What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once External stylesheets are stored in CSS files

Why Use CSS?

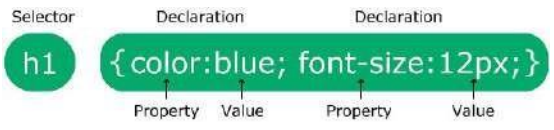
CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes. CSS Example

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

h1        {
color:
white;
text-
align:
center;
}

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

CSS Syntax



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 CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

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

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

The CSS element Selector

The element selector selects HTML elements based on the element name.

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

The CSS id Selector

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

The id of an element is unique within a page, so the id selector is used to select one unique element!

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

Example

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

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

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

Example

In this example all HTML elements with class="center" will be red and center-aligned:

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

Example

In this example only <p> elements with class="center" will be red and center-aligned: p.center {   text-align: center;   color: red; }

HTML elements can also refer to more than one class.

Example

In this example the <p> element will be styled according to class="center" and to class="large":

<p class="center large">This paragraph refers to two

classes.</p> The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Example

The CSS rule below will affect every HTML element on the page:

```
* {
  text-
align:
center;
color:
blue; }
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style

definitions. Look at the following CSS code (the h1, h2, and p elements have the

same style definitions):

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

h2        {
text-
align:
center;
color:
red;
}

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

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

```
h1, h2, p
{   text-
align:
center;
color:
red; }
```

All CSS Simple Selectors

Selector	Example	Example description
#id	#firstname	Selects the element with id="firstname"
.class	.intro	Selects all elements with class="intro"
element.class	p.intro	Selects only <p> elements with class="intro"
*	*	Selects all elements
element	p	Selects all <p> elements

element,element,..	div, p	Selects all <div> elements and all <p> elements

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

Example

External styles are defined within the <link> element, inside the <head> section of an HTML page:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

The external .css file should not contain any HTML tags.

Here is how the "mystyle.css" file looks:

"mystyle.css"

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

h1 {
color:
navy;
margin-
left:
20px; }
```

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

Example

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<!DOCTYPE html>
<html>
<head> <style>
body {
background-
color: linen;
}

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

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Example

Inline styles are defined within the "style" attribute of the relevant element:

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

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

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

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

CSS Comments

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

Comments are ignored by browsers.

A CSS comment is placed inside the <style> element, and starts with /* and ends with */:

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

Example

```
p {   color: red;  /* Set text
color to red */ }
```

Example

```
/*
T
hi
s
is
a
m
ul
ti-
li
ne
co
m
m
en
t
*/
```

```
p
{
co
lo
r:
re
d;
}
```

HTML and CSS Comments

From the HTML tutorial, you learned that you can add comments to your HTML source by using the `<!--...-->` syntax.

In the following example, we use a combination of HTML and CSS comments:

```
<!DOCTYPE html>
<html>
<head> <style> p {   color:
red; /* Set text color to red
*/
}
</style
>
</head
>
<body>

<h2>My Heading</h2>

<!-- These paragraphs will be red -->
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>

</body>
</html>
```

CSS Background Color

You can set the background color for HTML elements:

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

CSS Text Color

You can set the color of text:

Hello World

Example

```
<h1 style="color:Tomato;">Hello World</h1>
```

```
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

CSS Border Color

You can set the color of borders:

Hello World

Hello World

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

CSS Color Values

In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

The CSS background properties are used to add background effects for elements.

In these chapters, you will learn about the following CSS background properties:

background-color

background-image

background-repeat

background-attachment

background-position

background (shorthand

property)

CSS background-color

The background-color property specifies the background color of an element.

Example

The background color of a page is set like this:

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

With CSS, a color is most often specified by:

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

Other Elements

You can set the background color for any HTML elements:

Example

Here, the <h1>, <p>, and <div> elements will have different background colors:

```
h1 {
background-
color: green;
}

div {
  background-color: lightblue;
}

p {   background-
color: yellow; }
```

Opacity / Transparency

The opacity property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0. The lower value, the more transparent:

opacity 1, opacity 0.6, opacity 0.3, opacity 0.1

Example

```
div {
  background-
color: green;
opacity: 0.3; }
```

Transparency using RGBA

If you do not want to apply opacity to child elements, like in our example above, use RGBA color values. The following example sets the opacity for the background color and not the text:

100% opacity, 60% opacity, 30% opacity, 10% opacity

You learned from our CSS Colors Chapter, that you can use RGB as a color value. In addition to RGB, you can use an RGB color value with an alpha channel (RGBA) - which specifies the opacity for a color.

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).

```
div {   background: rgba(0, 128, 0, 0.3) /* Green background with
30% opacity */ }
```

CSS 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.

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

CSS background-repeat

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

vertically. Some images should be repeated only horizontally or vertically, or they will

look strange, like this:

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

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

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

CSS background-repeat: no-repeat

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

Example

Show the background image only once:

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

CSS background-position

The background-position property is used to specify the position of the background image.

Example

Position the background image in the top-right corner:

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

CSS background-attachment

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page):

Example

Specify that the background image should be fixed:

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

Example

Specify that the background image should scroll with the rest of the page:

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

CSS background - Shorthand property

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

Instead of writing:

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

All CSS Background Properties

Property	Description
background	Sets all the background properties in one declaration
backgroundattachment	Sets whether a background image is fixed or scrolls with the rest of the page

background-clip	Specifies the painting area of the background
background-color	Sets the background color of an element
backgroundimage	Sets the background image for an element
background-origin	Specifies where the background image(s) is/are positioned
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated
background-size	Specifies the size of the background image(s)

CSS Border Style

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

The following values are allowed:

- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- 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 none - Defines no border
- hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example

Demonstration of the different border styles:

```
p.dotted {border-style: dotted;}
p.dashed {border-style:
dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style:
groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

CSS Border Width

The border-width property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick:

Example

Demonstration of the different border widths:

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

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

p.three {
border-style:
dotted;
border-
width: 2px;
}

p.four {
border-
style:
dotted;
border-
width:
thick; }
```

Specific Side Widths

The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border):

```
p.one {   border-style: solid;   border-width: 5px 20px; /*
5px top and bottom, 20px on the sides */ }

p.two {
  border-style: solid;
  border-width: 20px 5px; /* 20px top and bottom, 5px on
the sides */ }

p.three {
border-style:
solid;
  border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and
35px left */ }
```

CSS Border Color

The border-color property is used to set the color of the four borders.

The color can be set by:

- name - specify a color name, like "red"
- HEX - specify a HEX value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
- transparent

Note: If border-color is not set, it inherits the color of the element.

Example

Demonstration of the different border colors:

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

p.two       {
border-
style:  solid;
border-
color:
green;
}

p.three {
border-
style:
```

```
dotted;
border-
color: blue;
}
```

If the border-style property has four values:

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

If the border-style property has three values:

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

If the border-style property has two values:

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

If the border-style property has one value:

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

```
/* Four
values */
p {
 border-style: dotted solid double dashed;
}

/* Three
values */
p {
 border-style: dotted solid double;
}

/* Two
values */
p {
 border-style: dotted solid;
}

/* One
value */ p {
```

```
border-
style:
dotted; }
```

CSS Border - Shorthand Property

Like you saw in the previous page, 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.

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

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

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

CSS Rounded Borders

The border-radius property is used to add rounded borders to an element:

Normal border

Round border

Rounder border

Roundest border

```
p {   border:
2px solid
red;   border-
radius: 5px;
}
```

All CSS Border Properties

Property	Description
border	Sets all the border properties in one declaration
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border

border-bottom-style	Sets the style of the bottom border
border-bottomwidth	Sets the width of the bottom border
border-color	Sets the color of the four borders
border-left	Sets all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
borderradius	Sets all the four border-*-radius properties for rounded corners
border-right	Sets all the right border properties in one declaration
borderrightcolor	Sets the color of the right border
borderrightstyle	Sets the style of the right border

borderrightwidth	Sets the width of the right border
border-style	Sets the style of the four borders
border-top	Sets all the top border properties in one declaration
bordertop-color	Sets the color of the top border
bordertop-style	Sets the style of the top border
bordertop-width	Sets the width of the top border
borderwidth	Sets the width of the four borders

CSS Margins The CSS margin properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- margin-top
- margin-right
- margin-bottom
- margin-left

All the margin properties can have the following values:

- auto - the browser calculates the margin length - specifies a margin in px, pt, cm, etc.
- % - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

```
p {   margin-
top: 100px;
margin-
bottom:
```

```
100px;
margin-right:
150px;
margin-left:
80px; }
```

Margin - Shorthand Property

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

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

- margin-top
- margin-right
- margin-bottom
- margin-left

So, here is how it works:

If the margin property has four values:

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

```
p {   margin: 25px 50px
75px 100px; }
```

Example

Use the margin shorthand property with three values:

```
p {   margin:
25px 50px 75px;
}
```

The auto Value

You can set the margin property to auto to horizontally center the element within its container.

The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

Example

Use margin: auto:

```
div {   width:
300px;
margin: auto;
border: 1px
solid red; }
```

The inherit Value

This example lets the left margin of the <p class="ex1"> element be inherited from the parent element (<div>):

Example

Use of the inherit value:

```
div {
border: 1px
solid red;
margin-left:
100px;
}

p.ex1 {
margin-
left:
inherit; }
```

All CSS Margin Properties

Property	Description
margin	A shorthand property for setting all the margin properties in one declaration
marginbottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

Margin Collapse

Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.

This does not happen on left and right margins! Only top and bottom margins!

Look at the following example:

Example

Demonstration of margin collapse:

```
h1 {
margin: 0 0
50px 0;
}

h2 {
  margin:
20px 0 0 0; }
```

All CSS Margin Properties

Property	Description
margin	A shorthand property for setting all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

CSS Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

- padding-top
- padding-right
- padding-bottom
- padding-left

All the padding properties can have the following values:

- length - specifies a padding in px, pt, cm, etc.
- % - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

Example

Set different padding for all four sides of a <div> element:

```
div {
  padding-top:
50px;
padding-right:
30px;
padding-
bottom: 50px;
padding-left:
80px; }
```

Padding - Shorthand Property

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

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

- padding-top
- padding-right
- padding-bottom
- padding-left

So, here is how it works:

If the padding property has four values:

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

Example

Use the padding shorthand property with four values:

```
div {   padding: 25px
50px 75px 100px; }
```

Example

Use the padding shorthand property with three values:

```
div {   padding:
25px 50px 75px;
}
```

Example

Use the padding shorthand property with one value:

```
div {
paddin
g:
```

```
25px;
}
```

Padding and Element Width

The CSS width property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element (the box model).

So, if an element has a specified width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

Example

Here, the <div> element is given a width of 300px. However, the actual width of the <div> element will be 350px (300px + 25px of left padding + 25px of right padding):

```
div {
width:
300px;
paddin
g:
25px;
}
```

Example

Use the box-sizing property to keep the width at 300px, no matter the amount of padding:

```
div {   width:
300px;
padding: 25px;
box-sizing:
border-box; }
```

All CSS Padding Properties

Property	Description
padding	A shorthand property for setting all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element

padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element

The CSS height and width properties are used to set the height and width of an element.

The CSS max-width property is used to set the maximum width of an element.

CSS Setting height and width

The height and width properties are used to set the height and width of an element.

The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

CSS height and width Values

The height and width properties may have the following values:

- auto - This is default. The browser calculates the height and width length - Defines the height/width in px, cm, etc.
- % - Defines the height/width in percent of the containing block
- initial - Sets the height/width to its default value
- inherit - The height/width will be inherited from its parent value

```
div {   height: 200px;
width: 50%;
background-color:
powderblue;
}
```

Example

Set the height and width of another <div> element:

```
div {   height: 100px;
width: 500px;
background-color:
powderblue; }
```

Setting max-width

The max-width property is used to set the maximum width of an element.

The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

The problem with the <div> above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.

Using max-width instead, in this situation, will improve the browser's handling of small

windows. The CSS Box Model

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 every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

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

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

Example

Demonstration of the box model:

```
div {   width:
300px;   border:
15px solid
green;   padding:
50px;   margin:
20px; }
```

CSS Outline

An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".

CSS has the following outline properties:

- outline-style
- outline-color
- outline-width outline-offset
- outline

CSS Outline Style

The outline-style property specifies the style of the outline, and can have one of the following values:

- dotted - Defines a dotted outline
- dashed - Defines a dashed outline
- solid - Defines a solid outline
- double - Defines a double outline
- groove - Defines a 3D grooved outline
- ridge - Defines a 3D ridged outline
- inset - Defines a 3D inset outline
- outset - Defines a 3D outset outline
- none - Defines no outline

- hidden - Defines a hidden outline

The following example shows the different outline-style values:

Example

Demonstration of the different outline styles:

```
p.dotted {outline-style: dotted;}
p.dashed {outline-style:
dashed;}
p.solid {outline-style: solid;}
p.double {outline-style:
double;}
p.groove {outline-style:
groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
```

CSS Outline Width

The outline-width property specifies the width of the outline, and can have one of the following values:

- thin (typically 1px)
- medium (typically 3px)
- thick (typically 5px)
- A specific size (in px, pt, cm, em, etc)

Example:

```
p.ex1 {
border: 1px
solid black;
outline-style:
solid;   outline-
color: red;
outline-width:
thin;
}

p.ex2 {
border: 1px
solid black;
outline-style:
solid;   outline-
color: red;
outline-width:
medium;
}

p.ex3 {
border: 1px
solid black;
outline-style:
solid;   outline-
color: red;
outline-width:
thick;
}

p.ex4 {
  border: 1px
solid black;
outline-style:
solid;   outline-
color: red;
outline-width:
4px; }
```

CSS Outline Color

The outline-color property is used to set the color of the outline.

The color can be set by:

- name - specify a color name, like "red"

- HEX - specify a hex value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
- invert - performs a color inversion (which ensures that the outline is visible, regardless of color background)

Example:

```
p.ex1 {
border: 2px
solid black;
outline-style:
solid;   outline-
color: red;
}

p.ex2 {
border: 2px
solid black;
outline-style:
dotted;
outline-color:
blue;
}

p.ex3 {
border: 2px
solid black;
outline-style:
outset;
```

outline-color:
grey; }

CSS Outline - Shorthand property

The outline property is a shorthand property for setting the following individual outline properties:

- outline-width
- outline-style (required)
- outline-color

The outline property is specified as one, two, or three values from the list above. The order of the values does not matter.

The following example shows some outlines specified with the shorthand outline property:

A dashed outline.

A dotted red outline.

A 5px solid yellow outline.

A thick ridge pink outline.

```
p.ex1 {outline: dashed;}
p.ex2 {outline: dotted red;}
p.ex3 {outline: 5px solid yellow;}
p.ex4 {outline: thick ridge pink;}
```

CSS Outline Offset

The outline-offset property adds space between an outline and the edge/border of an element. The space between an element and its outline is transparent.

Example:

```
p {   margin:
30px;   border:
1px solid
black;   outline:
1px solid red;
outline-offset:
15px; }
```

All CSS Outline Properties

Property	Description

outline	A shorthand property for setting outline-width, outline-style, and outline-color in one declaration
outline-color	Sets the color of an outline
outlineoffset	Specifies the space between an outline and the edge or border of an element
outline-style	Sets the style of an outline
outlinewidth	Sets the width of an outline

CSS Text

Text Color

The color property is used to set the color of the text. The color is specified by:

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

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

```
bod
y {
col
or:
blu
e;
}

h1 {
colo
r:
gree
n; }
```

Text Color and Background Color

In this example, we define both the background-color property and the color property:

```
body {   background-
color:        lightgrey;
color: blue;
}

h1                    {
background-
color:        black;
color: white;
}

div {
  background-
color: blue;
color: white; }
```

Text Alignment and Text Direction

In this chapter you will learn about the following properties:

- text-align
- text-align-last
- direction
- unicode-bidi
- vertical-align

Text Alignment

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

A text can be left or right aligned, centered, or justified.

The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

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

h2 {
text-
align:
left;
}

h3 {
text-
align:
right;
}
```

When the text-align property 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):

```
div {
text-
align:
justify; }
```

Text Align Last

The text-align-last property specifies how to align the last line

of a text.
```
p.a {   text-align-last: right;
}

p.b {   text-
align-last:
center;
}

p.c {   text-
align-last:
justify; }
```

Text Direction

The direction and unicode-bidi properties can be used to change the text direction of an element:

```
p {   direction: rtl;
unicode-bidi: bidi-
override; }
```

Vertical Alignment

The vertical-align property sets the vertical alignment of an element.

Example

Set the vertical alignment of an image in a text:

```
img.a {
vertical-align:
baseline; }

img.b {
vertical-align:
text-top;
}

img.c {   vertical-
align: text-bottom;
}

img.d {
vertical-
align: sub;
}
```

```
img.e {
vertical-
align: super;
}
```

The CSS Text Alignment/Direction Properties

Property	Description
direction	Specifies the text direction/writing direction
text-align	Specifies the horizontal alignment of text
text-align-last	Specifies how to align the last line of a text
unicode-bidi	Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document
vertical-align	Sets the vertical alignment of an element

Text Decoration

In this chapter you will learn about the following properties:

- text-decoration-line
- text-decoration-color
- text-decoration-style
- text-decoration-thickness
- text-decoration

Add a Decoration Line to Text

The text-decoration-line property is used to add a decoration line to text.

Tip: You can combine more than one value, like overline and underline to display lines both over and under a text.

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

h2 {   text-decoration-
line: line-through;
}

h3 {   text-decoration-
line: underline;
}

p   {          text-decoration-line:
overline underline; }
```

Specify a Color for the Decoration Line

The text-decoration-color property is used to set the color of the decoration line.

```
h1 {   text-decoration-
line:  overline;     text-
decoration-color: red;
}

h2 {
  text-decoration-line: line-
through;   text-decoration-color:
blue;
}

h3  {     text-decoration-
line:  underline;     text-
decoration-color:
green;
}

p   {          text-decoration-line:
overline   underline;          text-
decoration-color: purple; }
```

Specify a Style for the Decoration Line

The text-decoration-style property is used to set the style of the decoration line.

```
h1 {
  text-decoration-line: underline;
text-decoration-style: solid;
}
```

```
h2 {    text-decoration-
line: underline;    text-
decoration-style:
double;
}

h3 {    text-decoration-
line: underline;    text-
decoration-style:
dotted;
}

p.ex1 {
  text-decoration-line: underline;
text-decoration-style: dashed;
}

p.ex2    {              text-
decoration-line:
underline;              text-
decoration-style: wavy;
}

p.ex3 {
  text-decoration-line:
underline;              text-
decoration-color:   red;
text-decoration-style:
wavy; }
```

Specify the Thickness for the Decoration Line

The text-decoration-thickness property is used to set the thickness of the decoration line.

```
h1 {
  text-decoration-line: underline;
text-decoration-thickness: auto;
}

h2 {    text-decoration-
line: underline;    text-
decoration-thickness:
5px;
}

h3 {
  text-decoration-line: underline;
text-decoration-thickness: 25%;
}

p {   text-decoration-
line: underline;   text-
decoration-color: red;
text-decoration-style:
double;   text-
decoration-thickness:
5px; }
```

The Shorthand Property

The text-decoration property is a shorthand property for:

- text-decoration-line (required)
- text-decoration-color (optional)
- text-decoration-style (optional)
- text-decoration-thickness (optional)

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

h2 {   text-decoration:
underline red;
}

h3 {   text-decoration:
underline red double;
}

p {
  text-decoration: underline red
double 5px; }
```

All CSS text-decoration Properties

Property	Description
text-decoration	Sets all the text-decoration properties in one declaration

text-decoration-color	Specifies the color of the text-decoration
text-decoration-line	Specifies the kind of text decoration to be used (underline, overline, etc.)
text-decoration-style	Specifies the style of the text decoration (solid, dotted, etc.)
text-decoration-thickness	Specifies the thickness of the text decoration line

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:

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

p.lowercase {
text-transform:
lowercase;
}

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

Text Spacing

In this chapter you will learn about the following properties:

- text-indent
- letter-spacing
- line-height
- word-spacing
- white-space

Text Indentation

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

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

Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text. The following example demonstrates how to increase or decrease the space between characters:

```
h1 {
  letter-spacing: 5px;
}

h2 {   letter-
spacing: -
2px; }
```

Line Height

The line-height property is used to specify the space between lines:

```
p.small {
line-height:
0.8;
}

p.big {
line-
height:
1.8; }
```

Word Spacing

The word-spacing property is used to specify the space between the words in a text. The following example demonstrates how to increase or decrease the space between words: p.one {   word-spacing: 10px;

}

```
p.two {
word-
spacing: -
2px; }
```

White Space

The white-space property specifies how white-space inside an element is handled.

This example demonstrates how to disable text wrapping inside an element:

```
p {   white-
space:
nowrap; }
```

The CSS Text Spacing Properties

Property	Description
letter-spacing	Specifies the space between characters in a text
line-height	Specifies the line height
text-indent	Specifies the indentation of the first line in a text-block
white-space	Specifies how to handle whitespace inside an element
word-spacing	Specifies the space between words in a text

Text Shadow

The text-shadow property adds shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

Text shadow
effect! h1 {
text-shadow:
2px 2px; }

Text shadow effect!

```
h1 { text-
shadow: 2px 2px
red; }
```

Font Selection is Important

Choosing the right font has a huge impact on how the readers experience a website.

The right font can create a strong identity for your brand.

Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

Example

Specify some different fonts for three paragraphs:

```
.p1 {
  font-family: "Times New Roman", Times, serif;
}
```

```
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}

.p3 {   font-family: "Lucida Console", "Courier
New", monospace;
}

p { font-family: Tahoma,
Verdana, sans-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)

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

}

p.italic {
font-style:
italic;
}

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

Font Weight

The font-weight property specifies the weight of a font:

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

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

Font Variant

The font-variant property specifies whether or not a text should be displayed in a small-caps font.

In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text. p.normal

```
{   font-variant: normal;
}

p.small {   font-
variant: small-
caps; }
```

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

Set Font Size With Pixels

Setting the text size with pixels gives you full control over the text size:

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

h2 {
font-
size:
30px;
}

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

Set Font Size With Em

To allow users to resize the text (in the browser menu), many developers use em instead of pixels.

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

```
h1 {   font-size: 2.5em; /*
40px/16=2.5em */
}

h2 {
  font-size: 1.875em; /* 30px/16=1.875em */
}

p {   font-size: 0.875em; /*
14px/16=0.875em */ }
```

Use a Combination of Percent and Em

The solution that works in all browsers, is to set a default font-size in percent for the <body> element:

```
body {
font-
size:
100%;
}

h1 {
  font-size: 2.5em;
}

h2 {
  font-size: 1.875em;
}

p {
  font-size:
0.875em; }
```

How To Use Google Fonts

Just add a special style sheet link in the <head> section and then refer to the font in the CSS.

Example

Here, we want to use a font named "Sofia" from Google Fonts:

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<st
yle
>
bod
y {
  font-family: "Sofia", sans-serif;
}
</style
>
</head
>
```

Use Multiple Google Fonts

To use multiple Google fonts, just separate the font names with a pipe character (|), like this:

Example

Request multiple fonts:

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Audiowide|Sofia|Trirong">
<style> h1.a {font-family:
"Audiowide", sans-serif;} h1.b
{font-family: "Sofia", sans-serif;}
h1.c {font-family: "Trirong",
serif;}
</style
>
</head
>
```

Styling Google Fonts

Of course you can style Google Fonts as you like, with CSS!

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<st
yle
>
bod
y {
  font-family: "Sofia", sans-serif;
font-size: 30px;
  text-shadow: 3px 3px 3px #ababab;
}
</style
>
</head
>
```

Exampl
e

Add multiple effects to the "Sofia" font:

```
<head>
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia&effect=neon|outline|emboss|shad ow-multiple">
<style> body {   font-family: "Sofia", sans-serif;   font-size: 30px;
}
</style
>
</head
>
<body>

<h1 class="font-effect-neon">Neon Effect</h1>
<h1 class="font-effect-outline">Outline Effect</h1>
<h1 class="font-effect-emboss">Emboss Effect</h1>
<h1 class="font-effect-shadow-multiple">Multiple Shadow Effect</h1>

</body>
```

Font Pairing Rules

Here are some basic rules to create great font pairings:

1. Complement

It is always safe to find font pairings that complement one another.

A great font combination should harmonize, without being too similar or too different.

2. Use Font Superfamilies

A font superfamily is a set of fonts designed to work well together. So, using different fonts within the same superfamily is safe.

For example, the Lucida superfamily contains the following fonts: Lucida Sans, Lucida Serif, Lucida Typewriter Sans, Lucida Typewriter Serif and Lucida Math.

3. Contrast is King

Two fonts that are too similar will often conflict. However, contrasts, done the right way, brings out the best in each font.

Example: Combining serif with sans serif is a well known combination.

A strong superfamily includes both serif and sans serif variations of the same font (e.g. Lucida and Lucida Sans).

4. Choose Only One Boss

One font should be the boss. This establishes a hierarchy for the fonts on your page. This can be achieved by varying the size, weight and color.

```
body  {      background-
color:  black;        font-
family:  Verdana,  sans-
serif;     font-size:  16px;
color: gray;
}
```

```
h1 {   font-family:
Georgia, serif;
font-size: 60px;
color: white; }
```

The CSS Font Property

To shorten the code, it is also possible to specify all the individual font properties in one property.

The font property is a shorthand property for:

- font-style
- font-variant
- font-weight
- font-size/line-height
- font-family

Note: The font-size and font-family values are required. If one of the other values is missing, their default value are used.

Example

Use font to set several font properties in one

declaration: p.a {   font: 20px Arial, sans-serif;

}

```
p.b {   font: italic small-caps bold 12px/30px
Georgia, serif; }
```

How To Add Icons

The simplest way to add an icon to your HTML page, is with an icon library, such as Font Awesome.

Add the name of the specified icon class to any inline HTML element (like <i> or <span>).

All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.)

Font Awesome Icons

To use the Font Awesome icons, go to fontawesome.com, sign in, and get a code to add in the <head> section of your HTML page:

```
<script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>
```

```
<!DOCTYPE html>
<html>
<head>
<script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script>
</head>
<body>
```

```
<i class="fas fa-cloud"></i>
<i class="fas fa-heart"></i>
<i class="fas fa-car"></i>
<i class="fas fa-file"></i>
<i class="fas fa-bars"></i>

</body>
</html>
```

Bootstrap Icons

To use the Bootstrap icons, add the following line inside the <head> section of your HTML page:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body>

<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
<i class="glyphicon glyphicon-user"></i>
<i class="glyphicon glyphicon-envelope"></i>
<i class="glyphicon glyphicon-thumbs-up"></i>

</body>
</html>
```

Google Icons

To use the Google icons, add the following line inside the <head> section of your HTML page:

```
<link rel="stylesheet"

href="https://fonts.googleapis.com/icon?family=Material+Icons">
```
Note: No

downloading or installation is required!

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
<body>

<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>
```

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

CSS Links

With CSS, links can be styled in many different ways.

Text Link Text Link Link Button Link Button

Styling Links

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

```
a {
color:
hotpin
k; }
```

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

```
/*
unvisited
link */
a:link {
color:
red;
}


/*
visited
link */
a:visite
d {
color:
green;
}

/* mouse
over link */
a:hover {
color:
hotpink;
}
```

```
/*
selected
link */
a:active
{
color:
blue; }
```

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:

```
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 can be used to specify a background color for links:

```
a:link {
  background-color: yellow;
}

a:visited {
background-color:
cyan;
}

a:hover {
  background-color: lightgreen;
}
```

```
a:active {
background-color:
hotpink; }
```

Link Buttons

This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes/buttons:

```
a:link, a:visited {
background-color:
#f44336;   color:
white;   padding:
14px 25px;   text-
align: center;   text-
decoration: none;
display: inline-
block;
}

a:hover,
a:active         {
background-
color: red; }
```

Example

This example demonstrates how to add other styles to hyperlinks:

```
a.one:link {color: #ff0000;}
a.one:visited {color: #0000ff;}
a.one:hover {color: #ffcc00;}

a.two:link {color: #ff0000;}
a.two:visited {color: #0000ff;}
a.two:hover {font-size: 150%;}

a.three:link {color: #ff0000;}
a.three:visited {color: #0000ff;}
a.three:hover {background: #66ff66;}

a.four:link {color: #ff0000;}
a.four:visited {color: #0000ff;}
a.four:hover {font-family: monospace;}

a.five:link {color: #ff0000; text-decoration: none;}
a.five:visited {color: #0000ff; text-decoration: none;}
a.five:hover {text-decoration:
underline;}
```