CENTER OF ADVANCED COMPUTER LEARNING AND DEVELOPMENT

WEB DEVELOPMENT COURSE MATERIAL

CSS (CASCADING STYLE SHEETS) Study Material

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 does HTML need CSS?

HTML uses “tags” that have some form of meaning. A headline tag says that this is a headline for what follows. A paragraph tag says that this text forms a unified thought. A table header tag says that this cell is a header for a row or column. Cascading style sheets provide no meaning. They allow you to change how an HTML tag displays, but not what an HTML tag means. In general, when you create web pages using HTML and CSS, the HTML should describe your web page, and your CSS should describe its presentation. The page should remain readable in a wide variety of browsers and browser types without any of the CSS.

How To Add CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External Stylesheet

An external CSS stylesheet can be applied to any number of HTML documents by placing a element in each HTML document. The attribute rel of the tag has to be set to "stylesheet", and the href attribute to the relative or absolute path to the stylesheet. While using relative URL paths is generally considered good practice, absolute paths can be used, too. In HTML5 the type attribute can be omitted. It is recommended that the tag be placed in the HTML file's tag so that the styles are loaded before the elements they style. Otherwise, users will see a flash of unstyled content.

Example: hello-world.html

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<h1>Hello world!</h1>

<p>I ♥ CSS</p>

</body>
```

```
</html>
```

style.css

```
h1 {

color: green;

text-decoration: underline;

}

p {

font-size: 25px;

font-family: 'Trebuchet MS', sans-serif;

}
```

Make sure you include the correct path to your CSS file in the href. If the CSS file is in the same folder as your HTML file then no path is required (like the example above) but if it's saved in a folder, then specify it like this href="foldername/style.css".

External stylesheets are considered the best way to handle your CSS. There's a very simple reason for this: when you're managing a site of, say, 100 pages, all controlled by a single stylesheet, and you want to change your link colors from blue to green, it's a lot easier to make the change in your CSS file and let the changes "cascade" throughout all 100 pages than it is to go into 100 separate pages and make the same change 100 times. Again, if you want to completely change the look of your website, you only need to update this one file. You can load as many CSS files in your HTML page as needed. CSS rules are applied with some basic rules, and order does matter. For example, if you have a main.css file with some code in it:

```
p.green { color: #00FF00; }
```

All your paragraphs with the 'green' class will be written in light green, but you can override this with another .css file just by including it after main.css. You can have override.css with the following code follow main.css, for example:

```
p.green { color: #006600; }
```

Now all your paragraphs with the 'green' class will be written in darker green rather than light green. Other principles apply, such as the '!important' rule, specificity, and inheritance. https://w3schools.com.com When someone first visits your website, their browser downloads the HTML of the current page plus the linked CSS file. Then when they navigate to another page, their browser only needs to download the HTML of that page; the CSS file is cached, so it does not need to be downloaded again. Since browsers cache the external stylesheet, your pages load faster.

Internal Styles

CSS enclosed in <style></style> tags within an HTML document functions like an external

stylesheet, except that it lives in the HTML document it styles instead of in a separate file, and

therefore can only be applied to the document in which it lives. Note that this element must be

inside the <head> element for HTML validation (though it will work in all current browsers if placed

in body).

```
<head>

 <style>

 h1 {
```

```
 color: green;

 text-decoration: underline;

 }

 p {

 font-size: 25px;

 font-family: 'Trebuchet MS', sans-serif;

 }

 </style>

</head>

<body>

 <h1>Hello world!</h1>

 <p>I ♥ CSS</p>

</body>
```

Inline Styles

Use inline styles to apply styling to a specific element. Note that this is not optimal. Placing style

rules in a <style> tag or external CSS file is encouraged in order to maintain a distinction between

content and presentation.

Inline styles override any CSS in a <style> tag or external style sheet. While this can be useful in

some circumstances, this fact more often than not reduces a project's maintainability.

The styles in the following example apply directly to the elements to which they are attached.

```
<h1 style="color: green; text-decoration: underline;">Hello world!</h1>

<p style="font-size: 25px; font-family: 'Trebuchet MS';">I ♥ CSS</p>
```

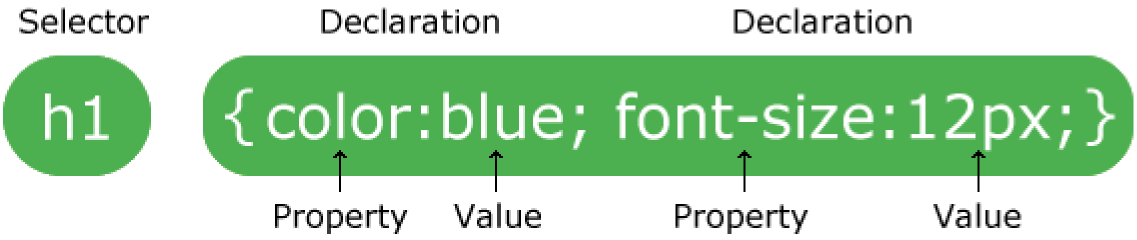
Inline styles are generally the safest way to ensure rendering compatibility across various email

clients, programs and devices, but can be time-consuming to write and a bit challenging to

manage.

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 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.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>


<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>


</body>
</html>
```

Example Explained

- **p is a selector in CSS (it points to the HTML element you want to style: <p>).**
- **color is a property, and red is the property value**
- **text-align is a property, and center is the property value**

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 `*/`:

Example

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

CSS Colors

CSS Background Color

You can set the background color for HTML elements:

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

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

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam
erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper
suscipit lobortis nisl ut aliquip ex ea commodo consequat.
</p>

</body>
</html>
```

CSS Text Color

You can set the color of text:

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

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

Hello World

Example

```
<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 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.

Example

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

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the background!</p>

</body>
```

```
</html>
```

CSS Borders

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**

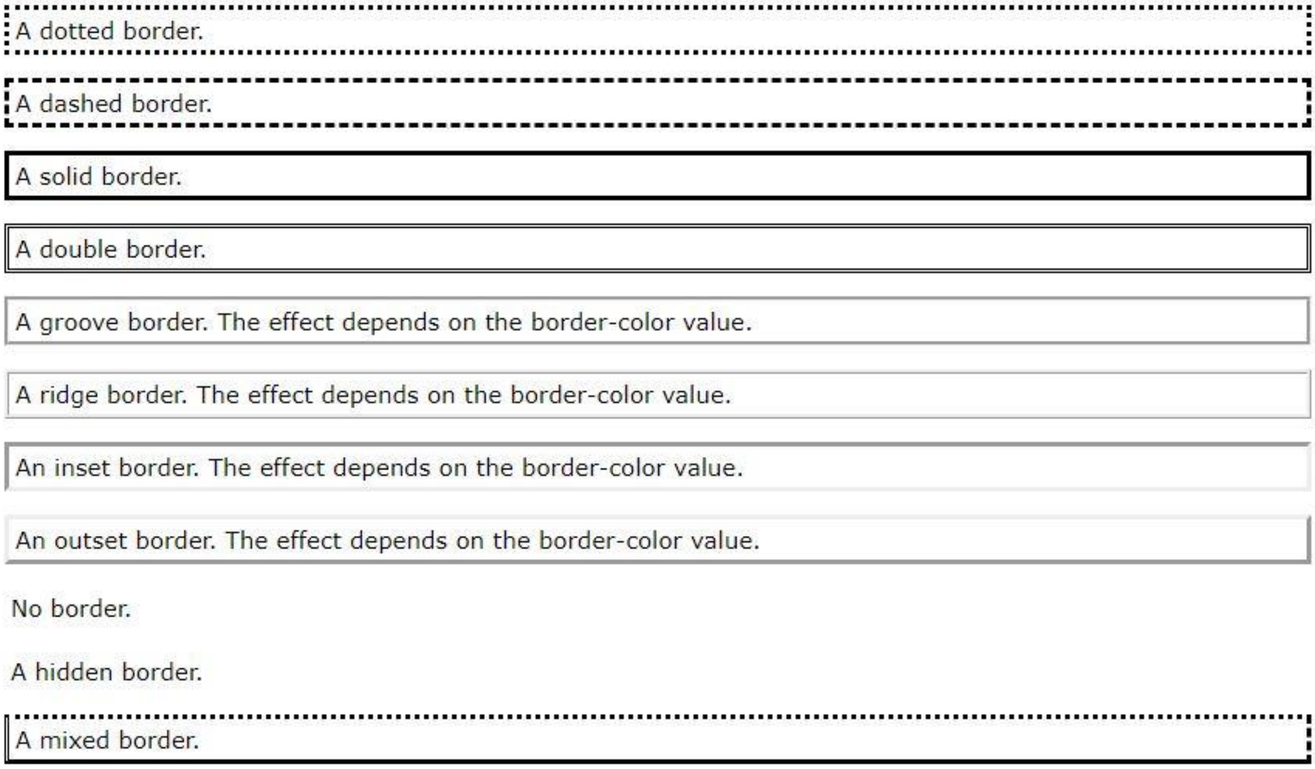
The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example

Demonstration of the different border styles:

```
<!DOCTYPE html>
<html>
<head>
<style>
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;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>

</body>
</html>
```



A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

CSS Border Width

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

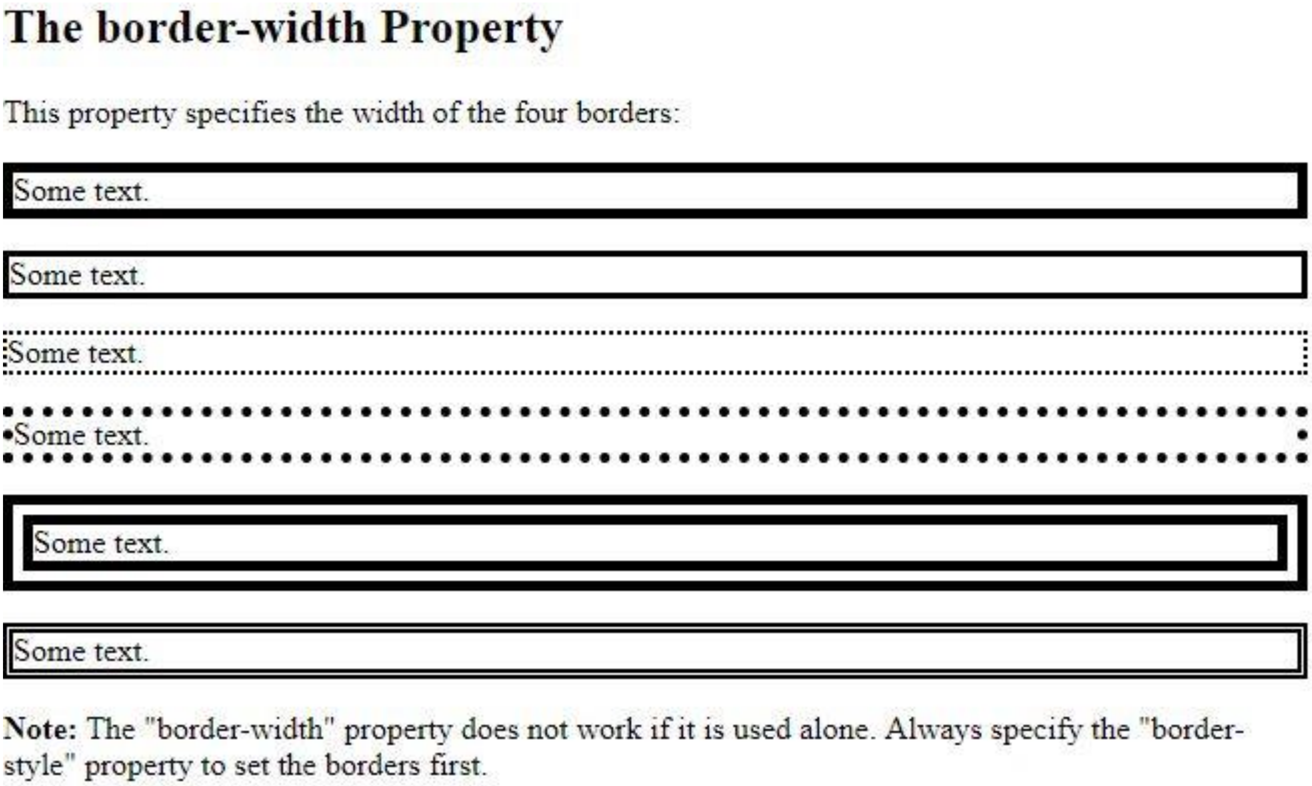
The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick:

Example

Demonstration of the different border widths:

```
<!DOCTYPE html>
<html>
<head>
<style>
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;
}

p.five {
  border-style: double;
  border-width: 15px;
}

p.six {
  border-style: double;
  border-width: thick;
}
</style>
</head>
<body>

<h2>The border-width Property</h2>
<p>This property specifies the width of the four borders:</p>

<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>
<p class="four">Some text.</p>
<p class="five">Some text.</p>
<p class="six">Some text.</p>
```

```
<p><b>Note:</b> The "border-width" property does not work if it is
used alone.
Always specify the "border-style" property to set the borders
first.</p>

</body>
</html>
```



The border-width Property

This property specifies the width of the four borders:

Some text.

Some text.

Some text.

Some text.

Some text.

Some text.

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

CSS Border Color

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

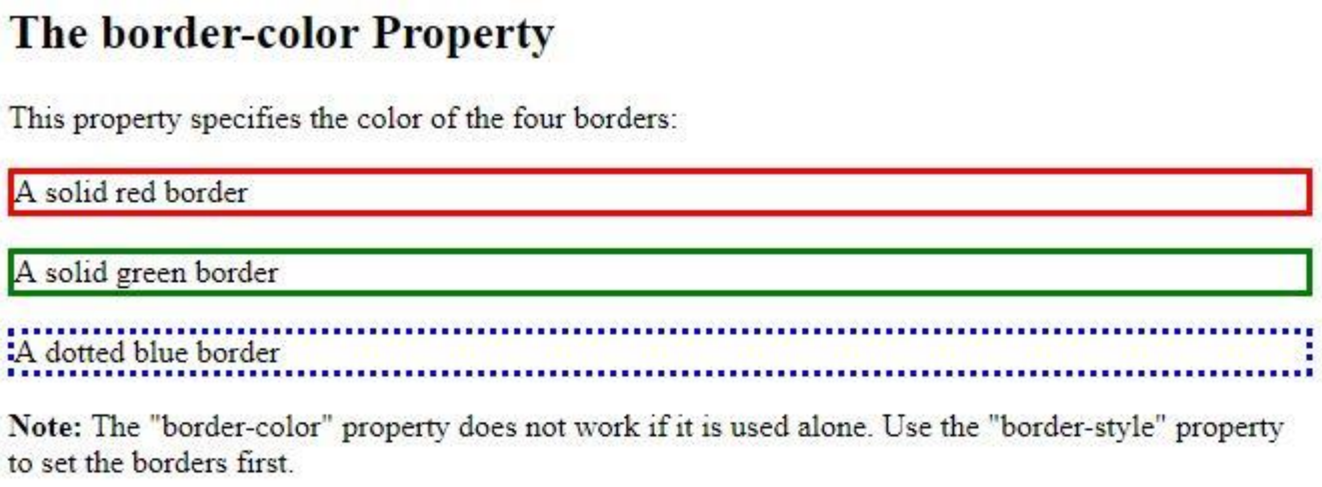
The color can be set by:

- **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**

Example

Demonstration of the different border colors:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-color: red;
}

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

p.three {
  border-style: dotted;
  border-color: blue;
}
</style>
</head>
<body>

<h2>The border-color Property</h2>
<p>This property specifies the color of the four borders:</p>

<p class="one">A solid red border</p>
<p class="two">A solid green border</p>
<p class="three">A dotted blue border</p>

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

</body>
</html>
```



The border-color Property

This property specifies the color of the four borders:

A solid red border

A solid green border

A dotted blue border

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

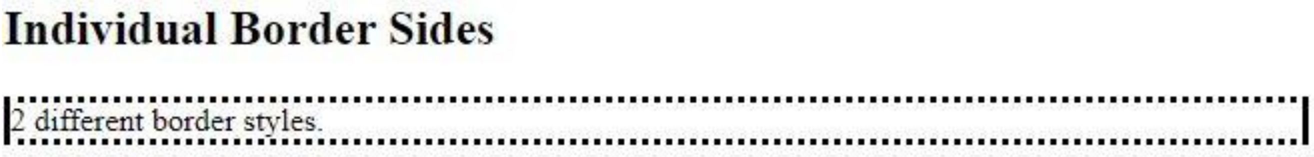
CSS Border - Individual Sides

From the examples on the previous pages, you have seen that it is possible to specify a different border for each side.

In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left):

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;
}
</style>
</head>
<body>

<h2>Individual Border Sides</h2>
<p>2 different border styles.</p>

</body>
```

```
</html>
```



Individual Border Sides

2 different border styles.

CSS Margins

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

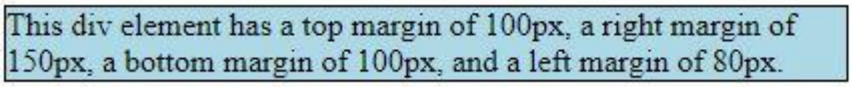
With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Example

Set different margins for all four sides of a <p> element:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>Using individual margin properties</h2>
```

```
<div>This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.</div>

</body>
</html>
```



This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

CSS Padding

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

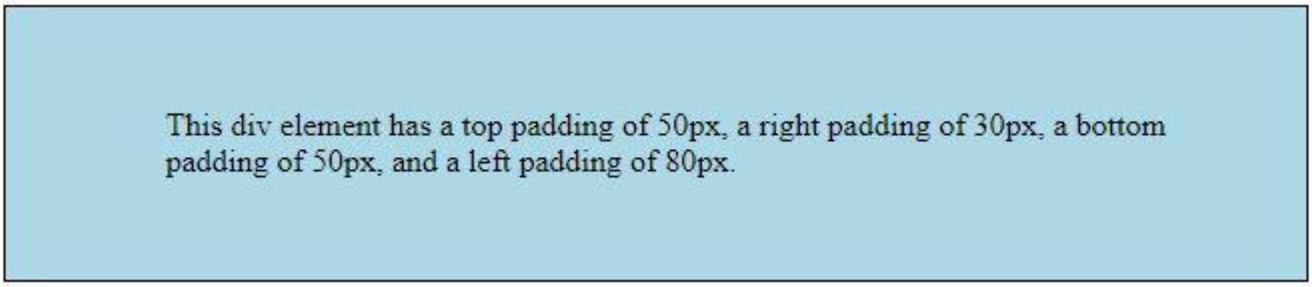
With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Example

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

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
</head>
<body>
```

```
<h2>Using individual padding properties</h2>

<div>This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.</div>

</body>
</html>
```



This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

CSS Box Model

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.

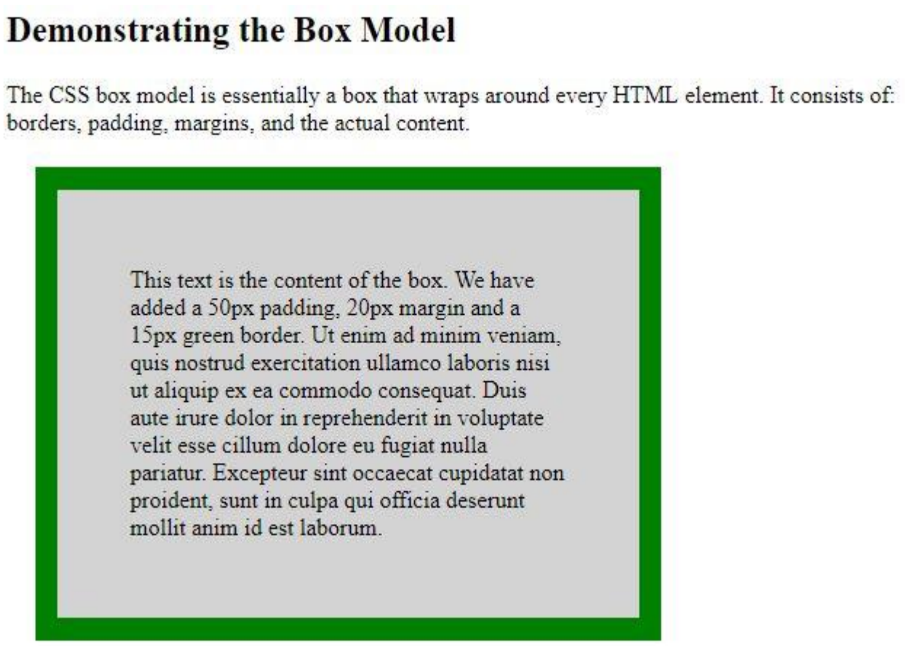
The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

Example

Demonstration of the box model:

```
<!DOCTYPE html>

<html>

<head>

<style>

div {
  background-color: lightgrey;

  width: 300px;

  border: 15px solid green;

  padding: 50px;

  margin: 20px;

}

</style>

</head>

<body>


<h2>Demonstrating the Box Model</h2>


<p>The CSS box model is essentially a box that wraps around every
HTML element. It consists of: borders, padding, margins, and the
actual content.</p>


<div>This text is the content of the box. We have added a 50px
padding, 20px margin and a 15px green border. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur
sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.</div>
```

```
</body>

</html>
```



Demonstrating the Box Model

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

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CSS Text

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)"**

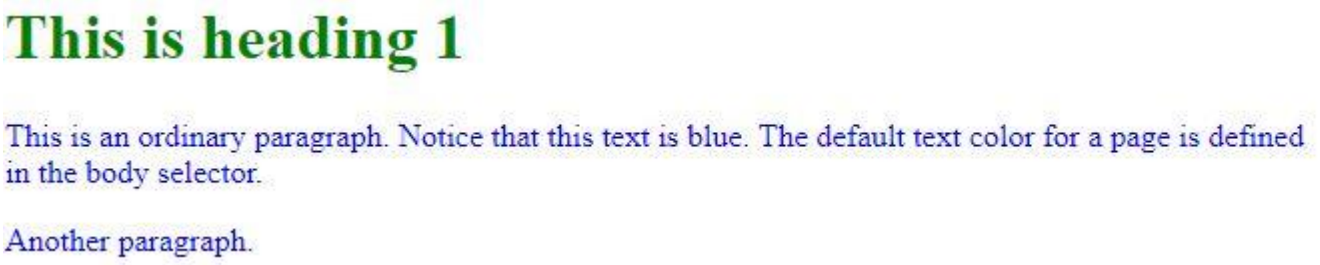
Look at CSS Color Values for a complete list of possible color values.

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

Example

```
<!DOCTYPE html>

<html>

<head>

<style>
```

```
body {
  color: blue;
}

h1 {
  color: green;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<p>This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.</p>
<p>Another paragraph.</p>

</body>
</html>
```



This is heading 1

This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.

Another paragraph.

Font Family

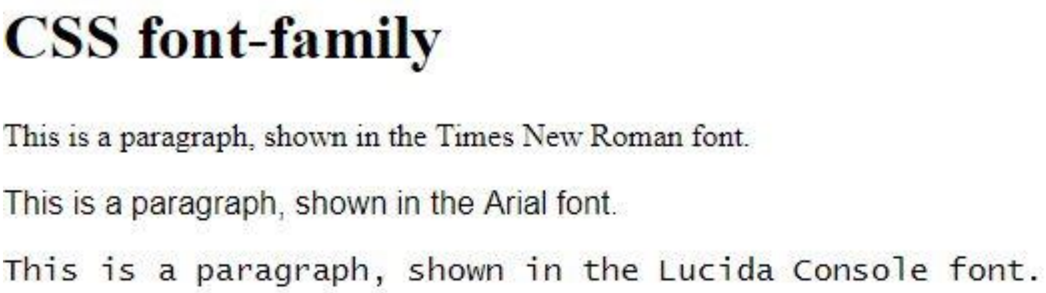
The font family of a text is set with the font-family property.

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, and so on.

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.

```
<!DOCTYPE html>
<html>
<head>
<style>
.serif {
  font-family: "Times New Roman", Times, serif;
}

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

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

```
</style>

</head>

<body>



<h1>CSS font-family</h1>

<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>

<p class="sansserif">This is a paragraph, shown in the Arial font.</p>

<p class="monospace">This is a paragraph, shown in the Lucida Console font.</p>



</body>

</html>
```



CSS font-family

This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.

This is a paragraph, shown in the Lucida Console font.

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

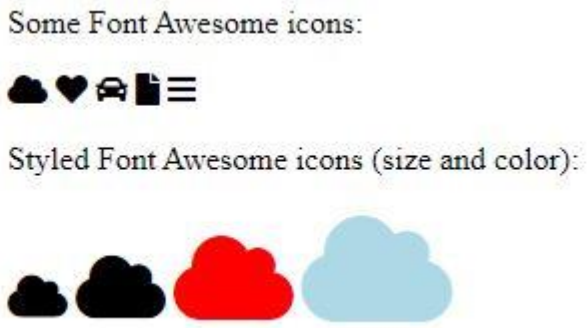
To use the Font Awesome icons, go to fontawesome.com, sign in, and get a code to add in the `<head>` section of your HTML page:

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

Read more about how to get started with Font Awesome in our Font Awesome 5 tutorial.

```
<!DOCTYPE html>

<html>

<head>

<title>Font Awesome Icons</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="https://kit.fontawesome.com/a076d05399.js"></script>

<!--Get your own code at fontawesome.com-->

</head>

<body>



<p>Some Font Awesome icons:</p>

<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>



<p>Styled Font Awesome icons (size and color):</p>

<i class="fas fa-cloud" style="font-size:24px;"></i>

<i class="fas fa-cloud" style="font-size:36px;"></i>

<i class="fas fa-cloud" style="font-size:48px;color:red;"></i>

<i class="fas fa-cloud" style="font-size:60px;color:lightblue;"></i>



</body>

</html>
```



Some Font Awesome icons:

Styled Font Awesome icons (size and color):

Styling Links

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

Example

```
a {

  color: hotpink;

}
```

Example

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

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

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

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

Different List Item Markers

The list-style-type property specifies the type of list item marker.

The following example shows some of the available list item markers:

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

The list-style-image property specifies an image as the list item marker:

Example

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

Styling List With Colors

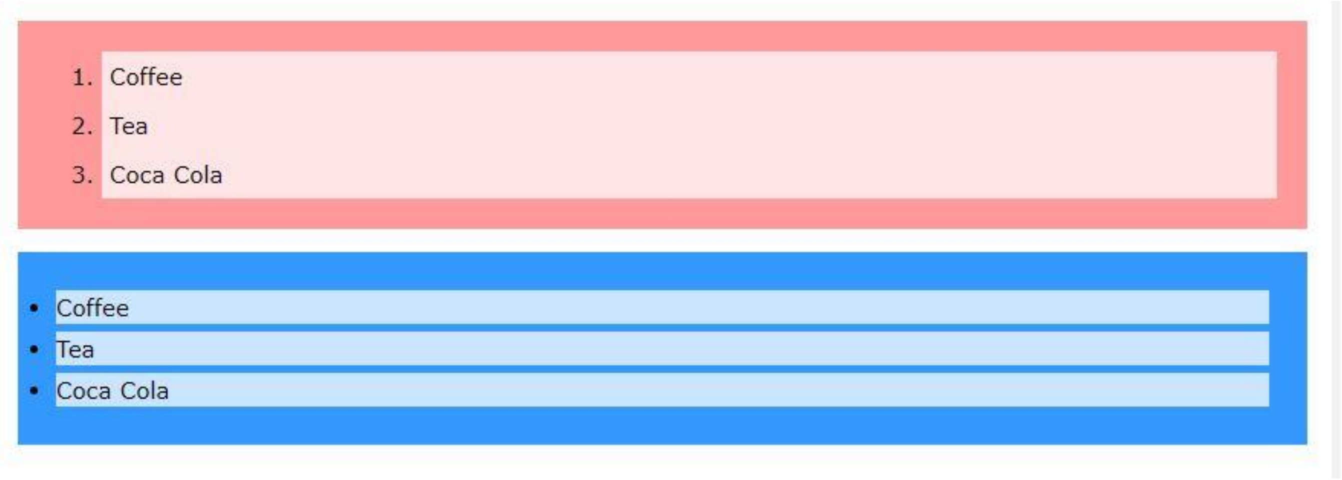
We can also style lists with colors, to make them look a little more interesting.

Anything added to the <ol> or <ul> tag, affects the entire list, while properties added to the <li> tag will affect the individual list items:

Example

```
ol {
  background: #ff9999;
  padding: 20px;
}
```

```
ul {
  background: #3399ff;
  padding: 20px;
}

ol li {
  background: #ffe5e5;
  padding: 5px;
  margin-left: 35px;
}

ul li {
  background: #cce5ff;
  margin: 5px;
}
```



1. Coffee
2. Tea
3. Coca Cola

- Coffee
- Tea
- Coca Cola

CSS Overflow

The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

The overflow property has the following values:

- visible - Default. The overflow is not clipped. The content renders outside the element's box

- hidden - The overflow is clipped, and the rest of the content will be invisible
- scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
- auto - Similar to scroll, but it adds scrollbars only when necessary

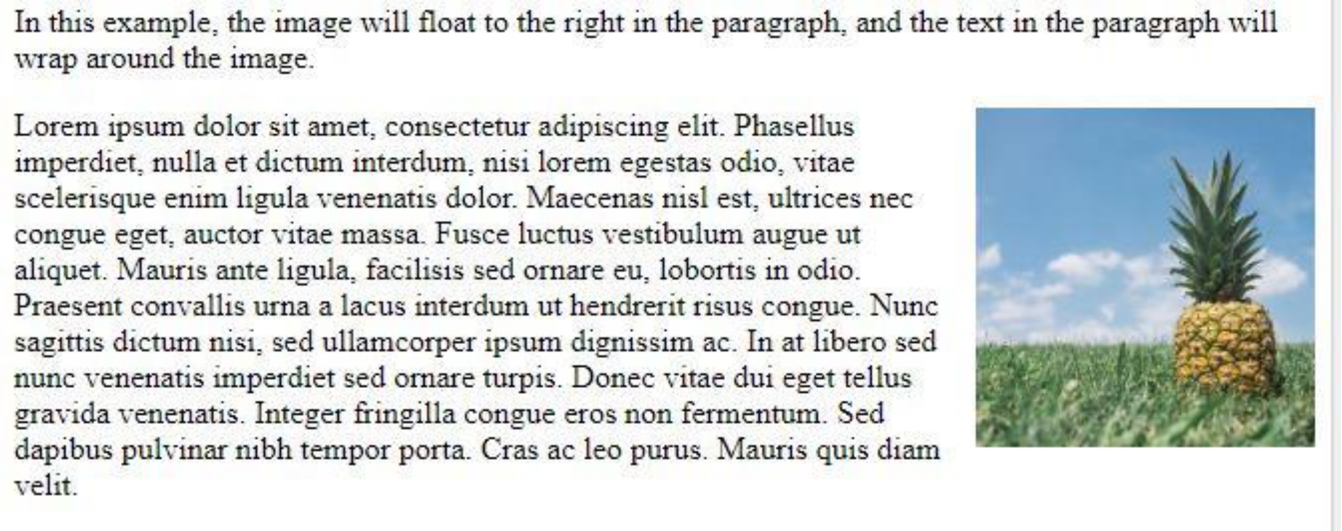
Example

```
div {
  width: 200px;
  height: 50px;
  background-color: #eee;
  overflow: visible;
}
```

overflow:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>

<p>In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.</p>

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;height:170px;margin-left:15px;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue
```

```
ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>
```



In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

CSS Layout - display: inline-block

Compared to `display: inline`, the major difference is that `display: inline-block` allows to set a width and height on the element.

Also, with `display: inline-block`, the top and bottom margins/paddings are respected, but with `display: inline` they are not.

Compared to `display: block`, the major difference is that `display: inline-block` does not add a line-break after the element, so the element can sit next to other elements.

The display: inline-block Value

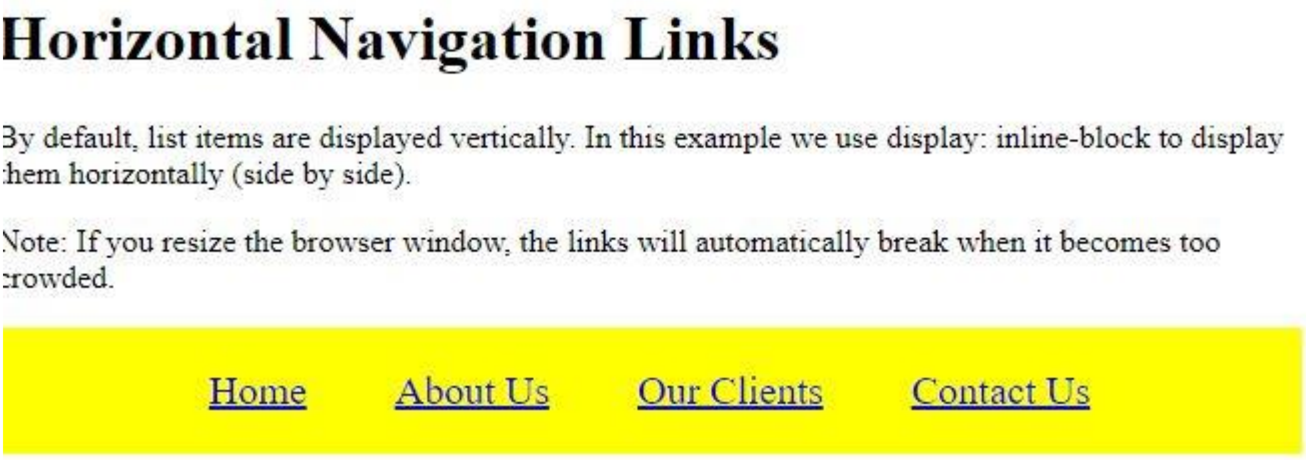
```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
span.a {
  display: inline; /* the default for span */
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}

span.b {
  display: inline-block;
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}

span.c {
  display: block;
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}
</style>
</head>
<body>

<h1>The display Property</h1>
```

```
<h2>display: inline</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vestibulum consequat scelerisque elit sit amet consequat. Aliquam
erat volutpat. <span class="a">Aliquam</span> <span
class="a">venenatis</span> gravida nisl sit amet facilisis. Nullam
cursus fermentum velit sed laoreet. </div>

<h2>display: inline-block</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vestibulum consequat scelerisque elit sit amet consequat. Aliquam
erat volutpat. <span class="b">Aliquam</span> <span
class="b">venenatis</span> gravida nisl sit amet facilisis. Nullam
cursus fermentum velit sed laoreet. </div>

<h2>display: block</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vestibulum consequat scelerisque elit sit amet consequat. Aliquam
erat volutpat. <span class="c">Aliquam</span> <span
class="c">venenatis</span> gravida nisl sit amet facilisis. Nullam
cursus fermentum velit sed laoreet. </div>

</body>
</html>
```

Using inline-block to Create Navigation Links

```
<!DOCTYPE html>
<html>
<head>
<style>
.nav {
  background-color: yellow;
  list-style-type: none;
  text-align: center;
  margin: 0;
  padding: 0;
}

.nav li {
  display: inline-block;
  font-size: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<h1>Horizontal Navigation Links</h1>
<p>By default, list items are displayed vertically. In this example we
use display: inline-block to display them horizontally (side by
side).</p>
<p>Note: If you resize the browser window, the links will
automatically break when it becomes too crowded.</p>

<ul class="nav">
  <li><a href="#home">Home</a></li>
  <li><a href="#about">About Us</a></li>
  <li><a href="#clients">Our Clients</a></li>
  <li><a href="#contact">Contact Us</a></li>
</ul>

</body>
</html>
```



Horizontal Navigation Links

3y default, list items are displayed vertically. In this example we use display: inline-block to display :hem horizontally (side by side).

Note: If you resize the browser window, the links will automatically break when it becomes too :rowded.

CSS Pseudo-elements

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- **Style the first letter, or line, of an element**
- **Insert content before, or after, the content of an element**

Syntax

The syntax of pseudo-elements:

```
selector::pseudo-element {
  property: value;
}
```

The ::first-line Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all <p> elements:

Example

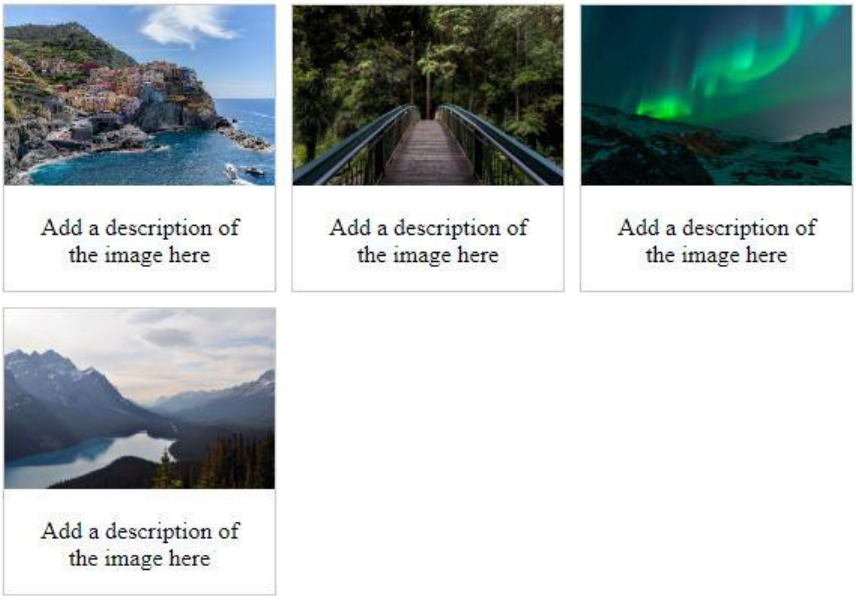
```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}
</style>
</head>
<body>

<p>You can use the ::first-line pseudo-element to add a special effect to the first line of a text. Some more text. And even more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more.</p>

</body>
</html>
```

CSS Image Gallery

```
<!DOCTYPE html>
<html>
<head>
<style>
div.gallery {
  margin: 5px;
  border: 1px solid #ccc;
  float: left;
  width: 180px;
}

div.gallery:hover {
  border: 1px solid #777;
}

div.gallery img {
  width: 100%;
  height: auto;
}

div.desc {
  padding: 15px;
  text-align: center;
}
</style>
</head>
<body>

<div class="gallery">
  <a target="_blank" href="img_5terre.jpg">
    <img src="img_5terre.jpg" alt="Cinque Terre" width="600"
height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_forest.jpg">
    <img src="img_forest.jpg" alt="Forest" width="600" height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_lights.jpg">
    <img src="img_lights.jpg" alt="Northern Lights" width="600"
height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>
```

```
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="img_mountains.jpg" alt="Mountains" width="600"
height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

</body>
</html>
```



Add a description of the image here

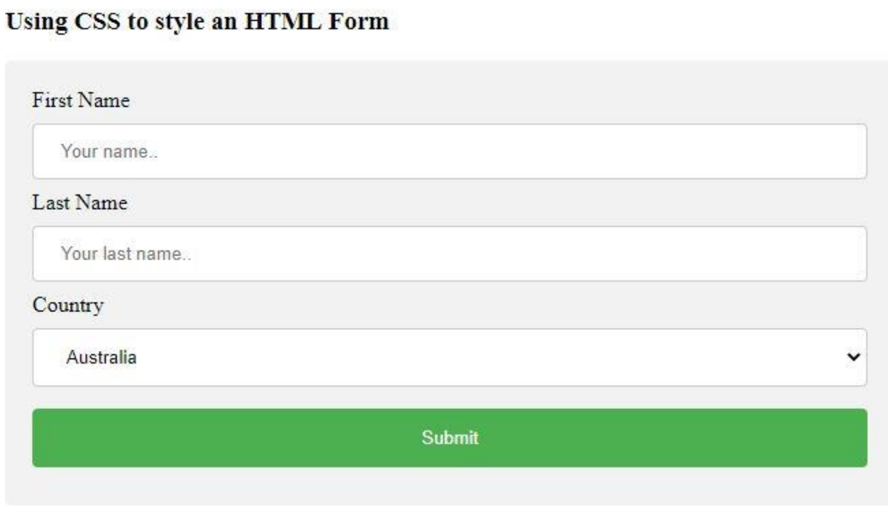
Add a description of the image here

Add a description of the image here

Add a description of the image here

CSS Forms

```
<!DOCTYPE html>
<html>
<style>
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type=submit]:hover {
  background-color: #45a049;
}

div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}
</style>
<body>

<h3>Using CSS to style an HTML Form</h3>

<div>
  <form action="/action_page.php">
    <label for="fname">First Name</label>
    <input type="text" id="fname" name="firstname"
placeholder="Your name..">

    <label for="lname">Last Name</label>
    <input type="text" id="lname" name="lastname"
placeholder="Your last name..">

    <label for="country">Country</label>
    <select id="country" name="country">
      <option value="australia">Australia</option>
      <option value="canada">Canada</option>
      <option value="usa">USA</option>
    </select>

    <input type="submit" value="Submit">
  </form>
</div>

</body>
</html>
```



Using CSS to style an HTML Form

First Name

Your name..

Last Name

Your last name..

Country

Australia

The @keyframes Rule

When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

The following example binds the "example" animation to the <div> element. The animation will last for 4 seconds, and it will gradually change the background-color of the <div> element from "red" to "yellow":

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
</style>
</head>
<body>

<p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>

<div></div>

<p><b>Note:</b> When an animation is finished, it changes back to its original style.</p>

</body>
```