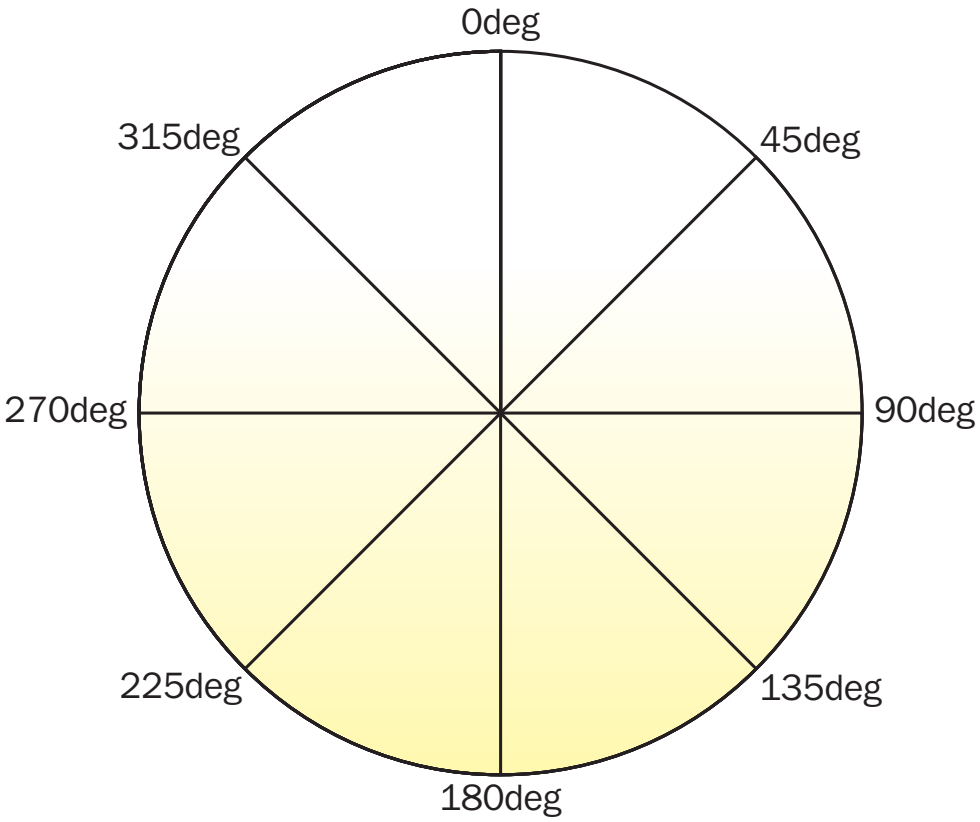
...
<style>
div.linear_grad_1{
background:linear-gradient(to right, silver, yellow);
height:100px;
width:100px;
}
</style>
...
 	
 This is a test div. It tests linear gradients!
 	
	

...
<style>
div.linear_grad_2{
background:linear-gradient(to top, yellow, lightblue);
height:100px;
width:100px;
}
</style>
</head>
<body>
 	
 This is a test div. It tests linear gradients!
 	
</body>
</html>
	

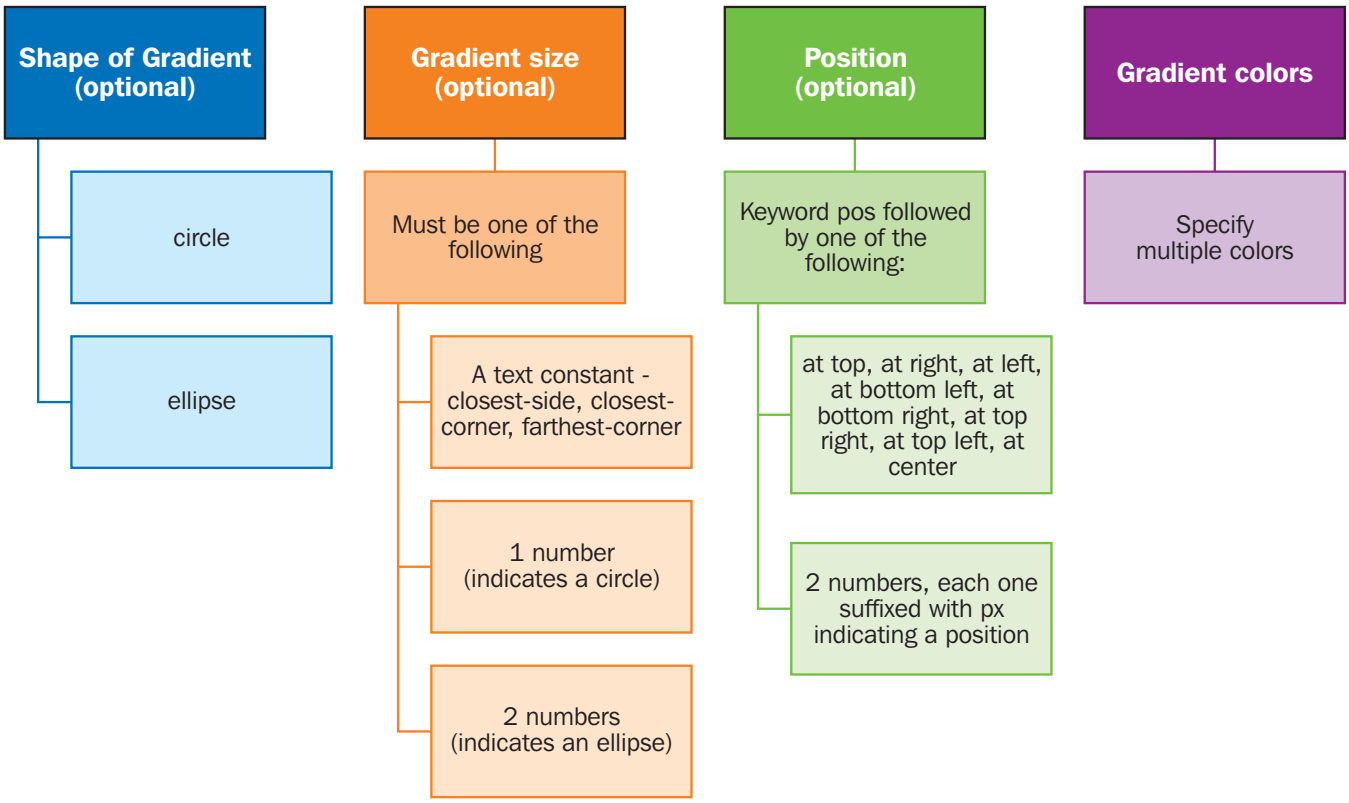
...
<style>
div.linear_grad_3{
background:linear-gradient(180deg, red, beige);
height:100px;
width:100px;
}
</style>
</head>
<body>
 	
 This is a test div. It tests linear gradients!
 	
</body>
</html>
	

CSS3 LINEAR GRADIENT DIRECTION VALUES

Direction as a Text Literal	Corresponding Degree
to top	0deg
to right	90deg
to bottom	180deg
to left	270deg
to bottom right	135deg
to bottom left	225deg
to top right	45deg
to top right	315deg

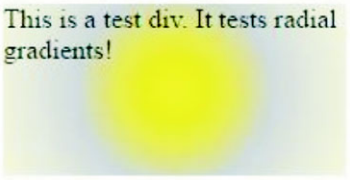


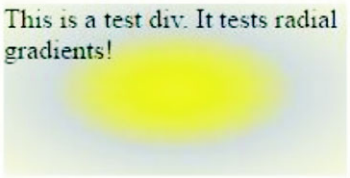
CSS3 RADIAL GRADIENT DETAILS

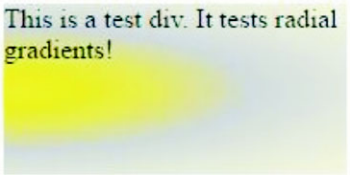


EXAMPLES OF CSS3 RADIAL GRADIENTS

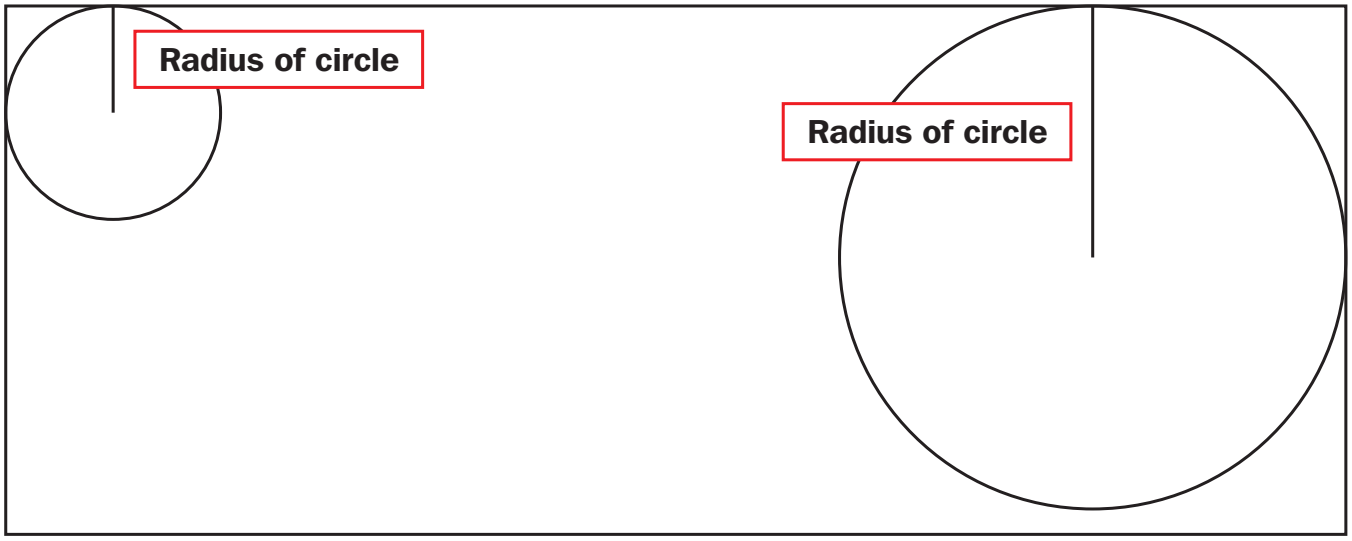
Code	Browser

<style>
div.radial_grad_1{
background: radial-gradient(
circle /*shape*/
closest-corner /*size*/
at center, /*position*/
yellow, silver, beige /*colors*/
);
height:100px;
width:200px;
}
</style>
...
 	
 This is a test div. It tests radial gradients!
 	
	

<style>
div.radial_grad_2{
background: radial-gradient(
yellow, silver, beige /*colors*/
);
height:100px;
width:200px;
}
</style>
...
 	
 This is a test div. It tests radial gradients!
 	
	

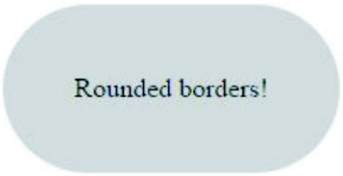
<style>
 	
div.radial_grad_3{
background: radial-gradient(
at left, /*position*/
yellow, silver, beige /*colors*/
);
height:100px;
width:200px;
}
</style>
...
 	
 This is a test div. It tests radial gradients!
 	
	

CSS3 ROUNDING CORNERS



CSS3 ROUNDED CORNER EXAMPLES

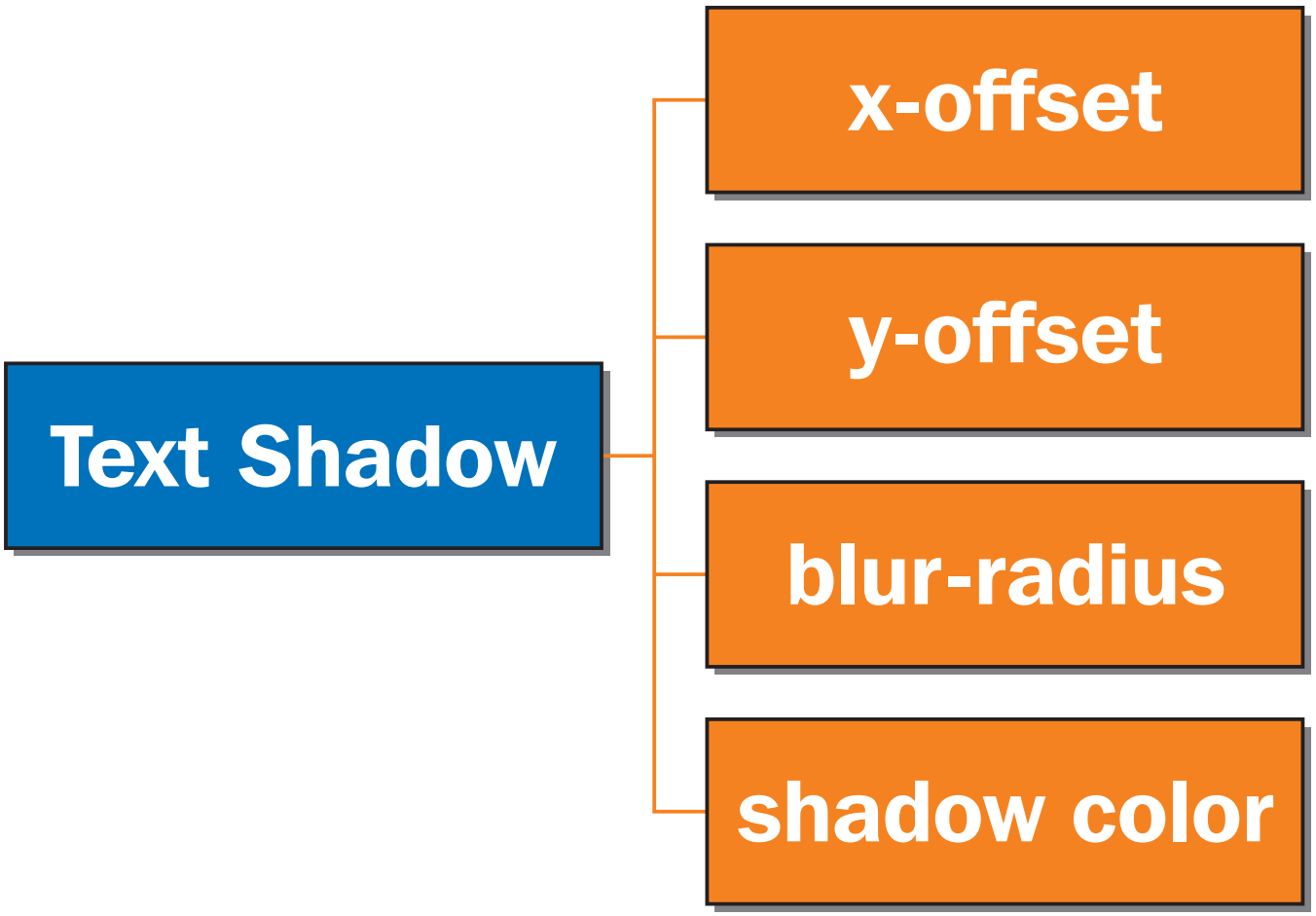
Code	Browser

<style>
div.rounded_border_1{
background: silver;
height:100px;
width:200px;
border-radius: 100px;
text-align:center;
line-height:100px;
}
</style>
...
 	
 Rounded borders!
 	
	

<style>
div.radial_grad_2{
background: beige;
height:100px;
width:200px;
border-radius: 50% ;
text-align:center;
line-height:100px;
}
</style>
...
 	
 Rounded borders!
 	
	

<style>
div.radial_grad_3{
background: salmon;
height:100px;
width:200px;
/*top-left, bottom-right are 100px
top-right, bottom-left are 50px*/
border-radius: 100px 50px;
text-align:center;
line-height:100px;
}
</style>
...
 	
 Rounded borders!
 	
	

CSS3 TEXT SHADOWS



CSS3 TEXT SHADOW EXAMPLES

Code	Browser

<style>
.shadow1{
font-size:4em;
text-shadow:
4px /*x-offset*/
2px /*y offset*/
10px /*blur radius */
gray; /*shadow color*/
}
</style>
..
 	
Shadows!
 	
	

<style>
.shadow2{
font-size:4em;
text-shadow:3px 3px 10px yellow,
5px 8px 10px red;
}
</style>
..
 	
Shadows!
 	
	

<style>
.shadow3{
font-size:4em;
text-shadow:-6px -10px lightblue;
}
</style>
..
 	
Shadows!
 	
	

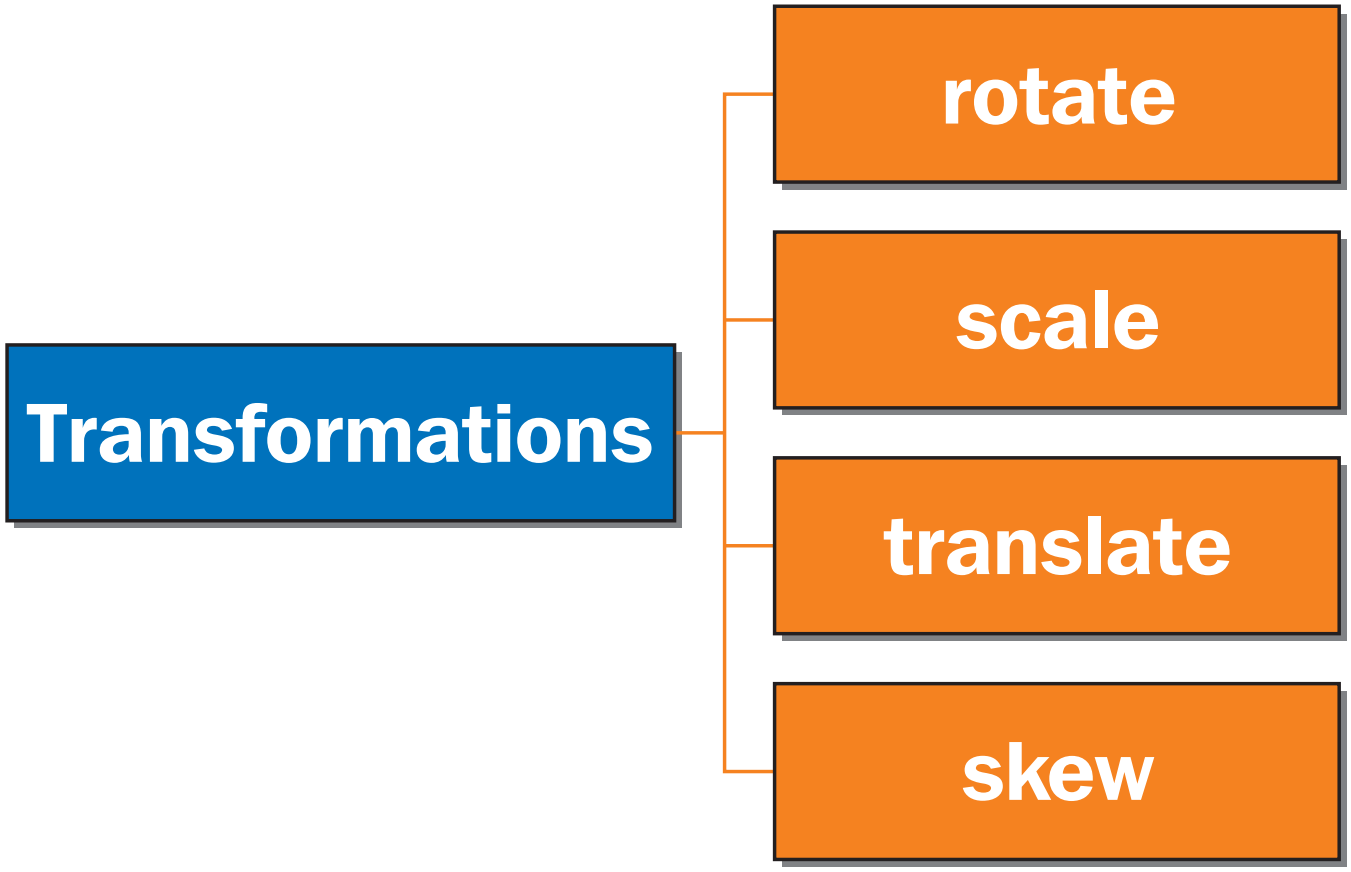
CSS3 BOX AND TEXT SHADOWS

Code	Browser

<style>
div.d_shadow1{
background-color:beige;
border:1px solid black;
box-shadow:10px 10px 15px yellow;
border-radius:20px;
height:150px;
width:150px;
margin:50px;
}
p.p_shadow1
{
font-size:2em;
text-shadow:5px -10px 5px green;
}
</style>
..
 	
 Shadows!
 	
	

<style>
div.d_shadow2{
background-color:silver;
border:1px solid black;
box-shadow:10px -10px 15px gray;
height:150px;
width:150px;
margin:50px;
}
</style>
..
 	
 Shadows!
 	
	

CSS3 TRANSFORM ACTIONS

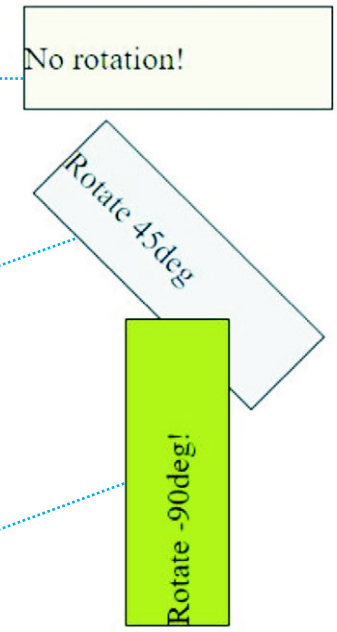


VM–V

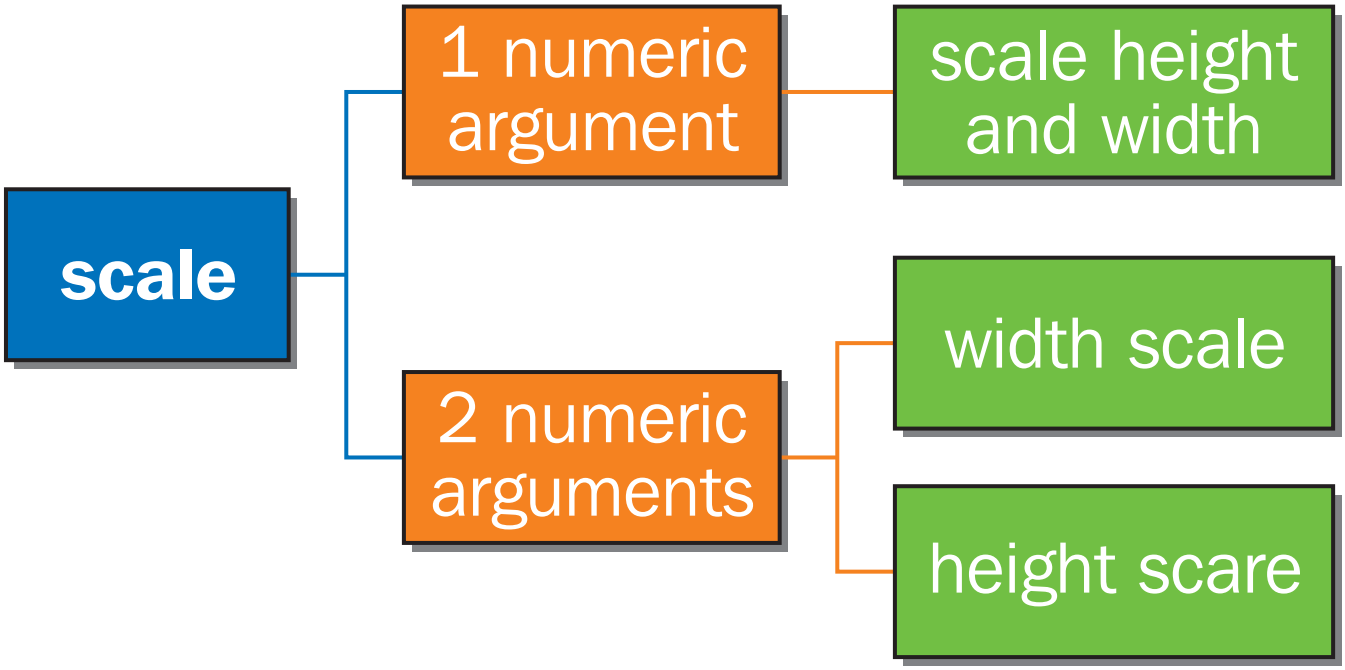
CSS3 TRANSFORM: ROTATE

Code	Browser

```
<!DOCTYPE html>
<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
    <style>
        div.no_rotate{
            border:1px solid black;
            height:50px;
            width:150px;
            margin:50px;
            background:ivory;
        }
        div.rotate_45{
            border:1px solid black;
            height:50px;
            width:150px;
            margin:50px;
            transform:rotate(45deg);
            background:aliceblue;
        }
        div.rotate_minus_90{
            border:1px solid black;
            height:50px;
            width:150px;
            margin:50px;
            transform:rotate(-90deg);
            background:yellowgreen;
        }
    </style>
</head>
<body>
    <div class = "no_rotate">
        <p>No rotation!</p>
    </div>
    <div class = "rotate_45">
        <p class = "p_shadow1">Rotate 45deg</p>
    </div>
    <div class = "rotate_minus_90">
        <p>Rotate -90deg!</p>
    </div>

</body>
</html>
```

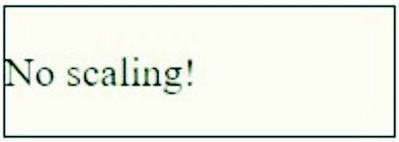


CSS3 TRANSFORM: SCALE

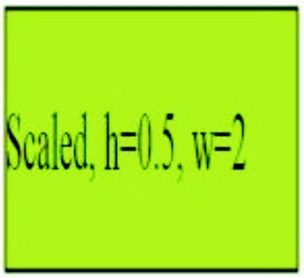


CSS3 TRANSFORM: SCALE EXAMPLES

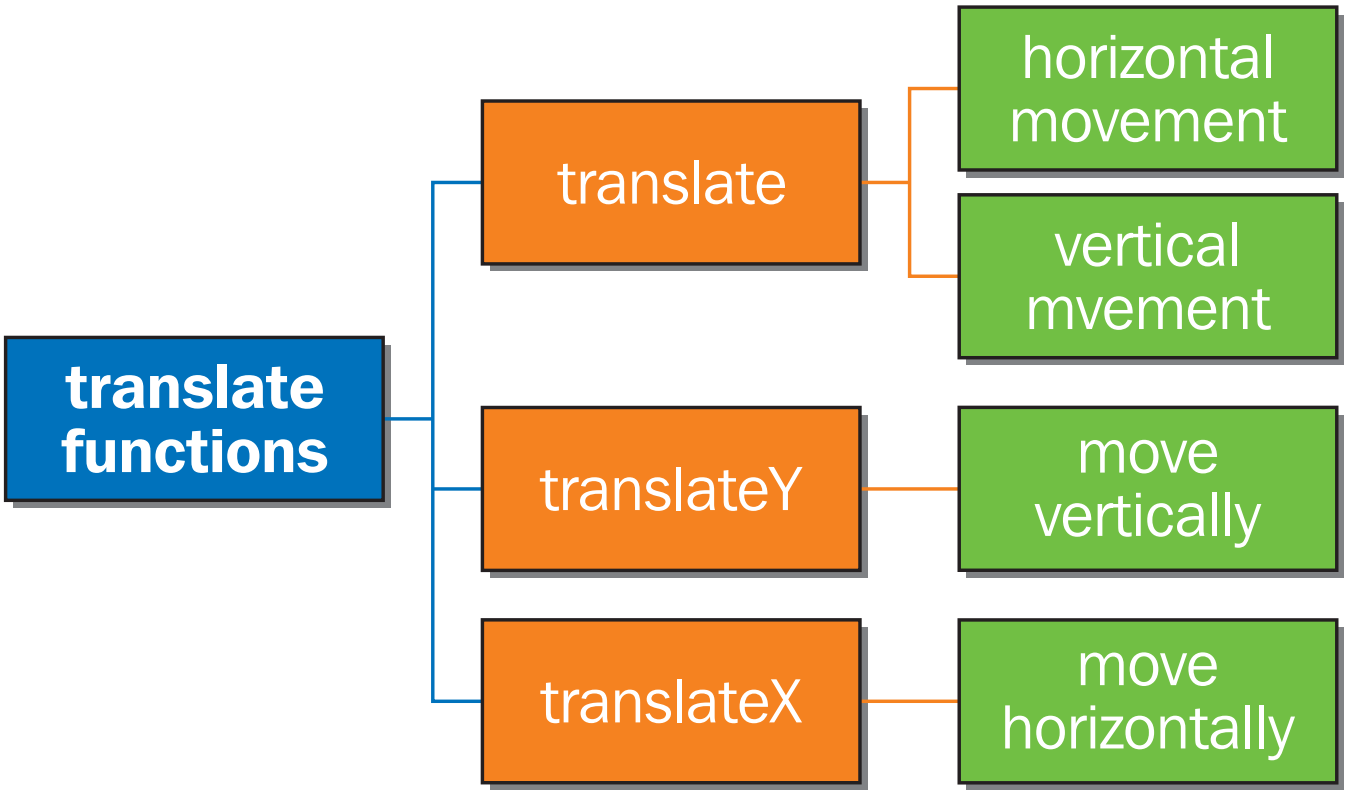
Code	Browser

`div.no_scale{`
`  border:1px solid black;`
`  height:50px;`
`  width:150px;`
`  margin:50px;`
`  background:ivory;`
`}`
`.`
`.`
`<div class = "no_scale">`
`  <p>No scaling!</p>`
`</div>`
	

`div.scale_1_argument{`
`  border:1px solid black;`
`  height:50px;`
`  width:150px;`
`  margin:50px;`
`  transform:scale(0.5);`
`  background:aliceblue;`
`}`
 	
`<div class = "scale_1_argument">`
`  <p class = "p_shadow1">`
`  Scaled to 0.5`
`  </p>`
`</div>`
	

`div.scale_2_arguments{`
`  border:1px solid black;`
`  height:50px;`
`  width:150px;`
`  margin:50px;`
`  transform:scale(0.75, 2);`
`  background:yellowgreen;`
`}`
 	
`<div class = "scale_2_arguments">`
`  <p>Scaled, h=0.5, w=2</p>`
`</div>`
	

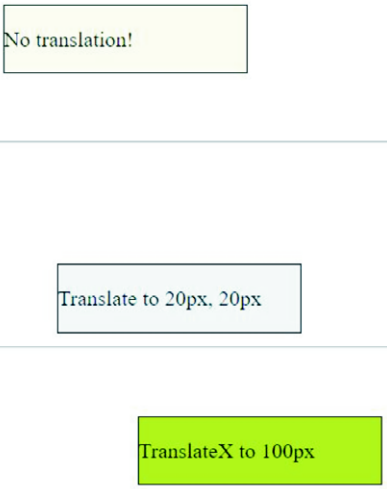
CSS3 TRANSFORM: TRANSLATE



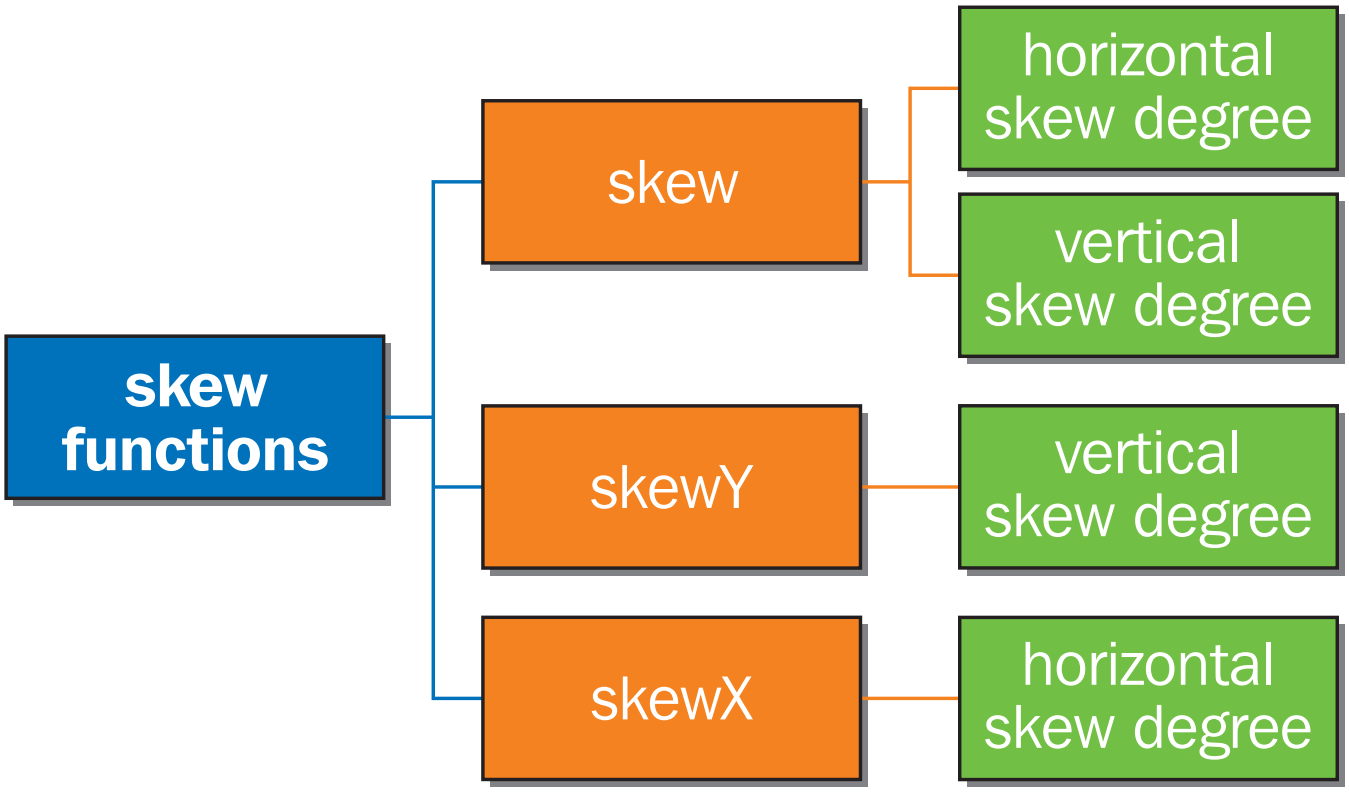
CSS3 TRANSFORM: TRANSLATE EXAMPLES

Code	Browser

```
<!DOCTYPE html>
<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
    <style>
        div.no_translate{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;
            background:ivory;
        }
        div.translate_2_arguments{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;
            transform:translate(40px,40px);
            background:aliceblue;
        }
        div.translateX{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;

transform:translateX(100px);
            background:yellowgreen;
        }
    </style>
</head>
<body>
    <div class = "no_translate">
        <p>No translation!</p>
    </div>
    <hr/>
    <div class =
"translate_2_arguments">
        <p class = "p_shadow1">
       Translate to 20px, 20px
   </p>
    </div>
    <hr/>
    <div class = "translateX">
        <p>TranslateX to 100px</p>
    </div>

</body>
</html>
```



CSS3 TRANSFORM: SKEW

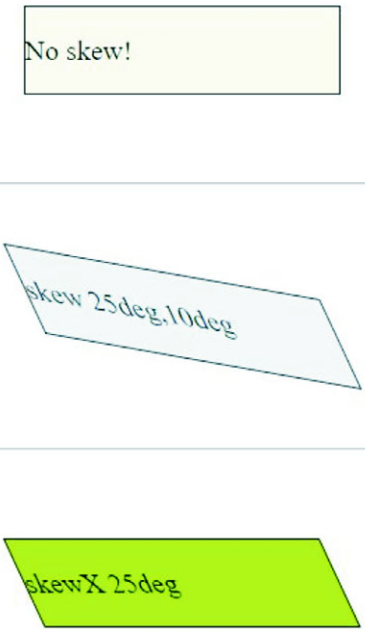


VM–BB

CSS3 TRANSFORM: SKEW EXAMPLES

Code	Browser

```
<!DOCTYPE html>
<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
    <style>
        div.no_skew{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;
            background:ivory;
        }
        div.skew_2_arguments{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;
            transform:skew(25deg, 10deg);
            background:aliceblue;
        }
        div.skewX{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;
            transform:skewX(25deg);
            background:yellowgreen;
        }
    </style>
</head>
<body>
    <div class = "no_skew">
        <p>No skew!</p>
    </div>
    <hr/>
    <div class = "skew_2_arguments">
        <p class = "p_shadow1">skew
25deg,10deg </p>
    </div>
    <hr/>
    <div class = "skewX">
        <p>skewX 25deg</p>
    </div>

</body>
</html>
```



CSS3 APPLYING MULTIPLE TRANSFORMATIONS

Code	Browser

```
<!DOCTYPE html>

<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
    <style>
        div.no_skew{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;
            background:ivory;
        }

        div.skewX_translate{
            border:1px solid black;
            height:50px;
            width:180px;
            margin:50px;
            transform:skewX(25deg)
translateX(100px);
            background:yellowgreen;
        }
    </style>
</head>
<body>
    <div class = "no_skew">
        <p>No skew!</p>
    </div>
    <hr/>
    <div class = "skewX_translate">
        <p class = "p_shadow1">skew 25deg,
translateX 100px</p>
    </div>
</body>
</html>
```

CSS3 TRANSITIONS

property

- Can be any CSS3 property
- Specify the end value for the property

duration

- seconds or milliseconds

timing

- *ease,linear,ease-in, ease-out, ease-in-out*
- *Custom function*

delay

- specified in seconds or milliseconds
- a single number or a comma separated list

CSS3 TRANSITION EXAMPLES

Code

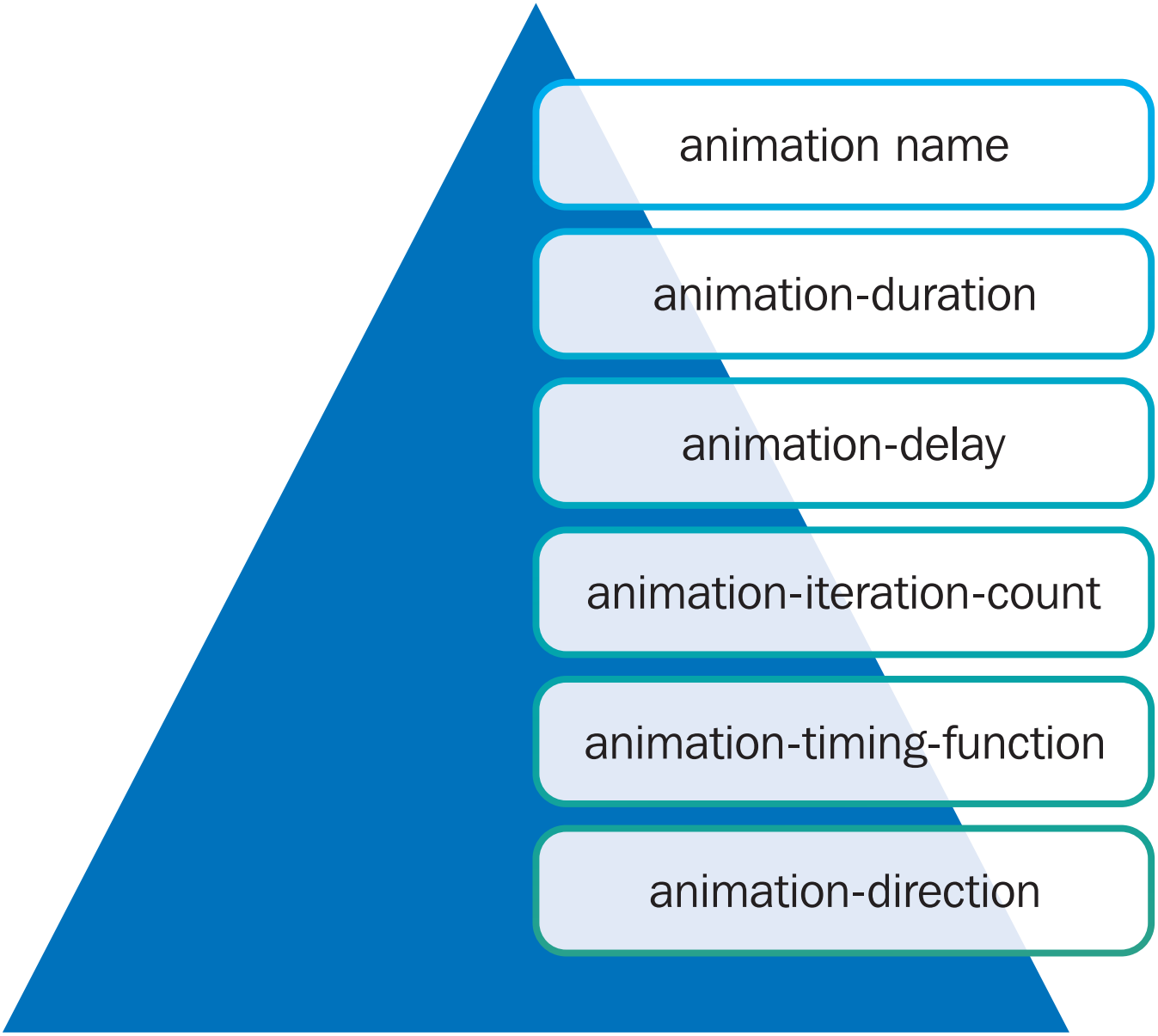
```
<!DOCTYPE html>

<html lang="en">
<head>
<style>
    div{
        width: 200px;
        padding: 10px;
        background-color:beige;
        border-radius:10px;
        transition:background-color 5s;
    }
    div:hover{
        background-color:silver;
        transition:background-color 5s;
    }

</style>
</head>

<body>
<div>
<p>Hello!</p>
</div>
</body>
</html>
```

APPLYING MULTIPLE TRANSITIONS

Code

```
<!DOCTYPE html>
<html lang="en">
<head>
<style>
    div{
        width: 200px;
        padding: 10px;
        background-color:beige;
        border-radius:10px;
        transition:background-color 5s,
                box-shadow 5s;
    }
    div:hover{
        background-color:silver;
        box-shadow: 5px 5px lightgreen;
        transition:background-color 5s,
                box-shadow 5s;
    }

</style>
</head>

<body>
<div>
<p>Hello!</p>
</div>
</body>
</html>
```

CSS3 ANIMATIONS



CSS3 ANIMATION EXAMPLE

HTML Code

```
<!doctype html>
<html lang="en">
<head>
<style>
@keyframes example {
    0% {width : 100px;}
    50% {width : 50px}
    100% {width : 100px};
}

div:hover {
    animation-name: example;
    animation-duration: 500ms;
    animation-iteration-count: infinite;
}
div{
    width: 100px;
    padding:5px;
    background-color: green;
}
</style>
</head>

<body>
<div>
<p>Hello!</p>
</div>
</body>
</html>
```

CSS3 ANIMATION EXAMPLE

HTML Code

```
<!doctype html>
<html lang="en">
<head>
<style>
@keyframes opacity {
    0%  {opacity : 1;width:100px; padding:5px;}
    20% {opacity : 1;width:100px; padding:4px;}
    25% {opacity : .2;width:80px; padding:2px;}
    50% {opacity : .2;width:80px; padding:0px;}
    60% {opacity : .2;width:80px; padding:2px;}
    80% {opacity : 1;width:100px; padding:4px;}
    100% {opacity : 1;width:100px; padding:5px;}
}

div:hover {
    animation-name: opacity;
    animation-duration: 500ms;
    animation-iteration-count: infinite;
}
div{
    width: 100px;
    padding:5px;
    background-color: cyan;
}
</style>
</head>

<body>
<div>
<p>Hello!</p>
</div>
</body>
</html>
```

LS–A

Name _____

Create Webpages Using Webfonts and Features of CSS3

Purpose

The purpose of this activity is to practice using webfonts and new CSS3 features in a webpage.

Objectives

1. Set up a CSS document with class selectors using CSS3 properties.
2. Create an HTML document.
3. Create content in the HTML document to test class selectors.
4. Validate the HTML document.
5. Test the HTML document in an HTML5 compliant browser.

Materials

- lab sheet
- computer with a text editor
- flashdrive

Procedure

1. Read the following instructions, and code CSS and HTML files. Save both the files on your flashdrive.
2. Create a CSS file called site.css.
    a. Create a class selector to display text with a webfont and a shadow. Name the class selector “text_shadow.”

b. Create a class selector called “rounded_shaded_box” that creates a rounded border and places a gradient background in the element.

3. Create an HTML document, and save it to the same folder as the CSS file.
   a. Link the CSS file to the HTML document.
   b. Place content in the HTML file, and use the selectors placed in the CSS file.
   c. Validate the HTML document at https://validator.w3.org/.
   d. Test the HTML document in an HTML5 compliant browser.
4. Turn in your completed lab sheet and files to your instructor.

Create Webpages Using Webfonts and Features of CSS3

There are two sample documents. One is the CSS file, and the other is the HTML document. Student work will look different depending upon the properties selected.

Code for Lab A site.css File

```
.text_shadow{
  font-family: 'Mogra', cursive;
  text-shadow:-6px -10px lightblue;
}
.rounded_shaded_box{
  background:linear-gradient(to top, yellow, lightblue);
  width: 400px;
  text-align:center;/*performs horizontal centering*/
  display:table;/*performs vertical cenering*/
  padding:5px;
  border-radius: 50px;
}
```

Code for Lab-A.html File

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8" />
  <title>Lab Sheet A</title>
  <link href="https://fonts.googleapis.com/css?family=Mogra" rel="stylesheet"/>
  <link rel="stylesheet" type="text/css" href="site.css" />
</head>

<body>
  <h1 class = "text_shadow">Here is a h1 Heading</h1>
  <div class = "rounded_shaded_box">
     <p>Lorem ipsum dolor sit amet, consectetur
  elit, sed do eiusmod tempor
  incididunt ut labore
  et dolore magna aliqua.
     </p>
  </div>
</body>
</html>
```

LS–B

Name _____

Create Webpages that Implement CSS3 Transforms, Transitions, and Animations

Purpose

The purpose of this activity is to practice implementing CSS3 transforms, transitions, and animations.

Objectives

1. Set up a CSS file with classes.
2. Practice with CSS3 transforms, transitions, and animations.
3. Create an HTML document.
4. Validate the HTML document.
5. Test the HTML document in an HTML5 compliant browser.

Materials

- lab sheet
- computer with multiple browser applications (e.g., Google Chrome, Firefox, Internet Explorer, and NotePad++)

Procedure

1. Read the following instructions. Write the HTML statement that fulfills the instruction.
2. Create a CSS file called site.css.
    a. Create a class called rollover_text that renders text in black without a shadow.

b. Create a pseudo-class for rollover_text that is used when the mouse moves over a rolloever_text element. The style should render text with a shadow and set text color to blue. The transition should be performed over two seconds.

3. Create an HTML document that makes use of the site.css file created above.
   a. Create a heading element that makes use of the styles created in the site.css file, and test the transition.
   b. Validate the HTML document at https://validator.w3.org/.
   c. Test the HTML document in an HTML5 compliant browser.
4. Turn in your completed lab sheet and code to your instructor.

Create Webpages that Implement CSS3 Transforms, Transitions, and Animations

There are two sample documents. One is the CSS file, and the other is the HTML document. Student work will look different depending upon the properties selected.

Code for Lab B site.css File

```
.rollover_text {
  color:black;
  text-shadow:0 0;
  transition:color 5s, text-shadow 2s;
}

.rollover_text:hover{
  color:blue;
  text-shadow:2px -4px beige;
  transition:color 5s, text-shadow 2s;

}
```

Code for Lab-B.html File

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8" />
  <title>Lab Sheet B</title>
  <link href="https://fonts.googleapis.com/css?family=Mogra" rel="stylesheet"/>
  <link rel="stylesheet" type="text/css" href="site.css" />
</head>

<body>
  <h1 class ="rollover_text">Here is a h1 Heading</h1>
  <p>Lorem ipsum dolor sit amet, consectetur
  elit, sed do eiusmod tempor
  incididunt ut labore
  et dolore magna aliqua.
  </p>
</body>
</html>
```