**CSS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TOPICS TITLE** | **PAGE NO.** |
| **1** | **WHAT IS CSS** |  |
| **2** | **WHY USE CSS** |  |
| **3** | **HOW CSS WORKS** |  |
| **4** | **CASCADING AND INHERITANCE** |  |
| **5** | **SELECTORS** |  |
| **6** | **READABLE CSS** |  |
| **7** | **TEXT STYLES** |  |
| **8** | **COLOR** |  |
| **9** | **CONTENT** |  |
| **10** | **LISTS** |  |
| **11** | **BOXES** |  |
| **12** | **LAYOUT** |  |
| **13** | **TABLES** |  |
| **14** | **MEDIA** |  |

**WHAT IS CSS**

Cascading Style Sheets () is a language for specifying how documents are presented to users. These documents are written in a markup language such as HTML

A document is a collection of information that is structured using a markup language.

Presenting a document to a user means converting it into a usable form for your audience. Browsers, like Firefox, Chrome or Internet Explorer, are designed to present documents visually, for example, on a computer screen, projector or printer.

**CSS** is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. **CSS** is independent of HTML and can be used with any XML-based markup language.

## **Action: Creating a document**

1. Create a new directory on your computer to save and organize the tutorial exercises.
2. Open your text editor and create a new text file. This file will contain the document for the next few tutorial exercises.
3. Copy and paste the HTML shown below. Save the file using the name doc1.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sample document</title>

</head>

<body>

<p>

<strong>C</strong>ascading

<strong>S</strong>tyle

<strong>S</strong>heets

</p>

</body>

</html>

4.Open a new tab or a new window in your browser, then open the file you just created.

You should see the text with the initial letters bold, like this:

|  |
| --- |
| Cascading Style Sheets |

What you see in your browser might not look exactly the same because of settings in your browser and in this wiki. Some differences in the font, spacing and colors are not important.

**CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

**Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

**Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

**Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing

**Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

**Offline Browsing** − CSS can store web applications locally with the help of an offline catche.Using of this, we can view offline websites.The cache also ensures faster loading and better overall performance of the website.

**Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

**WHY USE CSS**

Use CSS to define styles for your documents, including the design, layout and variations in display for different devices and screen sizes. You can place your CSS in the <head> of a document with an embedded style sheet, or attach a separate file that defines your styles with an external style sheet. To link an external style sheet to your document, you'll simply add a link to the style sheet in the <head> of the document.

An external style sheet has many advantages. Keeping the styles separate from your HTML content:

* Helps avoid duplication
* Makes maintenance easier
* Allows you to make a site-wide change in one place

## **Action: Creating a stylesheet :**

1. Create another text file in the same directory as the doc1.html document you created in the first section.
2. Save your document as: style1.css. This file will be your stylesheet.
3. In your CSS file, copy and paste this one line, then save the file:

strong {color: red;}

### Linking your document to your stylesheet

1. To link your document to your stylesheet, edit your HTML file. Add the line highlighted here:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sample document</title>

<link rel="stylesheet" href="style1.css">

</head>

<body>

<p>

<strong>C</strong>ascading

<strong>S</strong>tyle

<strong>S</strong>heets

</p>

</body>

</html>

1. Save the file and refresh your browser's display. The stylesheet makes the initial letters red, like this:

Cascading Style Sheets

**HOW CSS WORKS**

When a browser displays a document, it must combine the document's content with its style information. It processes the document in two stages:

1. The browser converts the markup language and the CSS into the DOM (Document Object Model). The DOM represents the document in the computer's memory. It combines the document's content with its style.
2. The browser displays the contents of the DOM.

A markup language uses elements to define the document's structure. You mark an element using tags, which are strings beginning with '<' and ending with '>'. Most elements have paired tags, a start tag and an end tag. For the start tag, place the element name between '<' and '>'. For the end tag, place a '/' after the '<' and before the element name.

Depending on the markup language, some elements have only a start tag, or a single tag where the '/' comes after the element name. An element can also be a container and include other elements between its start tag and end tag. Just remember to always close the tags inside the container.

A DOM has a tree-like structure. Each element, attribute and run of text in the markup language becomes a node in the tree structure. The nodes are defined by their relationship to other DOM nodes. Some elements are parents of child nodes, and child nodes have siblings.

Understanding the DOM helps you design, debug and maintain your CSS, because the DOM is where your CSS and the document's content meet up.

**Example**

In your sample document, the <p> tag and its end tag </p> create a container:

<p>

<strong>C</strong>ascading

<strong>S</strong>tyle

<strong>S</strong>heets

</p>

### In the DOM, the corresponding P node is a parent. Its children are the STRONG nodes and the text nodes. The STRONG nodes are themselves parents, with text nodes as their children

### 

P

├─STRONG

│ └─"C"

├─"ascading"

├─STRONG

│ └─"S"

├─"tyle"

├─STRONG

│ └─"S"

└─"heets"

## **Action: Analyzing a DOM**

### Using DOM Inspector

To analyze a DOM, you need special software. You can use Mozilla's [DOM Inspector](https://developer.mozilla.org/en/DOM_Inspector)(DOMi) add-on to analyze a DOM. You simply need to install the add-on (see More details below).

1. Use your Mozilla browser to open your sample HTML document.
2. From your browser's menu bar, choose Tools > DOM Inspector, or Tools > Web Development > DOM Inspector.

3.In DOMi, expand your document's nodes by clicking on their arrows.

Note:Spacing in your HTML file may cause DOMi to show some empty text nodes, which you can ignore.

Part of the result might look like this, depending on which nodes you have expanded:

│ ▼╴P

│ │ │ ▼╴STRONG

│ │ └#text

│ ├╴#text

│ ►╴STRONG

│ │

When you select any node, you can use DOMi's right-hand pane to find out more about it. For example, when you select a text node, DOMi shows you the text in the right-hand pane.

When you select an element node, DOMi analyzes it and provides a huge amount of information in its right-hand pane. Style information is just part of the information it provides.

**CASCADING AND INHERITANCE**

The final style for an element can be specified in many different places, which can interact in a complex way. This complex interaction makes CSS powerful, but it can also make it confusing and difficult to debug.

Three main sources of style information form a cascade. They are:

* The browser's default styles for the markup language.
* Styles specified by a user who is reading the document.
* The styles linked to the document by its author. These can be specified in three places:
  + In an external file: this tutorial primarily discusses this method of defining styles.
  + In a definition at the beginning of the document: use this method only for styles that are used only on that page.
  + On a specific element in the body of the document: this is the least maintainable method, but can be used for testing.

The user's style modifies the browser's default style. The document author's style then modifies the style some more. In this tutorial, you are the author of your sample document, and you only work with author stylesheets.

When you open your sample document in your browser, the [<strong>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong)elements are bolder than the rest of the text. This comes from the browser's default style for HTML.

The[<strong>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong)elements are red. This comes from your own sample stylesheet.

The[<strong>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong)elements also inherit much of the[<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)element's style, because they are its children. In the same way, the[<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)element inherits much of the[<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body)element's style.

For styles in the cascade, author stylesheets have priority, then reader stylesheets, then the browser's defaults.

For inherited styles, a child node's own style has priority over style inherited from its parent.

These are not the only priorities that apply. A later page in this tutorial will explain more.

## **Action: Using inheritance**

1. Edit your sample CSS file.
2. Add this line by copying and pasting it. It does not really matter whether you add it above or below the line that is already there. However, adding it at the top is more logical because in your document the [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)element is the parent of the[<strong>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong)element:

p {color: blue; text-decoration: underline;}

1. Now refresh your browser to see the effect on your sample document. The underlining affects all the text in the paragraph, including the initial letters. The[<strong>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong)elements have inherited the underlined style from their parent[<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)element.

But the[<strong>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong)elements are still red. The red color is their own style, so it has priority over the blue color of their parent[<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)element.

## **Before**

### HTML Content

<p>

<strong>C</strong>ascading

<strong>S</strong>tyle

<strong>S</strong>heets

</p>

### CSS Content

strong {color:red}

Cascading Style Sheets

## **After**

### HTML Content

<p>

<strong>C</strong>ascading

<strong>S</strong>tyle

<strong>S</strong>heets

</p>

### CSS Content

p {color:blue; text-decoration:underline}

strong {color:red}

CascadingStyleSheets

**SELECTORS**

CSS has its own terminology to describe the CSS language. Previously in this tutorial, you created a line in your stylesheet like this:

strong {

color: red;

}

In CSS terminology, this entire line is a rule. This rule starts with strong, which is a selector (or a selector list). It selects which elements in the DOM the rule applies to.

The part inside the curly braces is the declaration.

The keyword color is a property, and red is a value.

The semicolon after the property-value pair separates it from other property-value pairs in the same declaration.In addition to tag names, you can use attribute values in selectors. This allows your rules to be more specific.

Two attributes have special status for CSS. They are [class](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes" \l "attr-class)and [id](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes" \l "id).

### Class selectors

Use the[class](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes" \l "attr-class)attribute in an element to assign the element to a named class. It is up to you what name you choose for the class. Multiple elements in a document can have the same class value.

In your stylesheet, type a full stop (period) before the class name when you use it in a selector.

### ID selectors

Use the [id](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes" \l "id)attribute in an element to assign an ID to the element. It is up to you what name you choose for the ID. The ID name must be unique in the document.

In your stylesheet, type a number sign (hash) before the ID when you use it in a selector.

**Example**

This HTML tag has both a class attribute and an id attribute:

<p class="key" id="principal">

The id value, principal, must be unique in the document, but other tags in the document can have the same class name, key.

In a CSS stylesheet, this rule makes all the elements with class key green. (They might not all be [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p) elements.)

.key {

color: green;

}

This rule makes the one element with the id principal bold:

#principal {

font-weight: bolder;

}

### Attribute Selectors

You are not restricted to the two special attributes, class and id. You can specify [other attributes](https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors)by using square brackets. Inside the brackets you put the attribute name, optionally followed by a matching operator and a value. Additionally, matching can be made case-insensitive by appending an " i" after the value, but not many browsers support this feature yet. Examples:

**[disabled]**

Selects all elements with a "disabled" attribute.

**[type='button']**

Selects elements with a "button" type.

**[class~=key]**

Selects elements with the class "key" (but not e.g. "keyed", "monkey", "buckeye"). Functionally equivalent to .key.

**[lang|=es]**

Selects elements specified as Spanish. This includes "es" and "es-MX" but not "eu-ES" (which is Basque).

**[title\*="example" i]**

Selects elements whose title contains "example", ignoring case. In browsers that don't support the "i" flag, this selector probably won't match any element.

**a[href^="https://"]**

Specifies what the attribute's value should start with;in this case, it selects secure links.

**img[src$=".png"]**

Selects elements whose value ends with the provided string. Indirectly selects PNG images; any images that are PNGs but whose URL doesn't end in ".png" (e.g. `src="some-image.png?\_=cachebusterhash"`) won't be selected.

### Pseudo-classes selectors

A CSS[pseudo-class](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes)is a keyword added to selectors that specifies a special state of the element to be selected. For example[:hover](https://developer.mozilla.org/en-US/docs/Web/CSS/:hover)will apply a style when the user hovers over the element specified by the selector.

Pseudo-classes, together with pseudo-elements, let you apply a style to an element not only in relation to the content of the document tree, but also in relation to external factors like the history of the navigator ([:visited](https://developer.mozilla.org/en-US/docs/Web/CSS/:visited), for example), the status of its content (like[:checked](https://developer.mozilla.org/en-US/docs/Web/CSS/:checked)on some form elements), or the position of the mouse (like[:hover](https://developer.mozilla.org/en-US/docs/Web/CSS/:hover)which lets you know if the mouse is over an element or not).

**Syntax**

selector:pseudo-class {

property: value;

}

#### List of pseudo-classes

* [:link](https://developer.mozilla.org/en-US/docs/Web/CSS/:link)
* [:visited](https://developer.mozilla.org/en-US/docs/Web/CSS/:visited)
* [:active](https://developer.mozilla.org/en-US/docs/Web/CSS/:active)
* [:hover](https://developer.mozilla.org/en-US/docs/Web/CSS/:hover)
* [:focus](https://developer.mozilla.org/en-US/docs/Web/CSS/:focus)
* [:first-child](https://developer.mozilla.org/en-US/docs/Web/CSS/:first-child)
* [:last-child](https://developer.mozilla.org/en-US/docs/Web/CSS/:last-child)
* [:nth-child](https://developer.mozilla.org/en-US/docs/Web/CSS/:nth-child)
* [:nth-last-child](https://developer.mozilla.org/en-US/docs/Web/CSS/:nth-last-child)
* [:nth-of-type](https://developer.mozilla.org/en-US/docs/Web/CSS/:nth-of-type)
* [:first-of-type](https://developer.mozilla.org/en-US/docs/Web/CSS/:first-of-type)
* [:last-of-type](https://developer.mozilla.org/en-US/docs/Web/CSS/:last-of-type)
* [:empty](https://developer.mozilla.org/en-US/docs/Web/CSS/:empty)
* [:target](https://developer.mozilla.org/en-US/docs/Web/CSS/:target)
* [:checked](https://developer.mozilla.org/en-US/docs/Web/CSS/:checked)
* [:enabled](https://developer.mozilla.org/en-US/docs/Web/CSS/:enabled)
* [:disabled](https://developer.mozilla.org/en-US/docs/Web/CSS/:disabled)

### Selector lists

A rule can be shared by multiple selectors by using a comma (,) to separate selectors.

**Example**

In the example, both of elements that have the "content-1" class and elements that have the "content-2" class will display bold text.

.content-1, .content-2 {

font-weight: bold;

}

## **Information: Specificity**

Multiple rules may have selectors that each match the same element. If a property is given in only one rule, there is no conflict and the property is set on the element. If more than one rule applies to an element and sets the same property, then CSS gives priority to the rule that has the more[specific](https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity)selector. An ID selector is more specific than a class, pseudo-class or attribute selector, which in turn are more specific than a tag or pseudo-element selector.

When you have a problem with conflicting rules, try to resolve it by making one of the rules more specific, so that it has priority. If you cannot do that, try moving one of the rules nearer the end of the stylesheet so that it has priority.

## **Information: Selectors based on relationships**

CSS has some ways to select elements based on the relationships between elements. You can use these to make selectors that are more specific.

Common selectors based on relationships

|  |  |
| --- | --- |
| Selector | Selects |
| A E | Any E element that is a descendant of an A element (that is: a child, or a child of a child,etc.) |
| A > E | Any E element that is a child (i.e. direct descendant) of an A element |
| E:first-child | Any E element that is the first child of its parent |
| B + E | Any E element that is the next sibling of a B element (that is: the next child of the same parent) |

We can also use the symbol \* (asterisk) to mean "any element".

An HTML table has an id attribute, but its rows and cells do not have individual identifiers:

<table id="data-table-1">

...

<tr>

<td>Prefix</td>

<td>0001</td>

<td>default</td>

</tr>

...

data-table-1 td:first-child {text-decoration: underline;}

#data-table-1 td:first-child + td {text-decoration: line-through;}

The result looks like:

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Prefix | ~~0001~~ | default | |

## **Action: Using class and ID selectors**

1. Edit your HTML file, and duplicate the paragraph by copying and pasting it.
2. Then add id and class attributes to the first copy, and an id attribute to the second copy as shown below. Alternatively, copy and paste the entire file again:

<!doctype html>

<html>

<head>

<meta charset="UTF-8">

<title>Sample document</title>

<link rel="stylesheet" href="style1.css">

</head>

<body>

<p id="first">

<strong class="carrot">C</strong>ascading

<strong class="spinach">S</strong>tyle

<strong class="spinach">S</strong>heets

</p>

<p id="second">

<strong>C</strong>ascading

<strong>S</strong>tyle

<strong>S</strong>heets

</p>

</body>

</html>

3.Now edit your CSS file. Replace the entire contents with:

strong { color: red; }

.carrot { color: orange; }

.spinach { color: green; }

#first { font-style: italic; }

4.Save the files and refresh your browser to see the result:

|  |
| --- |
| *Cascading Style Sheets* |
| Cascading StyleSheets |

We can try rearranging the lines in your CSS file to show that the order has no effect.

The class selectors.carrotand.spinach have priority over the tag selector strong.

The ID selector#firsthas priority over the class and tag selectors.

1. Without changing your HTML file, add a single rule to your CSS file that keeps all the initial letters that same color as they are now, but makes all the other text in the second paragraph blue:

|  |
| --- |
| *Cascading StyleSheets* |
| Cascading Style Sheets |

2.Now change the rule you have just added (without changing anything else), to make the first paragraph blue too:

|  |
| --- |
| *CascadingStyleSheets* |
| CascadingStyleSheets |

## **Action: Using pseudo-classes selectors**

1.Create an HTML file with following content:

<!doctype html>

<html>

<head>

<meta charset="UTF-8">

<title>Sample document</title>

<link rel="stylesheet" href="style1.css">

</head>

<body>

<p>Go to our <a class="homepage" href="http://www.example.com/" title="Home page">Home page</a>.</p>

</body>

</html>

2.Now edit your CSS file. Replace the entire contents with:

a.homepage:link, a.homepage:visited {

padding: 1px 10px 1px 10px;

color: #fff;

background: #555;

border-radius: 3px;

border: 1px outset rgba(50,50,50,.5);

font-family: georgia, serif;

font-size: 14px;

font-style: italic;

text-decoration: none;

}

a.homepage:hover, a.homepage:focus, a.homepage:active {

background-color: #666;

}

3.Save the files and refresh your browser to see the result (put the mouse over the following link to see the effect):

|  |
| --- |
| Go to our [*Home page*](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Selectors) |

**Action: Using selectors based on relationships and pseudo-classes**

With selectors based on relationships and pseudo-classes you can create complex cascade algorithms. This is a common technique used, for example, in order to create pure-CSS dropdown menus (that is only CSS, without using [JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript)). The essence of this technique is the creation of a rule like the following:

div.menu-bar ul ul {

display: none;

}

div.menu-bar li:hover > ul {

display: block;

}

to be applied to an HTML structure like the following:

<div class="menu-bar">

<ul>

<li>

<a href="example.html">Menu</a>

<ul>

<li>

<a href="example.html">Link</a>

</li>

<li>

<a class="menu-nav" href="example.html">Submenu</a>

<ul>

<li>

<a class="menu-nav" href="example.html">Submenu</a>

<ul>

<li><a href="example.html">Link</a></li>

<li><a href="example.html">Link</a></li>

<li><a href="example.html">Link</a></li>

<li><a href="example.html">Link</a></li>

</ul>

</li>

<li><a href="example.html">Link</a></li>

</ul>

</li>

</ul>

</li>

</ul>

</div>

**READABLE CSS**

We can add white space and comments to your stylesheets to make them more readable. We can also group selectors together, when the same style rules apply to elements selected in different ways.

### *White space*

White space means actual spaces, tabs and new lines. You can add white space to make your stylesheets more readable.

In the context of page layout and composition, white space is the portion of a page that is left unmarked: margins, gutters, and space between columns and lines of type.

Your sample CSS file currently has one rule per line, and almost the minimum of white space. In a complex stylesheet this layout would be difficult to read, making the stylesheet difficult to maintain.

The layout you choose is usually a personal preference, but if your stylesheets are part of shared projects, those projects might have their own conventions.

**Examples**

Some people like the compact layout that we have been using, only splitting a line when it becomes very long:

.carrot {color: orange; text-decoration: underline; font-style: italic;}

Some people prefer one property-value per line:

.carrot

{

color: orange;

text-decoration: underline;

font-style: italic;

}

Some people use indention—two spaces, four spaces, or a tab are common:

.carrot {

color: orange;

text-decoration: underline;

font-style: italic;

}

Some people like everything to line up vertically (but a layout like this is difficult to maintain):

.carrot

{

color : orange;

text-decoration : underline;

font-style : italic;

}

Some people use mixed whitespace to enhance readability.

.vegetable { color: green; min-height: 5px; min-width: 5px }

.vegetable.carrot { color: orange; height: 90px; width: 10px }

.vegetable.spinach { color: darkgreen; height: 30px; width: 30px }

Some people use tabs for the layout. Some people only use spaces.

### **Comments**

Comments in CSS begin with/\*and end with\*/.

You can use comments to make actual comments in your stylesheet, and also to comment out parts of it temporarily for testing purposes.

To comment out part of a stylesheet, place that part in a comment so that the browser ignores it. Be careful where you start and end the comment. The rest of the stylesheet must still have correct syntax.

/\* style for initial letter C in first paragraph \*/

.carrot {

color: orange;

text-decoration: underline;

font-style: italic;

}

### **Grouped selectors**

When many elements have the same style, you can specify a group of selectors, separating them with commas. The declaration applies to all the selected elements.

Elsewhere in your stylesheet you can specify the same selectors again individually, to apply individual style rules to them.

**Example**

This rule makes[<h1>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h1),[<h2>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h2), and[<h3>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h3)elements the same color.

It is convenient to specify the color in only one place, in case it has to be changed.

/\* color for headings \*/

h1, h2, h3 {color: navy;}

## **Action: Adding comments and improving the layout**

1. Edit your CSS file, and ensure that it has these rules in it (in any order):

strong {color: red;}

.carrot {color: orange;}

.spinach {color: green;}

#first {font-style: italic;}

p {color: blue;}

2.Make it more readable by rearranging it in a way that you find logical, and by adding white space and comments in whatever way you think best.

3.Save the file and refresh your browser's display, to make sure that your changes do not affect how the stylesheet works:

|  |
| --- |
| *CascadingStyleSheets* |
| CascadingStyleSheets |

**TEXT STYLES**

CSS has several properties for styling text.

There is a convenient shorthand property,[font](https://developer.mozilla.org/en-US/docs/Web/CSS/font), which you can use to specify several aspects at once—for example:

* Bold, italic, and small-caps (small capitals)
* Size
* Line height
* Font typeface

**Example**

p {

font: italic 75%/125% "Comic Sans MS", cursive;

}

This rule sets various font properties, making all paragraphs italic.

The font size is set to three-quarters of the size in each paragraph's parent element, and the line height is set to 125% (a little more spaced than normal).

The font typeface is set to Comic Sans MS, but if this typeface is not available then the browser will use its default cursive (hand-written) typeface.

The rule has the side-effect of turning off bold and small-caps (setting them to normal).

### **Font faces**

You cannot predict what typefaces the readers of your document will have. When you specify font typefaces, it is a good idea to give a list of alternatives in order of preference.

End the list with one of the built-in default typefaces: serif, sans-serif,cursive,fantasyormonospace.

If the typeface does not support some features in the document, then the browser can substitute a different typeface. For example, the document might contain special characters that the typeface does not support. If the browser can find another typeface that has those characters, then it will use the other typeface.

To specify a typeface on its own, use the[font-family](https://developer.mozilla.org/en-US/docs/Web/CSS/font-family)property.

### **Font sizes**

Browser users can override the default font sizes or change the text size while they read a page, so it makes good sense for you to use relative sizes wherever you can.

You can use some built-in values for font sizes, likesmall,mediumandlarge. You can also use values relative to the font size of the parent elementlike:smaller,larger,150%or1.5em. An "em" is equivalent to the width of the letter "m" (for the font size of the parent element); thus 1.5em is one-and-a-half times the size of the font of the parent element.

If necessary you can specify an actual sizelike:14px(14 pixels) for a display device or 14pt (14 points) for a printer. This is not accessible for visually impaired users because it does not allow them to change the size. A more accessible strategy is to set a built-in value like medium on a top-level element of the document, and then set relative sizes for all its descendent elements.

To specify a font size on its own, use the[font-size](https://developer.mozilla.org/en-US/docs/Web/CSS/font-size) property.

**Line height**

The line height specifies the spacing between lines. If your document has long paragraphs with many lines, a larger-than-normal spacing makes it easier to read, especially if the font size is small. To specify a line height on its own, use the[line-height](https://developer.mozilla.org/en-US/docs/Web/CSS/line-height)property.

### **Decoration**

The separate[text-decoration](https://developer.mozilla.org/en-US/docs/Web/CSS/text-decoration)property can specify other styles, like underline or line-through. You can set it to none to explicitly remove any decoration.

### **Other properties**

To specify italic on its own, use [font-style](https://developer.mozilla.org/en-US/docs/Web/CSS/font-style): italic;  
To specify bold on its own, use [font-weight](https://developer.mozilla.org/en-US/docs/Web/CSS/font-weight): bold;  
To specify small capitals on its own, use [font-variant](https://developer.mozilla.org/en-US/docs/Web/CSS/font-variant): small-caps;

## **Specifying fonts**

For a simple document, you can set the font of the[<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body)element and the rest of the document inherits the settings.

1. Edit your CSS file.
2. Add the following rule to change the font throughout the document. The top of the CSS file is a logical place for it, but it has the same effect wherever you put it:

body {

font: 16px "Comic Sans MS", cursive;

}

3.Add a comment explaining the rule, and add white space to make it match your preferred layout.

4.Save the file and refresh your browser to see the effect. If your system has Comic Sans MS, or another cursive font that does not support italic, then your browser chooses a different font face for the italic text in the first line:

|  |
| --- |
| *CascadingStyleSheets* |
| Cascading StyleSheets |

5.From your browser's menu bar, choose View > Text Size > Increase (or View > Zoom > Zoom In). Even though you specified 16 pixels in the style, a user reading the document can change the size.

Without changing anything else, make all six initial letters twice the size in the browser's default serif font:

|  |
| --- |
| C*ascading*S*tyle*S*heets* |
| CascadingStyleSheets |

## **Color**

In this tutorial so far, you have used a limited number of named colors. CSS2 supports 17 named colors in all. Some of the names might not be what you expect:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | black |  | gray |  | silver |  | white |  |  |  |  |  |  |  |
| primaries | red |  | lime |  | blue |  |  | |  |  |  |  |  |  |
| secondaries | yellow |  | aqua |  | fuchsia |  |  | |  |  |  |  |  |  |
|  | maroon |  | orange |  | olive |  | purple |  | green |  | navy |  | teal |  |

For a larger palette, specify the red, green and blue components of the color you want by using a number sign (hash) and three hexadecimal digits in the range 0 – 9, a – f. The letters a – f represent the values 10 – 15:

|  |  |  |
| --- | --- | --- |
| black |  | #000 |
| pure red |  | #f00 |
| pure green |  | #0f0 |
| pure blue |  | #00f |
| white |  | #fff |

For the full palette, specify two hexadecimal digits for each component:

|  |  |  |
| --- | --- | --- |
| black |  | #000000 |
| pure red |  | #ff0000 |
| pure green |  | #00ff00 |
| pure blue |  | #0000ff |
| white |  | #ffffff |

You can usually get these six-digit hexadecimal codes from your graphics program or some other tool.

With a little practice, you can adjust the three-digit colors manually for most purposes:

|  |  |  |
| --- | --- | --- |
| Start with pure red: |  | #f00 |
| To make it paler, add some green and blue: |  | #f77 |
| To make it more orange, add a little extra green: |  | #fa7 |
| To darken it, reduce all its components: |  | #c74 |
| To reduce its saturation, make its components more equal: |  | #c98 |
| If you make them exactly equal, you get gray: |  | #ccc |

or a pastel shade like pale blue:

|  |  |  |
| --- | --- | --- |
| Start with pure white: |  | #fff |
| Reduce the other components a little: |  | #eef |

You can also specify a color using decimal RGB values in the range 0 – 255, or percentages.

For example, this is maroon (dark red):

rgb(128, 0, 0)

### **Color properties**

You have already used the[color](https://developer.mozilla.org/en-US/docs/Web/CSS/color)property on text.

You can also use the[background-color](https://developer.mozilla.org/en-US/docs/Web/CSS/background-color)property to change elements' backgrounds.

Backgrounds can be set to transparent to explicitly remove any color, revealing the parent element's background.

The Exampleboxes in this tutorial use this pale yellow background:

background-color: #fffff4;

TheMore details boxes use this pale gray:

background-color: #f4f4f4;

## **Using color codes** **Color page**

1.Edit your CSS file.

2.Make the change shown down here, to give the initial letters a pale blue background. (The layout and comments in your file probably differ from the file shown here. Keep the layout and comments the way you prefer them.)

### HTML Content

<p id = "first"> <strong>C</strong>ascading <strong class="spinach">S</strong>tyle <strong class="spinach">S</strong>heets</p>

<p> <strong>C</strong>ascading <strong>S</strong>tyle <strong>S</strong>heets</p>

### CSS Content

/\*\*\* CSS Tutorial: Color page \*\*\*/

/\* page font \*/

body {

font: 16px "Comic Sans MS", cursive;

}

/\* paragraphs \*/

p {

color: blue;

}

#first {

font-style: italic;

}

/\* initial letters \*/

strong {

background-color: #ddf;

color: red;

font: 200% serif;

}

.spinach {

color: green;

background-color: #ddf;

}

4.Save the file and refresh your browser to see the result.

5.The result should be like this:

C*ascading*S*tyle*S*heets*

CascadingStyleSheets

CONTENT

One of the important advantages of CSS is that it helps you to separate a document's style from its content. Yet there are situations where it makes sense to specify certain content as part of the stylesheet, not as part of the document.

Content specified in a stylesheet can consist of text or images. You specify content in your stylesheet when the content is closely linked to the document's structure.

Specifying content in a stylesheet can cause complications. For example, you might have different language versions of your document that share a stylesheet. If part of the stylesheet has to be translated, it means that you must put those parts of the stylesheet in separate files and arrange for them to be linked with the appropriate language versions of your document.

These complications do not arise if the content you specify consists of symbols or images that apply in all languages and cultures.

Content specified in a stylesheet does not become part of the DOM.

### Text content

CSS can insert text content before or after an element. To specify this, make a rule and add [::before](https://developer.mozilla.org/en-US/docs/Web/CSS/::before)or [::after](https://developer.mozilla.org/en-US/docs/Web/CSS/::after)to the selector. In the declaration, specify the[content](https://developer.mozilla.org/en-US/docs/Web/CSS/content)property with the text content as its value.

**Example**

#### HTML

A text where I need to <span class="ref">something</span>

#### CSS

.ref::before {

font-weight: bold;

color: navy;

content: "Reference: ";

}

#### Output

### Image content

To add an image before or after an element, you can specify the URL of an image file in the value of the [content](https://developer.mozilla.org/en-US/docs/Web/CSS/content)property.

**Example**

This rule adds a space and an icon after every link that has the class glossary:

a.glossary:after {content: " " url("../images/glossary-icon.gif");}

To add an image as an element's background, specify the URL of an image file in the value of the background property. This is a shorthand property that specifies the background color, the image, how the image repeats, and some other details.

**Example**

This rule sets the background of a specific element, using a URL to specify an image file.

The selector specifies the element's id. The value no-repeat makes the image appear only once:

#sidebar-box {background: url("../images/sidebar-ground.png") no-repeat;}

## **Adding a background image**

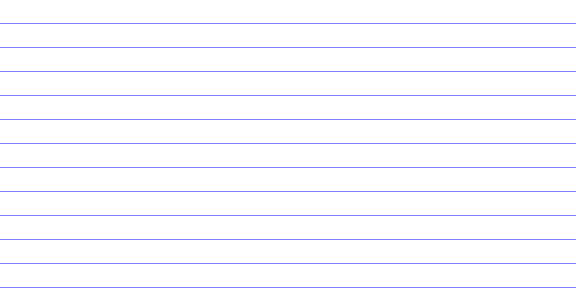
This image is a white square with a blue line at the bottom:

|  |
| --- |
|  |

1. Download the image file into the same directory as your CSS file. (For example, right-click it to get a context menu, then choose Save Image As and specify the directory that you are using for this tutorial.)
2. Edit your CSS file and add this rule to the body, setting a background image for the entire page.

background: url("Blue-rule.png");

The value repeat is the default, so it does not need to be specified. The image repeats horizontally and vertically, giving an appearance like lined writing paper:



C*ascading*S*tyle*S*heets*

CascadingStyle Sheets

**LISTS**

CSS provides special properties that are designed for lists. It is usually more convenient to use these properties whenever you can.

To specify the style for a list, use the[list-style](https://developer.mozilla.org/en-US/docs/Web/CSS/list-style)property to specify the type of marker.

The selector in your CSS rule can either select the list item elements (for example,[<li>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li)), or it can select the parent list element (for example,[<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul)) so that the list elements inherit the style.

### Unordered lists

In anunordered list, each list item is marked in the same way.

CSS has three types of markers, and here is how your browser displays them:

* disc
* circle
* square

Alternatively, you can specify the URL of an image.

**Example**

These rules specify different markers for different classes of list item:

li.open {list-style: circle;}

li.closed {list-style: disc;}

When these classes are used in a list, they distinguish between open and closed items (for example, in a to-do list):

<ul>

<li class="open">Lorem ipsum</li>

<li class="closed">Dolor sit</li>

<li class="closed">Amet consectetuer</li>

<li class="open">Magna aliquam</li>

<li class="closed">Autem veleum</li>

</ul>

The result might look like:

* Dolor sit
* Lorem ipsum
* Amet consectetuer
* Magna aliquam
* Autem veleum

Ordered lists

In anorderedlist, each list item is marked differently to show its position in the sequence.

Use the [list-style](https://developer.mozilla.org/en-US/docs/Web/CSS/list-style)property to specify the type of marker:

* decimal
* lower-roman
* upper-roman
* lower-latin
* upper-latin

**Example**

This rule specifies that in [<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol) elements with the class info, the items are identified with capital letters.

<ol class="info">

<li>Lorem ipsum</li>

<li>Dolor sit</li>

<li>Amet consectetuer</li>

<li>Magna aliquam</li>

<li>Autem veleum</li>

</ol>

ol.info {list-style: upper-latin;}

The[<li>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li)elements in the list inherit this style:

1. Lorem ipsum

2.Dolor sit

3.Amet consectetuer

4.Magna aliquam

5.Autem veleum

If you are using a markup language like HTML that provides conventional tags for unordered ([<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul)) and ordered ([<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol)) lists, then it is good practice to use the tags in the way they were intended. However, you can use CSS to make [<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul) display ordered and [<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol) display unordered if you wish.

Browsers differ in the way they implement the styles for lists. Do not expect your stylesheet to give identical results in all browsers.

### Counters

You can use counters to number any elements, not only list items. For example, in some documents you might want to number headings or paragraphs.

To specify numbering, you need a counter with a name that you specify.

In some element before the counting is to start, reset the counter with the property[counter-reset](https://developer.mozilla.org/en-US/docs/Web/CSS/counter-reset)and the name of your counter. The parent of the elements you are counting is a good place to do this, but you can use any element that comes before the list items.

In each element where the counter increments, use the property[counter-increment](https://developer.mozilla.org/en-US/docs/Web/CSS/counter-increment)and the name of your counter.

To display your counter, add[::before](https://developer.mozilla.org/en-US/docs/Web/CSS/::before)or[::after](https://developer.mozilla.org/en-US/docs/Web/CSS/::after)to the selector and use thecontentproperty (as you did on the previous page,[Content](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Content)).

In the value of the content property, specifycounter()with the name of your counter. Optionally specify a type. The types are the same as in theOrdered listssection above.

Normally the element that displays the counter also increments it.

**Example**

This rule initializes a counter for every [<h3>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h3)element with the class numbered:

h3.numbered {counter-reset: mynum;}

This rule displays and increments the counter for every[<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)element with the class numbered:

<p class="numbered">Lorem ipsum</p>

<p class="numbered">Dolor sit</p>

<p class="numbered">Amet consectetuer</p>

<p class="numbered">Magna aliquam</p>

<p class="numbered">Autem veleum</p>

body {

counter-reset: mynum;

}

p.numbered:before {

content: counter(mynum) ": ";

counter-increment: mynum;

font-weight: bold;

}

The result looks like this:

Lorem ipsum

Dolor sit

Amet consectetuer

Magna aliquam

Autem veleum

## **Styled lists**

Make a new HTML document, doc2.html. Copy and paste the content from here:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Sample document 2</title>

<link rel="stylesheet" href="style2.css">

</head>

<body>

<h3 id="oceans">The oceans</h3>

<ul>

<li>Arctic</li>

<li>Atlantic</li>

<li>Pacific</li>

<li>Indian</li>

<li>Southern</li>

</ul>

<h3 class="numbered">Numbered paragraphs</h3>

<p class="numbered">Lorem ipsum</p>

<p class="numbered">Dolor sit</p>

<p class="numbered">Amet consectetuer</p>

<p class="numbered">Magna aliquam</p>

<p class="numbered">Autem veleum</p>

</body>

</html>

Make a new stylesheet, style2.css. Copy and paste the content from here:

/\* numbered paragraphs \*/

h3.numbered {counter-reset: mynum;}

p.numbered:before {

content: counter(mynum) ": ";

counter-increment: mynum;

font-weight: bold;

}

If the layout and comment are not to your taste, change them.

Open the document in your browser. If your browser supports counters, you see something like the example below. If your browser does not support counters, then you do not see the numbers (and probably not even the colons):

### The oceans

* Arctic
* Atlantic
* Pacific
* Indian
* Southern

Numbered paragraphs

Lorem ipsum

Dolor sit

Amet consectetuer

Magna aliquam

Autem veleum

### BOXES

When your browser displays an element, the element takes up space. There are four parts to the space that it takes up.

In the middle, there is the space that th element needs to display its content. Around that, there is padding. Around that, there is a border. Around that, there is a margin that separates the element from other elements.

|  |  |
| --- | --- |
| margin  border  padding  **element**  The pale gray shows parts of the layout. | **element**  This is what you see in your browser. |

The padding, border and margin can have different sizes on the top, right, bottom and left of the element. Any or all of these sizes can be zero.

### Coloring

The padding is always the same color as the element's background. So when you set the background color, you see the color applied to the element itself and its padding. The margin is always transparent.

|  |  |
| --- | --- |
| margin  border  padding  **element**  The element has a green background. | **element**  This is what you see in your browser. |

### Borders

You can use borders to decorate elements with lines or boxes.

To specify the same border all around an element, use the [border](https://developer.mozilla.org/en-US/docs/Web/CSS/border)property. Specify the width (usually in pixels for display on a screen), the style, and the color.

We can also set the style to none orhidden to explicitly remove the border, or set the color to transparent to make the border invisible without changing the layout.

To specify borders one side at a time, use the properties: [border-top](https://developer.mozilla.org/en-US/docs/Web/CSS/border-top), [border-right](https://developer.mozilla.org/en-US/docs/Web/CSS/border-right), [border-bottom](https://developer.mozilla.org/en-US/docs/Web/CSS/border-bottom), [border-left](https://developer.mozilla.org/en-US/docs/Web/CSS/border-left). You can use these to specify a border on only one side, or different borders on different sides.

**Example**

This rule sets the background color and the top border of heading elements:

h3 {

border-top: 4px solid #7c7; /\* mid green \*/

background-color: #efe; /\* pale green \*/

color: #050; /\* dark green \*/

}

The result looks like :

|  |
| --- |
| **Stylish heading** |

This rule makes images easier to see by giving them a mid-gray border all round:

img {border: 2px solid #ccc;}

The result looks like:

|  |  |
| --- | --- |
| Image: |  |

### Margins and padding

Use margins and padding to adjust elements' positions and to create space around them.

Use the [margin](https://developer.mozilla.org/en-US/docs/Web/CSS/margin)property or the [padding](https://developer.mozilla.org/en-US/docs/Web/CSS/padding)property to set the margin or padding widths respectively.

If you specify one width, it applies all around the element (top, right, bottom and left).

If you specify two widths, the first applies to the top and bottom, the second to the right and left.

You can specify all four widths in the order: top, right, bottom, left.

**Example**

This rule marks out paragraphs with the class remark by giving them a red border all round.

Padding all round separates the border from the text a little.

A left margin indents the paragraph relative to the rest of the text:

p.remark {

border: 2px solid red;

padding: 4px;

margin-left: 24px;

}

The result looks like:

|  |
| --- |
| Here is a normal paragraph.  Here is a remark. |

## **Adding borders**

Edit your CSS file, style2.css. Add this rule to draw a line across the page over each heading:

h3 {border-top: 1px solid gray;}

If you took the challenge on the last page, modify the rule you created, otherwise add this new rule to add space underneath each list item:

li {

list-style: lower-roman;

margin-bottom: 8px;

}

Refresh your browser to see the result:

|  |
| --- |
| **(A) The oceans**   * Arctic * Atlantic * Pacific * Indian * Southern   **(B) Numbered paragraphs**  1: Lorem ipsum  2: Dolor sit  3: Amet consectetuer  4: Magna aliquam  5: Autem veleum |

### LAYOUT

We can use CSS to specify various visual effects that change the layout of your document. Some of the techniques for specifying layout are advanced, and beyond the scope of this basic tutorial.

When you design a layout to look similar in many browsers, your stylesheet interacts with the browser's default stylesheet and layout engine in ways that can be complex. This is also an advanced subject, beyond the scope of this basic tutorial.

### **Document structure**

If you want to control the layout of your document, then you might have to change its structure.Your document's markup language might have general-purpose tags for creating structure. For example, in HTML you can use the [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div)element to create structure.

**Example**

In your sample document, the numbered paragraphs under the second heading do not have a container of their own.

Your stylesheet cannot draw a border around those paragraphs, because there is no element to specify in the selector.

To fix this structural problem, you can add a [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div) tag around the paragraphs. This tag is unique, so it can be identified by an id attribute:

<h3>Numbered paragraphs</h3>

<div id="numbered">

<p>Lorem ipsum</p>

<p>Dolor sit</p>

<p>Amet consectetuer</p>

<p>Magna aliquam</p>

<p>Autem veleum</p>

</div>

Now your stylesheet can use one rule to specify borders around both lists:

ul, #numbered {

border: 1em solid #69b;

padding-right:1em;

}

### Size units

So far in this tutorial, you have specified sizes in pixels (px). These are appropriate for some purposes on a display device like a computer screen. But when the user changes the font size, your layout can look wrong.

For many purposes it is better to specify sizes as a percentage or in ems (em). An em is nominally the size of the current font (the width of a letter m). When the user changes the font size, your layout adjusts automatically.

**Example**

The border on the left of this text has its size specified in pixels.

The border on the right has its size specified in ems.

In your browser, change the size of the font to see how the border on the right adjusts but the border on the left does not:

RESIZE ME PLEASE

### **Text layout**

Two properties specify how the content of an element is aligned. You can use them for simple layout adjustments:

[**text-align**](https://developer.mozilla.org/en-US/docs/Web/CSS/text-align)

Aligns the content. Use one of these values: left, right, center, justify

[**text-indent**](https://developer.mozilla.org/en-US/docs/Web/CSS/text-indent)

Indents the content by an amount that you specify.

These properties apply to any text-like content in the element, not only to actual text. Remember that they are inherited by the element's children, so you might have to explicitly turn them off in the children to avoid surprising results.

**Example**

To center headings:

h3 {

border-top: 1px solid gray;

text-align: center;

}

### **Floats**

The [float](https://developer.mozilla.org/en-US/docs/Web/CSS/float)property forces an element to the left or right. This is a simple way to control its position and size.

The rest of the document's content normally flows around the floated element. You can control this by using the[clear](https://developer.mozilla.org/en-US/docs/Web/CSS/clear)property on other elements to make them stay clear of floats.

**Example**

In your sample document, the lists stretch across the window. You can prevent this by floating them to the left.

To keep the headings in place, you must also specify that they stay clear of floats on their left:

ul, #numbered {float: left;}

h3 {clear: left;}

### **Positioning**

You can specify an element's position in four ways by specifying the [position](https://developer.mozilla.org/en-US/docs/Web/CSS/position) property and one of the following values.

These are advanced properties. It is possible to use them in simple ways—that is why they are mentioned in this basic tutorial. But using them for complex layouts can be difficult.

**relative**

The element's position is shifted relative to its normal position. Use this to shift an element by a specified amount. You can sometimes use the element's margin to achieve the same effect.

**fixed**

The element's position is fixed. Specify the element's position relative to the document window. Even if the rest of the document scrolls, the element remains fixed.

**absolute**

The element's position is fixed relative to a parent element. Only a parent that is itself positioned with relative, fixed or absolute will do. You can make any parent element suitable by specifying position: relative; for it without specifying any shift.

**static**

The default. Use this value if you need to turn positioning off explicitly.

Along with these values of the position property (except for static), specify one or more of the properties: top, right, bottom, left,width, height to identify where you want the element to appear, and perhaps also its size.

**Example**

To position two elements on top of each other, create a parent container in your document with the two elements inside it:

<div id="parent-div">

<p id="forward">/</p>

<p id="back">\</p>

</div>

In your stylesheet, make the parent's position relative. There is no need to specify any actual shift. Make the children's positionabsolute:

#parent-div {

position: relative;

font: bold 200% sans-serif;

}

#forward, #back {

position: absolute;

margin:0px; /\* no margin around the elements \*/

top: 0px; /\* distance from top \*/

left: 0px; /\* distance from left \*/

}

#forward {

color: blue;

}

#back {

color: red;

}

## **Specifying layout**

1. Change your sample document,doc2.html, and stylesheet,style2.css, using the examples above in the sections[Document structure](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Layout" \l "Document_structure)and[Floats](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Layout" \l "Floats).
2. In the [Floats](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Layout" \l "Floats)example, add padding to separate the text from the right border by 0.5 em.

<img id="fixed-pin" src="Yellow-pin.png" alt="Yellow map pin">

### TABLES

A table is an arrangement of information in a rectangular grid. Some tables can be complex, and for complex tables different browsers can give different results.

When you design your document, use a table to express the [relationships](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Selectors" \l "relselectors)between the pieces of information. Then it does not matter if different browsers present the information in slightly different ways, because the meaning is still clear.

Do not use tables in unusual ways to produce particular visual layouts. The techniques on the previous page of this tutorial ([Layout](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Layout)) are better for that purpose.

### Table structure

In a table, each piece of information is displayed in a cell.

The cells in a line across the page make up a row.

In some tables, the rows might be grouped. A special group of rows at the start of the table is the header. A special group of rows at the end of the table is the footer. The main rows in the table are the body, and they might also be in groups.

The cells in a line down the page make up a column, but columns have limited use in CSS tables.

### Borders

Cells have no margins.

Cells do have borders and padding. By default, the borders are separated by the value of the table's [border-spacing](https://developer.mozilla.org/en-US/docs/Web/CSS/border-spacing)property. You can also remove the spacing completely by setting the table's [border-collapse](https://developer.mozilla.org/en-US/docs/Web/CSS/border-collapse)property tocollapse.

**Example**

Here are three tables.The table on the left has 0.5 em border spacing. The table in the center has zero border spacing. The table on the right has collapsed borders:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Clubs | Hearts | | Diamonds | Spades |  |  |  | | --- | --- | | Clubs | Hearts | | Diamonds | Spades | | |  |  | | --- | --- | | Clubs | Hearts | | Diamonds | Spades | |  |

### Captions

A [<caption>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/caption)element is a label that applies to the entire table. By default, it is displayed at the top of the table.

To move it to the bottom, set its[caption-side](https://developer.mozilla.org/en-US/docs/Web/CSS/caption-side)property tobottom. The property is inherited, so alternatively you can set it on the table or another ancestor element.

To style the text of the caption, use any of the usual properties for text.

**Example**

This table has a caption at the bottom

#demo-table > caption {

caption-side: bottom;

font-style: italic;

text-align: right;

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Suits   |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | Clubs | Hearts | | Diamonds | Spades | | |

### Empty cells

You can display empty cells (that is, their borders and backgrounds) by specifying [empty-cells](https://developer.mozilla.org/en-US/docs/Web/CSS/empty-cells): show; for the table element.

You can hide them by specifying empty-cells: hide;. Then, if a cell's parent element has a background, it shows through the empty cell.

**Example**

These tables have a pale green background. Their cells have a pale gray background and dark gray borders.

In the table on the left, the empty cell is shown. On the right, it is hidden:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | Hearts | | Diamonds | Spades |  |  |  | | --- | --- | |  | Hearts | | Diamonds | Spades | |  |

**Styling a table**

Make a new HTML document, doc3.html. Copy and paste the content from here, making sure that you scroll to get all of it:

<!DOCTYPE html>

<html>

<head>

<title>Sample document 3</title>

<link rel="stylesheet" href="style3.css">

</head>

<body>

<table id="demo-table">

<caption>Oceans</caption>

<thead>

<tr>

<th></th>

<th>Area</th>

<th>Mean depth</th>

</tr>

<tr>

<th></th>

<th>million km<sup>2</sup></th>

<th>m</th>

</tr>

</thead>

<tbody>

<tr>

<th>Arctic</th>

<td>13,000</td>

<td>1,200</td>

</tr>

<tr>

<th>Atlantic</th>

<td>87,000</td>

<td>3,900</td>

</tr>

<tr>

<th>Pacific</th>

<td>180,000</td>

<td>4,000</td>

</tr>

<tr>

<th>Indian</th>

<td>75,000</td>

<td>3,900</td>

</tr>

<tr>

<th>Southern</th>

<td>20,000</td>

<td>4,500</td>

</tr>

</tbody>

<tfoot>

<tr>

<th>Total</th>

<td>361,000</td>

<td></td>

</tr>

<tr>

<th>Mean</th>

<td>72,000</td>

<td>3,800</td>

</tr>

</tfoot>

</table>

</body>

</html>

Make a new stylesheet, style3.css. Copy and paste the content from here, making sure that you scroll to get all of it:

/\*\*\* Style for doc3.html (Tables) \*\*\*/

#demo-table {

font: 100% sans-serif;

background-color: #efe;

border-collapse: collapse;

empty-cells: show;

border: 1px solid #7a7;

}

#demo-table > caption {

text-align: left;

font-weight: bold;

font-size: 200%;

border-bottom: .2em solid #4ca;

margin-bottom: .5em;

}

/\* basic shared rules \*/

#demo-table th,

#demo-table td {

text-align: right;

padding-right: .5em;

}

#demo-table th {

font-weight: bold;

padding-left: .5em;

}

/\* header \*/

#demo-table > thead > tr:first-child > th {

text-align: center;

color: blue;

}

#demo-table > thead > tr + tr > th {

font-style: italic;

color: gray;

}

/\* fix size of superscript \*/

#demo-table sup {

font-size: 75%;

}

/\* body \*/

#demo-table td {

background-color: #cef;

padding:.5em .5em .5em 3em;

}

#demo-table tbody th:after {

content: ":";

}

/\* footer \*/

#demo-table tfoot {

font-weight: bold;

}

#demo-table tfoot th {

color: blue;

}

#demo-table tfoot th:after {

content: ":";

}

#demo-table > tfoot td {

background-color: #cee;

}

#demo-table > tfoot > tr:first-child td {

border-top: .2em solid #7a7;

}

Open the document in your browser. It should look very similar to this:

Compare the rules in the stylesheet with the displayed table, to ensure that you understand the effect of each rule. If you find a rule that you are not sure about, comment it out and refresh your browser to see what happens. Here are some notes about this table:

* The caption lies outside the table border.
* If you have a minimum point size set in Options, it might affect the superscript in km2.
* There are three empty cells. Two of them allow the table background to show through. The third has a background and a top border.
* The colons are added by the stylesheet.

### MEDIA

The purpose of CSS is to specify how documents are presented to the user. Presentation can take more than one form.

A document on a web site has a navigation area to allow the user to move around the site.

In the markup language, the navigation area's parent element has theidnav-area. (In [HTML5](https://developer.mozilla.org/en-US/docs/HTML/HTML5), this can be marked up with the [<nav>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav)element instead of[<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div)with anid attribute.)

When the document is printed the navigation area has no purpose, so the stylesheet removes it completely:

@media print {

#nav-area {display: none;}

}

Some of the common media types are:

|  |  |
| --- | --- |
| screen | Color computer display |
| print | Paged media |
| projection | Projected display |
| all | All media (the default) |

There are other ways to specify the media type of a set of rules.

The document's markup language might allow the media type to be specified when the stylesheet is linked to the document. For example, in HTML you can optionally specify the media type with a media attribute in the LINK tag.

In CSS you can use [@import](https://developer.mozilla.org/en-US/docs/Web/CSS/@import)at the start of a stylesheet to import another stylesheet from a URL, optionally specifying the media type.

By using these techniques you can separate style rules for different media types into different files. This is sometimes a useful way to structure your stylesheets.

### **Printing**

CSS has some specific support for printing and for paged media in general.

A [@page](https://developer.mozilla.org/en-US/docs/Web/CSS/@page)rule can set the page margins. For double-sided printing, you can specify the margins separately for@page:leftand@page:right.

For print media, you normally use appropriate length units like inches (in) and points (pt = 1/72 inch), or centimeters (cm) and millimeters (mm). It is equally appropriate to use ems (em) to match the font size, and percentages (%).

You can control how the content of the document breaks across page boundaries, by using the [page-break-before](https://developer.mozilla.org/en-US/docs/Web/CSS/page-break-before),[page-break-after](https://developer.mozilla.org/en-US/docs/Web/CSS/page-break-after)and[page-break-inside](https://developer.mozilla.org/en-US/docs/Web/CSS/page-break-inside)properties.

**Example**

This rule sets the page margins to one inch on all four sides:

@page {margin: 1in;}

This rule ensures that every H1 element starts on a new page:

h1 {page-break-before: always;}

### **User interfaces**

CSS has some special properties for devices that support a user interface, like computer displays. These make the document's appearance change dynamically as the user works with the interface.

There is no special media type for devices with user interfaces.

There are five special selectors:

|  |  |
| --- | --- |
| Selector | Selects |
| E[:hover](https://developer.mozilla.org/en-US/docs/Web/CSS/:hover) | Any E element that has the pointer over it |
| E[:focus](https://developer.mozilla.org/en-US/docs/Web/CSS/:focus) | Any E element that has keyboard focus |
| E[:active](https://developer.mozilla.org/en-US/docs/Web/CSS/:active) | The E element that is involved in the current user action |
| E[:link](https://developer.mozilla.org/en-US/docs/Web/CSS/:link) | Any E element that is a hyperlink to a URL that the user has not visited recently |
| E[:visited](https://developer.mozilla.org/en-US/docs/Web/CSS/:visited) | Any E element that is a hyperlink to a URL that the user has visited recently  The [cursor](https://developer.mozilla.org/en-US/docs/Web/CSS/cursor)property specifies the shape of the pointer: Some of the common shapes are as follows. Place your mouse over the items in this list to see the actual pointer shapes in your browser:  SelectorSelectspointerIndicating a linkwaitIndicating that the program cannot accept inputprogressIndicating that the program is working, but can still accept inputdefaultThe default (usually an arrow) . |

An[outline](https://developer.mozilla.org/en-US/docs/Web/CSS/outline)property creates an outline that is often used to indicate keyboard focus. Its values are similar to the[border](https://developer.mozilla.org/en-US/docs/Web/CSS/border)property, except that you cannot specify individual sides.

Some other features of user interfaces are implemented using attributes, in the normal way. For example, an element that is disabled or read-only has the disabled attribute or thereadonlyattribute. Selectors can specify these attributes like any other attributes, by using square brackets:[disabled]or[readonly].

**Example**

These rules specify styles for a button that changes dynamically as the user interacts with it:

.green-button {

background-color:#cec;

color:#black;

border:2px outset #cec;

}

.green-button[disabled] {

background-color:#cdc;

color:#777;

}

.green-button:active {

border-style: inset;

}

This wiki does not support a user interface on the page, so these buttons do not "click". Here are some static images to illustrate the idea:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Click Me | Click Me | Click Me | |  |  | | | disabled | normal | active | |

A fully functional button also has a dark outline around the entire button when it is the default, and a dotted outline on the face of the button when it has keyboard focus. It might also have a hover effect when the pointer is over it.

## **Printing a document**

1. Make a new HTML document,doc4.html. Copy and paste the content from here:

<!DOCTYPE html>

<html>

<head>

<title>Print sample</title>

<link rel="stylesheet" href="style4.css">

</head>

<body>

<h1>Section A</h1>

<p>This is the first section...</p>

<h1>Section B</h1>

<p>This is the second section...</p>

<div id="print-head">

Heading for paged media

</div>

<div id="print-foot">

Page:

</div>

</body>

</html>

Make a new stylesheet, style4.css. Copy and paste the content from here:

/\*\*\* Print sample \*\*\*/

/\* defaults for screen \*/

#print-head,

#print-foot {

display: none;

}

/\* print only \*/

@media print {

h1 {

page-break-before: always;

padding-top: 2em;

}

h1:first-child {

page-break-before: avoid;

counter-reset: page;

}

#print-head {

display: block;

position: fixed;

top: 0pt;

left:0pt;

right: 0pt;

font-size: 200%;

text-align: center;

}

#print-foot {

display: block;

position: fixed;

bottom: 0pt;

right: 0pt;

font-size: 200%;

}

#print-foot:after {

content: counter(page);

counter-increment: page;

}

} /\* end print only \*/