What is CSS?

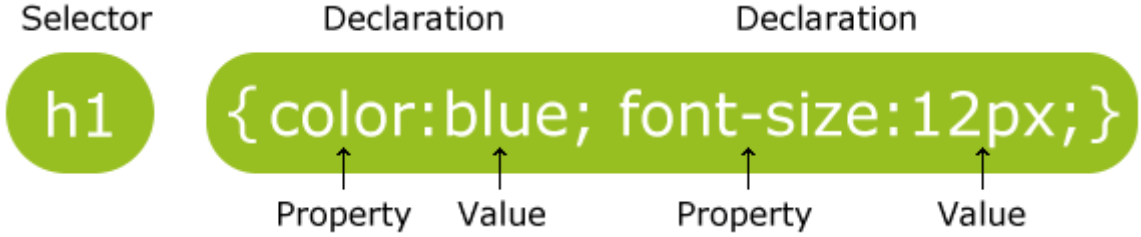
- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS defines **how HTML elements are to be displayed**
- Styles were added to HTML 4.0 **to solve a problem**
- CSS saves a lot of work
- External Style Sheets are stored in **CSS files**

CSS Saves a Lot of Work!

- The style definitions are normally saved in external .css files.
- With an external style sheet file, you can change the look of an entire Web site by changing just one file!

CSS Syntax

- A CSS rule set consists of a selector and a declaration block:



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.

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

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

- To make the CSS code more readable, you can put one declaration on each line.
- In the following example all <p> elements will be center-aligned, with a red text color:

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

CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

Example

```
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
}

/* This is
a multi-line
comment */
```

CSS Selectors

CSS selectors allow you to select and manipulate HTML elements.

CSS selectors are used to "find" (or select) HTML elements based on their id, class, type, attribute, and more.

The element Selector

The element selector selects elements based on the element name.

You can select all <p> elements on a page like this: (all <p> elements will be center-aligned, with a red text color)
```

Example

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

The id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

An id should be unique within a page, so the id selector is used if you want to select a single, unique element.

To select an element with a specific id, write a hash character, followed by the id of the element.

The style rule below will be applied to the HTML element with id="para1":

Example

```
#para1 {
    text-align: center;
    color: red;
}
```

The class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period character, followed by the name of the class:

In the example below, all HTML elements with class="center" will be center-aligned:

Example

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

You can also specify that only specific HTML elements should be affected by a class.

In the example below, all <p> elements with class="center" will be center-aligned:

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

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file.

Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file called "myStyle.css", is shown below:

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
```

```
    margin-left: 20px;
}
```

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the <style> tag, like this:

Example

```
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

Inline Styles

An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly!

To use inline styles, add the style attribute to the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a h1 element:

Example

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

Multiple Styles Will Cascade into One

Styles can be specified:

- inside an HTML element
- inside the <head> section of an HTML page
- in an external CSS file

Tip: Even multiple external style sheets can be referenced inside a single HTML document.

CSS background properties are used to define the background effects of an element.

CSS properties used for background effects:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Example

```
h1 {
    background-color: #6495ed;
}

p {
    background-color: #e0ffff;
}

div {
    background-color: #b0c4de;
}
```

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.

The background image for a page can be set like this:

Example

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

Background Image - Repeat Horizontally or Vertically

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

If the image is repeated only horizontally (repeat-x), the background will look better:

Example

```
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-x;
}
```

Background Image - Set position and no-repeat

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

Example

```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
}
```

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

The position of the image is specified by the background-position property:

Example

```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
}
```

Background - Shorthand property

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

To shorten the code, 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":

Example

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

TEXT FORMATTING

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented, aligned, and the space between characters is specified. The underline is removed from the "Try it yourself" link.

Text Color

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- a color name - like "red"

Look at CSS Color Values for a complete list of possible color values.

The default color for a page is defined in the body selector.

Example

```
body {
    color: blue;
}

h1 {
    color: #00ff00;
}
```

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.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

Example

```
h1 {
    text-align: center;
}
```

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:

Example

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

It can also be used to decorate text:

Example

```
h1 {
    text-decoration: overline;
}

h2 {
    text-decoration: line-through;
}

h3 {
    text-decoration: underline;
}
```

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.

Example

```
p.uppercase {
    text-transform: uppercase;
}

p.lowercase {
    text-transform: lowercase;
}

p.capitalize {
    text-transform: capitalize;
}
```

Text Indentation

The text-indent property is used to specify the indentation of the first line of a text.

Example

```
p {
    text-indent: 50px;
}
```

CSS font properties define the font family, boldness, size, and the style of a text.

Font Family

The font family of a text is set with the font-family property.

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".

More than one font family is specified in a comma-separated list:

Example

```
p {
    font-family: "Times New Roman", Times, serif;
}
```

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)

Example

```
p.normal {
    font-style: normal;
}

p.italic {
    font-style: italic;
}

p.oblique {
    font-style: oblique;
}
```

Font Size

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

The font-size value can be an absolute, or relative size.

Absolute size:

- Sets the text to a specified size
- Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
- Absolute size is useful when the physical size of the output is known

Relative size:

- Sets the size relative to surrounding elements
- Allows a user to change the text size in browsers
- ```
  h1 {
      font-size: 40px;
  }

  h2 {
      font-size: 30px;
  }
  ```
  ```

```
p {
    font-size: 14px;
}
```

Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links can be styled differently depending on what **state** they are in.

The four links states are:

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

Example

```
/* unvisited link */
a:link {
    color: #FF0000;
}

/* visited link */
a:visited {
    color: #00FF00;
}

/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
```

When setting the style for several link states, there are some order rules:

- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

- # Text Decoration
- The text-decoration property is mostly used to remove underlines from links:
- ## Example
- ```
  a:link {
      text-decoration: none;
  }

  a:visited {
      text-decoration: none;
  }

  a:hover {
      text-decoration: underline;
  }

  a:active {
      text-decoration: underline;
  }
  ```

Background Color

The background-color property specifies the background color for links:

Example

```
a:link {
    background-color: #B2FF99;
}

a:visited {
    background-color: #FFFF85;
}

a:hover {
    background-color: #FF704D;
}

a:active {
    background-color: #FF704D;
}
```
```

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker

List

In HTML, there are two types of lists:

- unordered lists (<ul>) - the list items are marked with bullets
- ordered lists (<ol>) - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

Different List Item Markers

The type of list item marker is specified with the list-style-type property:

Example

```
ul.a {
    list-style-type: circle;
}

ul.b {
    list-style-type: square;
}

ol.c {
    list-style-type: upper-roman;
}

ol.d {
    list-style-type: lower-alpha;
}
```

An Image as The List Item Marker

To specify an image as the list item marker, use the list-style-image property:

Example

```
ul {
   list-style-image: url('sqpurple.gif');
}
```

The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to add a border around elements, and to define space between elements.

Explanation of the different parts:

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

- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

Example

```
div {
    width: 300px;
    padding: 25px;
    border: 25px solid navy;
    margin: 25px;
}
```

CSS Border Properties

The CSS border properties allow you to specify the style, size, and color of an element's border.

Border Style

The border-style property specifies what kind of border to display.

border-style values:

none: Defines no border

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

Border Width

The border-width property is used to set the width of the border.

The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.

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

Example

```
p.one {
    border-style: solid;
    border-width: 5px;
}

p.two {
    border-style: solid;
    border-width: medium;
}
```

Try it yourself »

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".

If the border color is not set it is inherited from the color property of the element.

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

Example

```
p.one {
    border-style: solid;
    border-color: red;
}

p.two {
    border-style: solid;
    border-color: #98bf21;
}
```

Border - Individual sides

In CSS it is possible to specify different borders for different sides:

Example

```
p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
}
```

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

The border-style property is used in the example above. However, it also works with border-width and border-color.

Border - Shorthand property

As you can see from the examples above, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property. This is called a shorthand property.

The border property is a shorthand for the following individual border properties:

- border-width
- border-style (required)
- border-color

Example

```
p {
    border: 5px solid red;
}
```

CSS Outline

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".

However, the outline property is different from the border property.

The outline is not a part of an element's dimensions; the element's total width and height is not affected by the width of the outline.

All CSS Outline Properties

Property	Description
outline	Sets all the outline properties in one declaration
outline-color	Sets the color of an outline
outline-style	Sets the style of an outline

Margin

The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.

The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once.

Possible Values

Value	Description
auto	The browser calculates a margin
length	Specifies a margin in px, pt, cm, etc. Default value is 0px
%	Specifies a margin in percent of the width of the containing element
inherit	Specifies that the margin should be inherited from the parent element

Margin - Individual sides

In CSS, it is possible to specify different margins for different sides of an element:

Example

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 50px;
}
```

Margin - Shorthand property

To shorten the code, it is possible to specify all the margin properties in one property. This is called a shorthand property.

The shorthand property for all the margin properties is "margin":

Example

```
p {
    margin: 100px 50px;
}
```

The margin property can have from one to four values.

- **margin: 25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px

- **margin: 25px 50px 75px;**
  - top margin is 25px
  - right and left margins are 50px
  - bottom margin is 75px

- **margin: 25px 50px;**
  - top and bottom margins are 25px
  - right and left margins are 50px

- **margin: 25px;**
  - all four margins are 25px

CSS Padding

The CSS padding properties define the space between the element border and the element content.

Padding

The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.

The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.

Possible Values

Value	Description
length	Defines a fixed padding (in pixels, pt, em, etc.)
%	Defines a padding in % of the containing element

Padding - Individual sides

In CSS, it is possible to specify different padding for different sides:

Example

```
p {
    padding-top: 25px;
    padding-right: 50px;
    padding-bottom: 25px;
    padding-left: 50px;
}
```

Try it yourself »

Padding - Shorthand property

To shorten the code, it is possible to specify all the padding properties in one property. This is called a shorthand property.

The shorthand property for all the padding properties is "padding":

Example

```
p {
    padding: 25px 50px;
}
```

Try it yourself »

The padding property can have from one to four values.

- **padding: 25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px

- **padding: 25px 50px 75px;**
  - top padding is 25px
  - right and left paddings are 50px
  - bottom padding is 75px
  -
- **padding: 25px 50px;**
  - top and bottom paddings are 25px
  - right and left paddings are 50px

- **padding: 25px;**
  - all four paddings are 25px

Hiding an Element - display:none or visibility:hidden

Hiding an element can be done by setting the display property to "none" or the visibility property to "hidden". However, notice that these two methods produce different results:

visibility:hidden hides an element, but it will still take up the same space as before. The element will be hidden, but still affect the layout:

Example

```
h1.hidden {
    visibility: hidden;
}
```

Example

```
h1.hidden {
    display: none;
}
```

Changing How an Element is Displayed

Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow web standards.

The following example displays <li> elements as inline elements:

Example

```
li {
    display: inline;
}
```

CSS Positioning

Positioning

The CSS positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.

Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.

There are four different positioning methods.

Static Positioning

HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.

Static positioned elements are not affected by the top, bottom, left, and right properties.

Fixed Positioning

An element with a fixed position is positioned relative to the browser window, and will not move even if the window is scrolled:

Example

```
p.pos_fixed {
    position: fixed;
    top: 30px;
    right: 5px;
}
```

Relative Positioning

A relative positioned element is positioned relative to its normal position:

Example

```
h2.pos_left {
    position: relative;
    left: -20px;
}

h2.pos_right {
    position: relative;
    left: 20px;
}
```

The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.

Example

```
h2.pos_top {
    position: relative;
    top: -50px;
}
```

Absolute Positioning

An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>:

Example

```
h2 {
    position: absolute;
    left: 100px;
    top: 150px;
}
```

Try it yourself »

Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.

Absolutely positioned elements can overlap other elements.

Overlapping Elements

When elements are positioned outside the normal flow, they can overlap other elements.

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

An element can have a positive or negative stack order:

Example

```
img {
    position: absolute;
    left: 0px;
    top: 0px;
```

```
    z-index: -1;
}
```

What is CSS Float?




With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

Float is often used with images, but it is also useful when working with layouts.

How Elements Float

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.

A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.

The elements after the floating element will flow around it.

The elements before the floating element will not be affected.

If an image is floated to the right, a following text flows around it, to the left:

Example

```
img {
    float: right;
}
```

CSS Image Opacity Transparency

Creating transparent images with CSS is easy.

The CSS opacity property is a part of the CSS3 recommendation.

Example 1 - Creating a Transparent Image

The CSS3 property for transparency is **opacity**.

First we will show you how to create a transparent image with CSS.

Regular image:

The same image with transparency:

Look at the following CSS:

Example

```
img {
    opacity: 0.4;
    filter: alpha(opacity=40); /* For IE8 and earlier */
}
```

Example 2 - Text in Transparent Box

This is some text that is placed in the transparent box.

The source code looks like this:

Example

```
<html>
<head>
<style>
div.background {
    background: url(klematis.jpg) repeat;
    border: 2px solid black;
}

div.transbox {
    margin: 30px;
    background-color: #ffffff;
    border: 1px solid black;
    opacity: 0.6;
    filter: alpha(opacity=60); /* For IE8 and earlier */
}

div.transbox p {
    margin: 5%;
    font-weight: bold;
    color: #000000;
}
</style>
</head>
<body>

<div class="background">
  <div class="transbox">
```

```
    <p>This is some text that is placed in the transparent
box.</p>
  </div>
</div>

</body>
</html>
```

CSS3 Borders

With CSS3, you can create rounded borders, add shadow to boxes, and use an image as a border - without using a design program, like Photoshop.

In this chapter you will learn about the following border properties:

- border-radius
- box-shadow
- border-image

CSS3 The border-radius Property - Rounded Corners

Adding rounded corners in CSS2 was tricky. We had to use different images for each corner.

In CSS3, creating rounded corners is easy.

In CSS3, the border-radius property is used to create rounded corners:

This box has rounded corners!

Example

Add rounded corners to a element:

```
div {
    border: 2px solid;
    border-radius: 25px;
}
```

CSS3 The box-shadow Property

In CSS3, the box-shadow property is used to add shadow to boxes:

Example

Add a box-shadow to a element:

```
div {
    box-shadow: 10px 10px 5px #888888;
}
```

CSS3 The border-image Property

With the CSS3 border-image property you can use an image to create a border:

The border-image property allows you to specify an image as a border!

The original image used to create the border above:

Example

Use an image to create a border around a element:

```
div {
    -webkit-border-image: url(border.png) 30 30 round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 30 30 round; /* Opera 11-12.1 */
    border-image: url(border.png) 30 30 round;
}
```

CSS3 Backgrounds

CSS3 contains several new background properties,
which allow greater control of the background element.

In this chapter you will learn about the following background properties:

- background-size
- background-origin

CSS3 The background-size Property

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

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.

Example 1

Resize a background image:

```
div {
    background: url(img_flwr.gif);
    background-size: 80px 60px;
    background-repeat: no-repeat;
}
```

CSS3 Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

Example of Linear Gradient:

Syntax

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

Linear Gradient - Top to Bottom (this is default)

The following example shows a linear gradient that starts at the top. It starts red, transitioning to blue:

Example

A linear gradient from top to bottom:

```
#grad {
  background: -webkit-linear-gradient(red, blue); /* For Safari
5.1 to 6.0 */
  background: -o-linear-gradient(red, blue); /* For Opera 11.1 to
```

CSS3 Text Effects

CSS3 contains several new text features.

In this chapter you will learn about the following text properties:

- text-shadow

word-wrap

CSS3 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:

Example

Add a shadow to a <h1> element:

```
h1 {
    text-shadow: 5px 5px 5px #FF0000;
}
```

CSS3 2D Transforms

In this chapter you will learn about the 2d transform methods:

- translate()
- rotate()
- scale()
- skew()
- matrix()

You will learn about 3D transforms in the next chapter.

The rotate() Method

With the rotate() method, the element rotates clockwise at a given degree. Negative values are allowed and rotates the element counter-clockwise.

Example

```
div {
    -ms-transform: rotate(30deg); /* IE 9 */
    -webkit-transform: rotate(30deg); /* Chrome, Safari, Opera */
    transform: rotate(30deg);
}
```

The value rotate(30deg) rotates the element clockwise 30 degrees.

The scale() Method

With the scale() method, the element increases or decreases the size, depending on the parameters given for the width (X-axis) and the height (Y-axis):

Example

```
div {
    -ms-transform: scale(2,4); /* IE 9 */
    -webkit-transform: scale(2,4); /* Chrome, Safari, Opera */
    transform: scale(2,4);
}
```

The value scale(2,4) transforms the width to be twice its original size, and the height 4 times its original size.

The skew() Method

With the skew() method, the element turns in a given angle, depending on the parameters given for the horizontal (X-axis) and the vertical (Y-axis) lines:

Example

```
div {
    -ms-transform: skew(30deg,20deg); /* IE 9 */
    -webkit-transform: skew(30deg,20deg); /* Chrome, Safari, Opera */
    transform: skew(30deg,20deg);
}
```

The value skew(30deg,20deg) turns the element 30 degrees around the X-axis, and 20 degrees around the Y-axis.

The matrix() Method

The matrix() method combines all of the 2D transform methods into one.

The matrix method take six parameters, containing mathematic functions, which allows you to: rotate, scale, move (translate), and skew elements.

Example

How to rotate a div element 30 degrees, using the matrix method:

```
div {
    -ms-transform: matrix(0.866,0.5,-0.5,0.866,0,0); /* IE 9 */
    -webkit-transform: matrix(0.866,0.5,-0.5,0.866,0,0); /* Chrome,
Safari, Opera */
    transform: matrix(0.866,0.5,-0.5,0.866,0,0);
}
```

CSS3 Animation

When an animation is created in the @keyframe rule, you must bind it to a selector, otherwise the animation will have no effect.

Bind the animation to a selector (element) by specifying at least these two properties:

- the name of the animation
- the duration of the animation

Example

Bind the "myfirst" animation to the element. Animation duration: 5 seconds:

```
div {
    -webkit-animation: myfirst 5s; /* Chrome, Safari, Opera */
    animation: myfirst 5s;
}

/* Chrome, Safari, Opera */
@-webkit-keyframes myfirst {
    from {background: red;}
    to {background: yellow;}
}

/* Standard syntax */
@keyframes myfirst {
    from {background: red;}
    to {background: yellow;}
}
```

Note: If the duration part is not specified, the animation will have no effect, because the default value is 0.

CSS3 User Interface

In CSS3, the resize property specifies whether or not an element should be resizable by the user.

```
div {
    resize: both;
    overflow: auto;
}
```

CSS3 Box Sizing

The box-sizing property is used to tell the browser what the sizing properties (width and height) should include.

Should they include the border-box or just the content-box which is the default value of the width and height properties.

For example, if you want two bordered boxes side by side, it can be achieved through setting box-sizing to "border-box". This forces the browser to render the box with the specified width and height, and place the border and padding inside the box.

Example

Specify two bordered boxes side by side:

```
div {
    -moz-box-sizing: border-box; /* Firefox */
    box-sizing: border-box;
    width: 50%;
    float: left;

 }
```

CSS3 Multiple Columns

With CSS3, you can create multiple columns for laying out text - like in newspapers!

In this chapter you will learn about the following multiple column properties:

- column-count

- column-gap
- column-rule

CSS3 Create Multiple Columns

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

Example

Divide the text in a element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

CSS3 Specify the Gap Between Columns

The column-gap property specifies the gap between the columns:

Example

Specify a 40 pixels gap between the columns:

```
div {
    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

CSS3 Column Rules

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

Example

Specify the width, style and color of the rule between columns:

```
div {
    -webkit-column-rule: 3px outset #ff00ff; /* Chrome, Safari, Opera
*/
    -moz-column-rule: 3px outset #ff00ff; /* Firefox */
    column-rule: 3px outset #ff00ff;
}
```

CSS3 Outline Offset

The outline-offset property offsets an outline, and draws it beyond the border edge.

Outlines differ from borders in two ways:

- Outlines do not take up space
- Outlines may be non-rectangular

This div has an outline 15px outside the border edge.

The CSS code is as follows:

Example

Specify an outline 15px outside the border edge:

```
div {
    border: 2px solid black;
    outline: 2px solid red;
    outline-offset: 15px;
}
```

CSS Responsive Web Design

What is Responsive Web Design?

Responsive web design is to create web sites that look good on all devices.

Responsive web design is not a program or a JavaScript.

Responsive web design is a must for mobile devices.

Using Bootstrap

Another way to create a responsive design, is to use an already existing CSS framework.

Bootstrap is the most popular HTML, CSS, and JavaScript framework for responsive web design.

Bootstrap helps you develop sites that look nice at any size; screen, laptop, tablet, or phone:

```
<!DOCTYPE html>
<html>
<head>
```

```
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap
/3.2.0/css/bootstrap.min.css">
</head>

<body>
<div class="container">

<div class="jumbotron">
  <h1>W3Schools Demo</h1>
  <p>Resize this responsive page!</p>
</div>

<div class="row">
  <div class="col-md-4">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>It is the most populous city in the United Kingdom,
    with a metropolitan area of over 13 million inhabitants.</p>
  </div>
  <div class="col-md-4">
    <h2>Paris</h2>
    <p>Paris is the capital and most populous city of France.</p>
  </div>
  <div class="col-md-4">
    <h2>Tokyo</h2>
    <p>Tokyo is the capital of Japan, the center of the Greater Tokyo
Area,
    and the most populous metropolitan area in the world.</p>
  </div>
</div>

</div>
</body>
</html>
```