

Cascading style sheet(CSS)

reference: w3school

Prepared by: Er. Simanta Kasaju

CSS

- CSS was developed by Hakon wium lie in 1996.
- Cascading Style Sheets (CSS) is used to format the layout of a webpage.
- HTML was originally designed as a simple way of presenting information intended for sharing scientific documents and research papers online.

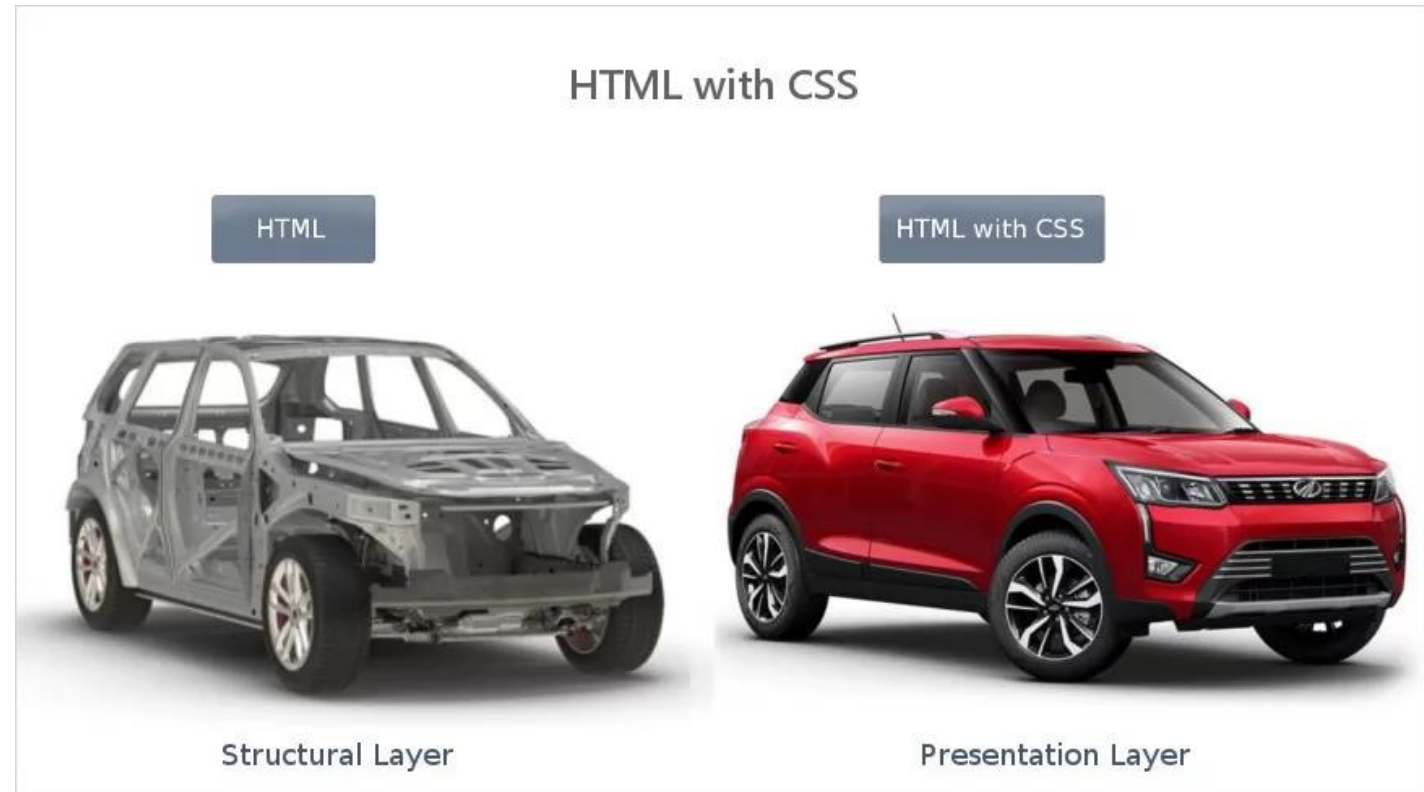
To improve web presentation capabilities CSS was introduced by W3C (World Wide Web Consortium). It was intended to allow web designers to define the look and feel of their web pages, and to separate content from document's layout

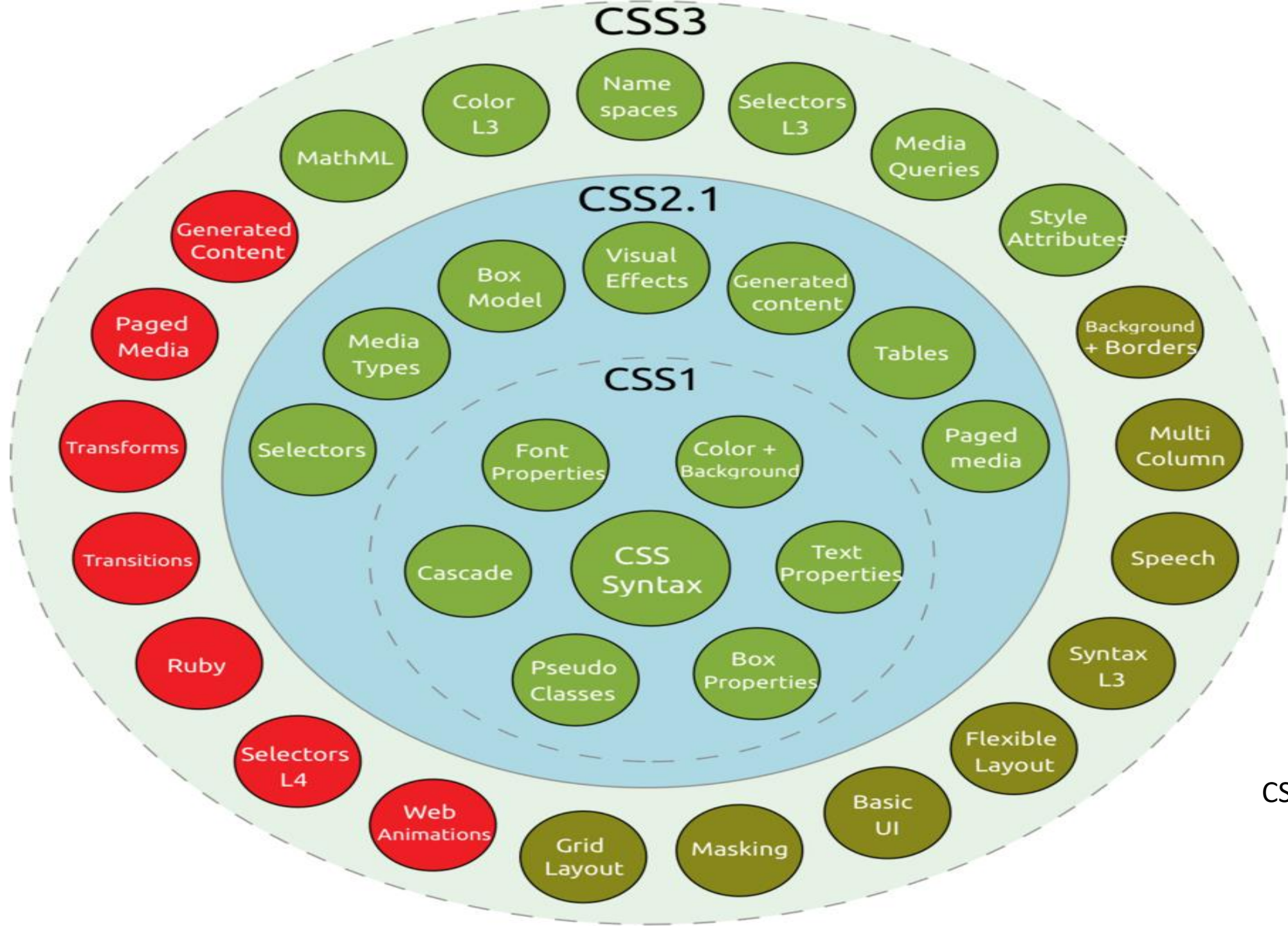
- With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

Cont.

- CSS stands for “Cascading Style Sheets.”
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.
- Cascading: refers to the procedure that determine which style will apply to a certain section, if you have more than one style rule.
- Style: how you want a certain part of your page to look. You can set things like color, margins, font etc for things like tables, paragraphs and headings.
- Sheets: the “sheets” are like templates, or a set of rules, for determine how the web page will look. So, CSS(all together) is a styling language-is set of rules to tell browsers how your web page should look.
- **Tip:** The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!

Imagine a web page without styling. A web page is written in HTML having a heading inside `<h1>` tag, paragraphs inside `<p>` element. And it doesn't contain any CSS. Thus, all the major web browser will exhibit your page in plain black unorganized text that will look awful. Therefore, you would require the cascading style sheet to beautify your web page. Cascading Style Sheet can transform the color, size, layout, typography, background, foreground, textures, and images.





CSS versions

CSS1 (1996) enabled users to set font size; align text center, left, or right; set body margins; and apply background and foreground colors to page elements.

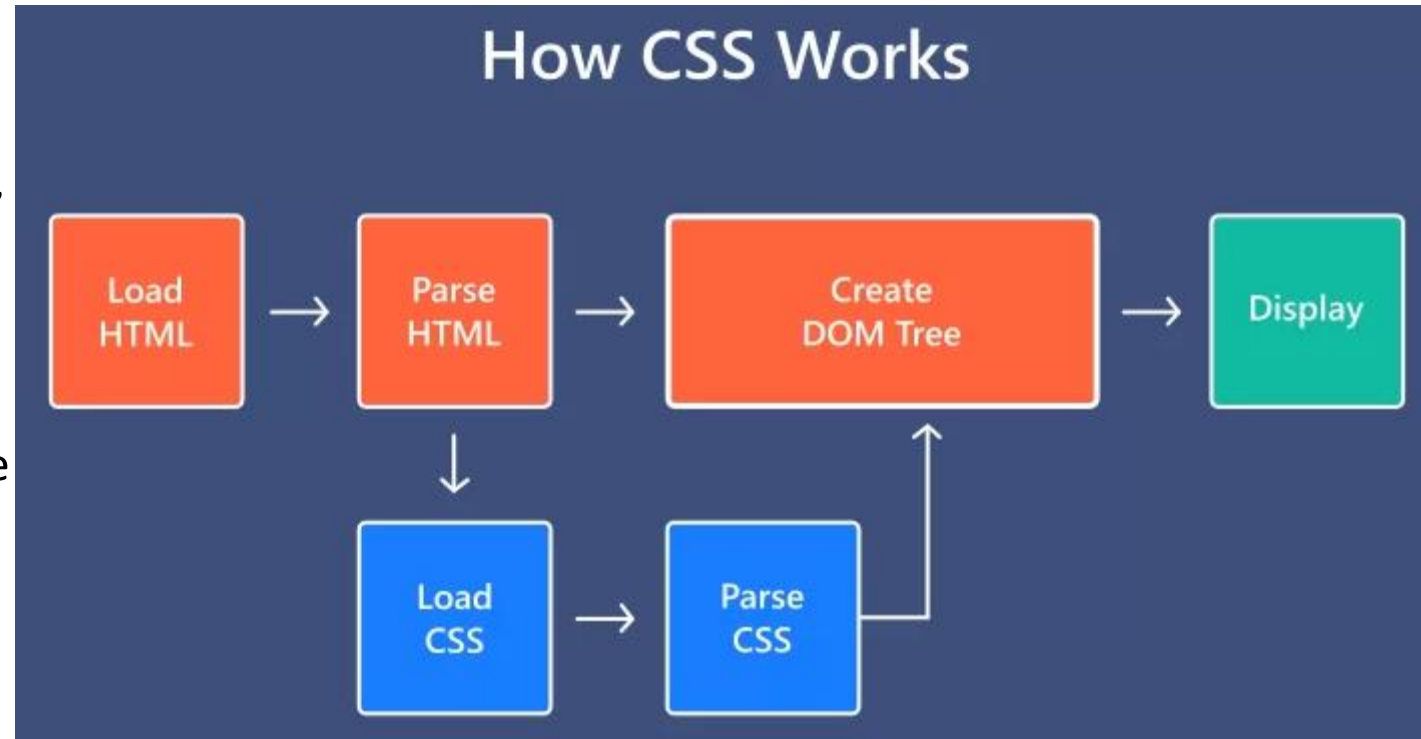
CSS2 (1998) included features such as design styles for different output devices such as print media and aural devices, and controlling the appearance and behavior of browser features.

CSS3 (2005) includes text effects such as drop shadows and Web fonts, semitransparent colors, box outlines, and rotating page elements.

About a web page styling and layout, Cascading Style Sheet has a totally different approach. Every time an HTML document is displayed on a browser, the content comes with the style information. Basically, the HTML file contains the content of a page and the style sheet has the information about the style of a page. Hence, the main purpose of Cascade Style Sheets is to allow the elements to appear in the HTML document. And these specified standards direct how the content will be executed.

you ever wish to give distinct styling to each element of your web page, there you will have to use CSS for every element individually. CSS gives you full control over the web page and lets you carry out modifications wherever you choose. For example, you prefer to highlight (making its colour to and text in italic) a particular paragraph of a web page. Then you would need to include the following code to a web page:

```
P { color: pink; font-weight: italic; }
```



Why Use CSS?

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

```
<style>
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
```

My First CSS Example

This is a paragraph.

Advantages of CSS

1. **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.
2. **Pages load faster:** If you are using CSS you don't 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.
3. **Easy maintenance:** To make global change, simply change the style and all elements in all the web pages will be updated automatically.
4. **Superior styles to HTML:** CSS has a much wider area of attributes than HTML so you can give a far better look to your HTML page in comparison to HTML attributes.
5. **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 PDA and cell phones or for printing.
6. **Global web standards:** Now, HTML attributes are being deprecated and it is being recommended to use CSS. so it's a good to start using CSS in all the HTML page to make them compatible to future browsers.
7. Easy for the user to customize the online page
8. It reduces the file transfer size.

Disadvantages of CSS

1. Cross-Browser Issues

- Implementing initial CSS changes on a website is easy on the developer's end. However, after the changes have been made, you will have to confirm the compatibility if the CSS is displaying similar change effects on all the browsers. This is simply due to the fact that CSS works differently on different browsers.

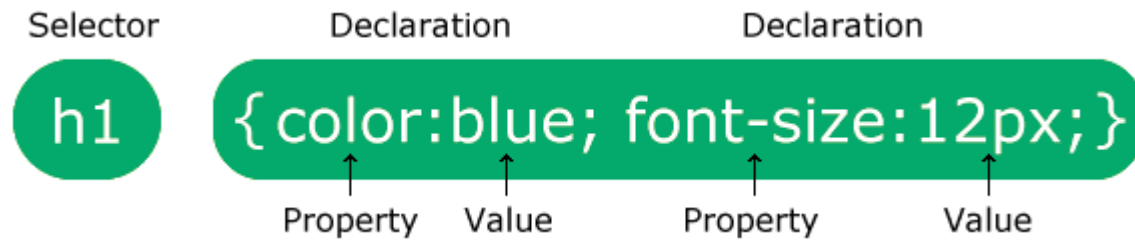
2. Confusion Due to Its Many Levels

- The programming language world in itself is crazily complicated for non-developers and beginners. To add to that, different levels of CSS i.e. CSS; CSS 2; CSS 3 can be quite confusing for the mentioned lot.

3. Vulnerable

- If you have worked with CSS, you probably know that it is easily accessible because of its open text-based system. An accident or a mere act of mischief with the files can end up disrupting the display and formatting of your entire website. It would only require a read/write access to the intended website to override the changes.

CSS Syntax



- A CSS rule set contains a selector and a declaration block.
- **Selector:** Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.
- **Declaration Block:** The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:
 1. color: blue;
 2. font-size: 12 px;
- Each declaration contains a property name and value, separated by a colon.
- **Property:** A Property is a type of attribute of HTML element. It could be color, border etc.
- **Value:** Values are assigned to CSS properties. In the above example, value "blue" is assigned to color property.

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>).

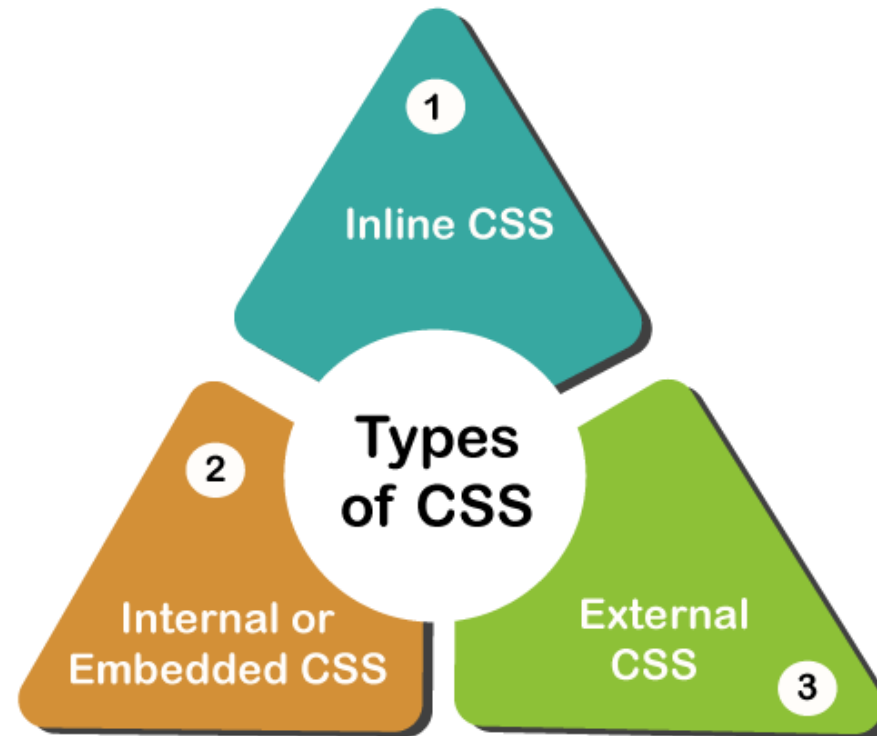
color is a property, and red is the property value

text-align is a property, and center is the property value

Hello World!

These paragraphs are styled with CSS.

Types of CSS(Linking style of CSS with HTML)



1. Inline CSS

- Inline CSS is used to style a specific HTML element. Add a style attribute to each HTML tag without using the selectors. Managing a website may be difficult if we use only inline CSS. However, Inline CSS in HTML is useful in some situations. We have no access to the CSS files or to apply styles to elements.
- An inline style loses many of the advantages of a style sheet.
- To use inline styles, add the style attribute to the relevant tag.
- The style attribute can contain any CSS property.
- In the following example, we have used the inline CSS in <p> and <h1> tags.

```
<!DOCTYPE html>
<html>
<body style="background-color:white;">
<h1 style="color:red;padding:20px;">CSS Tutorials</h1>
<p style="color:blue;">It will be useful here.</p>
</body>
</html>
```

Cont.

Pros of inline CSS:

- We can create CSS rules on the HTML page.
- We cannot create and upload a separate document in inline CSS.

• Cons of inline CSS:

- Inline CSS, adding **CSS** rules to HTML elements is **time-consuming** and **messes** up the HTML structure.
- It styles multiple elements at the same time which can affect the page size and download time of the page.

2. Internal CSS(Embedded Style)

- The Internal CSS has <style> tag in the <head> section of the HTML document. This CSS style is an effective way to style single pages. Using the CSS style for multiple web pages is time-consuming because we require placing the style on each web page.

- We can use the internal CSS by using the following steps:

1. Firstly, open the HTML page and locate the <head>
2. Put the following code after the <head>
3. Add the rules of CSS in the new line.
4. Close the style tag.

Pros of Internal CSS

- **Internal CSS** cannot upload multiple files when we add the code with the HTML page.

Cons of Internal CSS:

- Adding code in the **HTML** document will reduce the **page size** and **loading time** of the webpage.

Example:

```
<style type="text/css">
body {
    background-color: black;
}
h1 {
    color: white;
    padding: 50px;
}
</style>
```

3. External CSS

- In external CSS, we link the web pages to the external .css file. It is created by text editor. The CSS is more efficient method for styling a website. By editing the .css file, we can change the whole site at once.
- To use the external CSS, follow the steps, given below:

```
<head>
```

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

```
</head>
```

- An external style sheet can be written in any text editor.
- Your style sheet should be saved with a .css extension.
- Eg. `hr{ color : blue; } p{ margin-left : 20px; } body{ background-image : url("images/back40.gif"); }`

Cont.

Pros of External CSS:

- Our files have a cleaner structure and smaller in size.
- We use the same **.css** file for multiple web pages in external CSS.

Cons of External CSS:

- The pages cannot be delivered correctly before the external CSS is loaded.
- In External CSS, uploading many CSS files can increase the download time of a website.

4.Import method

- Imported CSS- @import Rule
- @import is used to import an external stylesheet in a manner similar to the link element.
- Syntax

```
<head>
```

```
@import url("mystyle.css");
```

```
</head>
```

```
1 <html>
2 <head>
3   <title>Web Page</title>
4
5   <link href="style.css" rel="stylesheet" type="text/css">
6
7   <style>
8
9     @import "../template/css/bootstrap.css";
10    @import "../template/style.css";
11
12    .main{
13      max-width:1200px;
14      margin:0 auto;
15      padding:0;
16    }
17  </style>
18
19 </head>
20
21 <body>
22   <div class="main">
23
24     <div class="wp_first_row row" style="background-color:#f0f0f0; padding:80px 0; text-align:center">
25
26       <div class="wp_first_row_col">
27         <h1>Services That We Offer</h1>
28         <p>Our moving company trusted worldwide, provides exceptional household moving services.</p>
29       </div>
30     </div>
31   </div>
32 </body>
33 </html>
```

External CSS

Import CSS

Internal CSS

Inline CSS

CSS rules Overriding

- Any inline style sheet takes highest priority. So it will override any rule defined in `<style>..</style>` tags or rules defined in any external style sheet file.
- Any rule defined in `<style>..</style>` tags will override rules defined in any external style sheet file.
- Any rule defined in external style sheet file takes lowest priority & rules defined in this file will be applied only when above rules are not applicable.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href=
"mystyle.css">
<style>
body {background-color: linen;}
</style>
</head>
<body style="background-color: lavender">

<h1>Multiple Styles Will Cascade into One</h1>
<p>Here, the background color of the page is set with inline
CSS, and also with an internal CSS, and also with an external
CSS.</p>
<p>Try experimenting by removing styles to see how the
cascading stylesheets work (try removing the inline CSS first,
then the internal, then the external).</p>

</body>
</html>
```

Due to Inline

Multiple Styles Will Cascade into One

Here, the background color of the page is set with inline CSS, and also with an internal CSS, and also with an external CSS.

Try experimenting by removing styles to see how the cascading stylesheets work (try removing the inline CSS first, then the internal, then the external).

Multiple Styles Will Cascade into One

Here, the background color of the page is set with inline CSS, and also with an internal CSS, and also with an external CSS.

Try experimenting by removing styles to see how the cascading stylesheets work (try removing the inline CSS first, then the internal, then the external).

Due to Internal

Due to External

Multiple Styles Will Cascade into One

Here, the background color of the page is set with inline CSS, and also with an internal CSS, and also with an external CSS.

Try experimenting by removing styles to see how the cascading stylesheets work (try removing the inline CSS first, then the internal, then the external).

CSS simple basic selectors

- select elements based on name, id, class

1.The CSS element Selector

- The element selector selects HTML elements based on the element name.
- Example
- Here, all <p> elements on the page will be center-aligned, with a red text color:

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

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

```
<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

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

Every paragraph will be affected by the style.

Me too!

And me!

2. The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element!

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

We can also do this

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

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

Note: A class name cannot start with a number!

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#para1 {
```

```
  text-align: center;
```

```
  color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p id="para1">Hello World!</p>
```

```
<p>This paragraph is not affected by the style.</p>
```

```
</body>
```

```
</html>
```

Hello World!

This paragraph is not affected by the style.

3. The CSS class Selector

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

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

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

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

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

HTML elements can also refer to more than one class.

```
<p class="center large">This paragraph refers  
to two classes.</p>
```

Example 1

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.center {
```

```
  text-align: center;
```

```
  color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1 class="center">Red and center-aligned heading</h1>
```

```
<p class="center">Red and center-aligned paragraph.</p>
```

```
</body>
```

```
</html>
```

Red and center-aligned heading

Red and center-aligned paragraph.

Example 2

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

This heading will not be affected

This paragraph will be red and center-aligned.

```
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-
aligned.</p>

</body>
</html>
```

Example3

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.center {
```

```
  text-align: center;
```

```
  color: red;
```

```
}
```

```
p.large {
```

```
  font-size: 300%;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1 class="center">This heading will not be affected</h1>
```

```
<p class="center">This paragraph will be red and center-aligned.</p>
```

```
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>
```

```
</body>
```

```
</html>
```

This heading will not be affected

This paragraph will be red and center-aligned.

This paragraph will be red, center-aligned, and in a large font-size.

4.The CSS Universal Selector

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

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

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

```
<h1>Hello world!</h1>
```

```
<p>Every element on the page will be affected by the
style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

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

Hello world!

Every element on the page will be affected by the style.

Me too!

And me!

5.The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.
Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

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



Equivalent to

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

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

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

We can group id selector as well

```
#content, #footer, #supplement {  
  position: absolute;  
  left: 510px;  
  width: 200px;  
}  
#content, #footer, .supplement {  
  position: absolute;  
  left: 510px;  
  width: 200px;  
}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1, h2, p {
```

```
  text-align: center;
```

```
  color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Hello World!</h1>
```

```
<h2>Smaller heading!</h2>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

Hello World!

Smaller heading!

This is a paragraph.

Multiple style sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more Specific style sheet.

For example an external style sheet has these properties for h1 selector

```
h1{  
color:red;  
text-align:left;  
Font-size:8px;  
}
```

And an external style sheet has these properties for h2 selector:

```
h1{  
text-align:right;  
Font-size:20px;  
}
```

If the page with the internal style sheet also links to the external style sheet the properties will be

```
h1{  
color:red;  
text-align:right;  
Font-size:20px;
```

CSS comment

```
/* This is comment */
```


Other selectors

CSS Combinators

A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

1. descendant selector (space)
2. child selector (>)
3. adjacent sibling selector (+)
4. general sibling selector (~)

Descendant Selector: The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

```
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>Descendant Selector</h2>
<p>The descendant selector matches all elements that are
descendants of a specified element.</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
</body>
```

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

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

Child Selector (>)

The child selector selects all elements that are the children of a specified element.

The following example selects all <p> elements that are children of a <div> element:

```
<style>
div > p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>Child Selector</h2>
<p>The child selector (>) selects all elements that are the
children of a specified element.</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
  <p>Paragraph 4 in the div.</p>
</div>
<p>Paragraph 5. Not in a div.</p>
<p>Paragraph 6. Not in a div.</p>
```

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

Child Selector

The child selector (>) selects all elements that are the children of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4 in the div.

Paragraph 5. Not in a div.

Paragraph 6. Not in a div.

Adjacent Sibling Selector (+)

The adjacent sibling selector is used to select an element that is directly after another specific element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects the first <p> element that are placed immediately after <div> elements:

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

```

<style>div + p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>Adjacent Sibling Selector</h2>
<p>The + selector is used to select an element that is directly
after another specific element.</p>
<p>The following example selects the first p element that are
placed immediately after div elements:</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>
<p>Paragraph 3. After a div.</p>
<p>Paragraph 4. After a div.</p>
<div>
  <p>Paragraph 5 in the div.</p>
  <p>Paragraph 6 in the div.</p>
</div>
<p>Paragraph 7. After a div.</p>
<p>Paragraph 8. After a div.</p>

```

Adjacent Sibling Selector

The + selector is used to select an element that is directly after another specific element.

The following example selects the first p element that are placed immediately after div elements:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. After a div.

Paragraph 4. After a div.

Paragraph 5 in the div.

Paragraph 6 in the div.

Paragraph 7. After a div.

Paragraph 8. After a div.

General Sibling Selector (~)

The general sibling selector selects all elements that are siblings of a specified element. The following example selects all <p> elements that are siblings of <div> elements:

```
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>General Sibling Selector</h2>
<p>The general sibling selector (~) selects all
elements that are siblings of a specified element.</p>
<p>Paragraph 1.</p>
<div>
  <p>Paragraph 2.</p>
</div>
<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>
```

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

General Sibling Selector

The general sibling selector (~) selects all elements that are siblings of a specified element.

Paragraph 1.

Paragraph 2.

Paragraph 3.

Some code.

Paragraph 4.

Summary

All CSS Combinator Selectors

Selector	Example	Example description
<u><i>element element</i></u>	div p	Selects all <p> elements inside <div> elements
<u><i>element>element</i></u>	div > p	Selects all <p> elements where the parent is a <div> element
<u><i>element+element</i></u>	div + p	Selects the first <p> element that are placed immediately after <div> elements
<u><i>element1~element2</i></u>	p ~ ul	Selects every element that are preceded by a <p> element

CSS Pseudo-classes

What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

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

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective! a:active MUST come after a:hover in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.


```

<!DOCTYPE html>
<html>
<head>
<style>
/* 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;
}
</style>
</head>
<body>
<h2>CSS Links</h2>
<p><b><a
href="Example.html"
target="_blank">This is a
link</a></b></p>
<p><b>Note:</b> a:hover
MUST come after a:link and
a:visited in the CSS definition
in order to be effective.</p>
<p><b>Note:</b> a:active
MUST come after a:hover in
the CSS definition in order to
be effective.</p>
</body>
</html>

```

CSS Links

[This is a link](#)

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

CSS Links

[This is a link](#)

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

CSS Links

[This is a link](#)

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

CSS Links

[This is a link](#)

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

Pseudo-classes and CSS Classes

Pseudo-classes can be combined with CSS classes:

When you hover over the link in the example, it will change color:

```
a.highlight:hover {  
    color: #ff0000;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a.highlight:hover {  
    color: #ff0000;  
}  
</style>  
</head>  
<body>  
<h2>Pseudo-classes and CSS Classes</h2>  
<p>When you hover over the first link below, it will change  
color:</p>  
<p><a class="highlight" href="css_syntax.asp">CSS  
Syntax</a></p>  
<p><a href="default.asp">CSS Tutorial</a></p>  
</body>  
</html>
```

Pseudo-classes and CSS Classes

When you hover over the first link below, it will change color:

[CSS Syntax](#)

[CSS Tutorial](#)

Hover on <div>

An example of using the :hover pseudo-class on a <div> element:

```
<style>
div {
  background-color: green;
  color: white;
  padding: 25px;
  text-align: center;
}
```

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

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Mouse over the div element below to change its
background color:</p>
```

```
<div>Mouse Over Me</div>
```

Mouse over the div element below to change its background color:



Mouse over the div element below to change its background color:



Simple Tooltip Hover

Hover over a `<div>` element to show a `<p>` element (like a tooltip):

```
<html>
<head>
<style>
p {
  display: none;
  background-color: yellow;
  padding: 20px;
}
div:hover p {
  display: block;
}
</style>
</head>
<body>
<div>Hover over me i want to say you something
  <p>You are Beautiful</p>
</div>
</body>
</html>
```

Hover over me i want to say you something

Hover over me i want to say you something

You are Beautiful

CSS - The :first-child Pseudo-class

The :first-child pseudo-class matches a specified element that is the first child of another element.

Match the first <p> element

In the following example, the selector matches any <p> element that is the first child of any element:

```
p:first-child {  
  color: blue;  
}
```

Match the first <i> element in all <p> elements

In the following example, the selector matches the first <i> element in all <p> elements:

```
p i:first-child {  
  color: blue;  
}
```

Match all <i> elements in all first child <p> elements

In the following example, the selector matches all <i> elements in <p> elements that are the first child of another element:

```
p:first-child i {  
  color: blue;  
}
```

```

<style>
p:first-child {
  color: blue;
}
</style>
</head>
<body>
<p>This is some text.</p>
<p>This is some text.</p>

```

This is some text.

This is some text.

```

<style>
p i:first-child {
  color: blue;
}
</style>
</head>
<body>
<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>

```

I am a *strong* person. I am a strong person.

I am a *strong* person. I am a strong person.

```

<style>
p:first-child i {
  color: blue;
}
</style>
</head>
<body>
<p>I am a <i>strong</i> person. I am a
<i>strong</i> person.</p>
<p>I am a <i>strong</i> person. I am a
<i>strong</i> person.</p>

```

I am a *strong* person. I am a strong person.

I am a strong person. I am a strong person.

Mouse over the links and watch them change layout:

This link changes color 

This link changes font-size

This link changes background-color

This link changes font-family

This link changes text-decoration

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

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

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

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

a.five:link {color:#ff0000;text-decoration:none;}
a.five:visited {color:#0000ff;text-decoration:none;}
a.five:hover {text-decoration:underline;}
</style>
</head>
<body>
```

```
<p>Mouse over the links and watch them change layout:</p>

<p><b><a class="one" href="default.asp" target="_blank">This
link changes color</a></b></p>
<p><b><a class="two" href="default.asp" target="_blank">This
link changes font-size</a></b></p>
<p><b><a class="three" href="default.asp"
target="_blank">This link changes background-
color</a></b></p>
<p><b><a class="four" href="default.asp" target="_blank">This
link changes font-family</a></b></p>
<p><b><a class="five" href="default.asp" target="_blank">This
link changes text-decoration</a></b></p>

</body>
</html>
```


All CSS Pseudo Classes

Selector	Example	Example description
<u>:active</u>	a:active	Selects the active link
<u>:checked</u>	input:checked	Selects every checked <input> element
<u>:disabled</u>	input:disabled	Selects every disabled <input> element
<u>:empty</u>	p:empty	Selects every <p> element that has no children
<u>:enabled</u>	input:enabled	Selects every enabled <input> element
<u>:first-child</u>	p:first-child	Selects every <p> elements that is the first child of its parent
<u>:first-of-type</u>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<u>:focus</u>	input:focus	Selects the <input> element that has focus
<u>:hover</u>	a:hover	Selects links on mouse over
<u>:in-range</u>	input:in-range	Selects <input> elements with a value within a specified range
<u>:invalid</u>	input:invalid	Selects all <input> elements with an invalid value
<u>:lang(<i>language</i>)</u>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<u>:last-child</u>	p:last-child	Selects every <p> elements that is the last child of its parent
<u>:last-of-type</u>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<u>:link</u>	a:link	Selects all unvisited links

<u>:not(selector)</u>	:not(p)	Selects every element that is not a <p> element
<u>:nth-child(n)</u>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<u>:nth-last-child(n)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<u>:nth-last-of-type(n)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<u>:nth-of-type(n)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u>:optional</u>	input:optional	Selects <input> elements with no "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects <input> elements with a value outside a specified range
<u>:read-only</u>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<u>:read-write</u>	input:read-write	Selects <input> elements with no "readonly" attribute
<u>:required</u>	input:required	Selects <input> elements with a "required" attribute specified
<u>:root</u>	root	Selects the document's root element
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<u>:valid</u>	input:valid	Selects all <input> elements with a valid value
<u>:visited</u>	a:visited	Selects all visited links

```
<style>
input:focus {
  background-color: yellow;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form action="/action_page.php" method="get">
```

```
  First name: <input type="text" name="fname"><br>
```

```
  Last name: <input type="text" name="lname"><br>
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

First name:

Last name:

Submit

First name:

Last name:

Submit

CSS Pseudo-elements

What are 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

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:

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

```
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}
```

Note: The `::first-line` pseudo-element can only be applied to block-level elements.

The following properties apply to the `::first-line` pseudo-element:

font properties

color properties

background properties

word-spacing

letter-spacing

text-decoration

vertical-align

text-transform

line-height

clear

Notice the double colon notation - ::first-line versus :first-line

The double colon replaced the single-colon notation for pseudo-elements in CSS3. This was an attempt from W3C to distinguish between **pseudo-classes** and **pseudo-elements**.

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.

```
<!DOCTYPE html>
```

<html>

<head>

<style>

```
p::first-line {
```

```
color: #ff0000;
```

font-variant: small-caps;

background-color: skyblue;

}

</style>

</head>

<body>

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.

</body>

</html>

The ::first-letter Pseudo-element

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

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

Note: The ::first-letter pseudo-element can only be applied to block-level elements.

The following properties apply to the ::first-letter pseudo-element:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

```
p::first-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p::first-letter {
  color: Blue;
  font-size: 100px;
  font-style: italic;
}
</style>
</head>
<body>
```

Y

ou can use the ::first-letter pseudo-element to add a special effect to the first character of a text!

```
<p>You can use the ::first-letter pseudo-element to add a
special effect to the first character of a text!</p>
```

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

Pseudo-elements and CSS Classes

Pseudo-elements can be combined with CSS classes:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.intro::first-letter {
  color: #ff0000;
  font-size: 200%;
}
</style>
</head>
<body>

<p class="intro">This is an introduction.</p>
<p>This is a paragraph with some text. A bit more text
even.</p>

</body>
</html>
```

```
p.intro::first-letter {
  color: #ff0000;
  font-size: 200%;
}
```

This is an introduction.

This is a paragraph with some text. A bit more text even.

Multiple Pseudo-elements

Several pseudo-elements can also be combined.

In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color:

```
<style>
p::first-letter {
  color: #ff0000;
  font-size: xx-large;
}
p::first-line {
  color: #0000ff;
  font-variant: small-caps;
}
</style>
</head>
<body>
<p>You can combine the ::first-letter and ::first-line pseudo-
elements to add a special effect to the first letter and the first
line of a text!</p>
```

YOU CAN COMBINE THE **::FIRST-LETTER** AND **::FIRST-LINE PSEUDO-ELEMENTS** TO ADD A
special effect to the first letter and the first line of a text!

CSS - The ::before Pseudo-element

The ::before pseudo-element can be used to insert some content before the content of an element.

The following example inserts an image before the content of each <h1> element:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1::after {
  content: url(smiley.gif);
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>The ::after pseudo-element inserts content after the
content of an element.</p>
<h1>This is a heading</h1>
</body>
</html>
```

Use before for 2nd
pic


This is a heading 

The ::after pseudo-element inserts content after the content of an element.

This is a heading 

 **This is a heading**

The ::before pseudo-element inserts content before the content of an element.

 **This is a heading**

CSS - The ::marker Pseudo-element

The ::marker pseudo-element selects the markers of list items.

The following example styles the markers of list items:

```
<style>
::marker {
  color: red;
  font-size: 23px;
}
```

- Coffee
- Tea
- Milk

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Milk</li>
```

```
</ul>
```

```
<ol>
```

```
<li>First</li>
```

```
<li>Second</li>
```

```
<li>Third</li>
```

```
</ol>
```

1. First
2. Second
3. Third

```
::marker {
  color: red;
  font-size: 23px;
}
```

CSS - The ::selection Pseudo-element

The ::selection pseudo-element matches the portion of an element that is selected by a user. The following CSS properties can be applied to ::selection: color, background, cursor, and outline. The following example makes the selected text red on a yellow background:

```
<!DOCTYPE html>
<html>
<head>
<style>
::selection {
  color: red;
  background: yellow;
}
</style>
</head>
<body>
<h1>Select some text on this page:</h1>
<p>This is a paragraph.</p>
<div>This is some text in a div element.</div>
<p><strong>Note:</strong> Firefox supports an alternative, the
::-moz-selection property.</p>
```

Select some text on this page:

This is a paragraph.

This is some text in a div element.

Note: Firefox supports an alternative, the ::-moz-selection property.

Select some text on this page:

This is a paragraph.

This is some text in a div element.

Note: Firefox supports an alternative, the ::-moz-selection property.

All CSS Pseudo Elements

Selector	Example	Example description
<u>::after</u>	p::after	Insert something after the content of each <p> element
<u>::before</u>	p::before	Insert something before the content of each <p> element
<u>::first-letter</u>	p::first-letter	Selects the first letter of each <p> element
<u>::first-line</u>	p::first-line	Selects the first line of each <p> element
<u>::marker</u>	::marker	Selects the markers of list items
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user

CSS [attribute] Selector

The [attribute] selector is used to select elements with a specified attribute. The following example selects all <a> elements with a target attribute:

```
<!DOCTYPE html>
<html>
<head>
<style>
a[target] {
  background-color: yellow;
}
```

```
</style>
</head>
```

```
<body>
```

```
<h2>CSS [attribute] Selector</h2>
```

```
<p>The links with a target attribute gets a yellow
background:</p>
```

```
<a href="https://www.w3schools.com">w3schools.com</a>
```

```
<a href="http://www.disney.com"
target="_blank">disney.com</a>
```

```
<a href="http://www.wikipedia.org"
target="_top">wikipedia.org</a>
```

```
a[target] {
  background-color: yellow;
}
```

CSS [attribute] Selector

The links with a target attribute gets a yellow background:

[w3schools.com](https://www.w3schools.com) [disney.com](http://www.disney.com) [wikipedia.org](http://www.wikipedia.org)

CSS [attribute="value"] Selector

The [attribute="value"] selector is used to select elements with a specified attribute and value.

The following example selects all <a> elements with a target="_blank" attribute:

```
<style>
a[target=_blank] {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>CSS [attribute="value"] Selector</h2>
<p>The link with target="_blank" gets a yellow
background:</p>
```

```
<a href="https://www.w3schools.com">w3schools.com</a>
<a href="http://www.disney.com"
target="_blank">disney.com</a>
<a href="http://www.wikipedia.org"
target="_top">wikipedia.org</a>
</body>
</html>
```

CSS [attribute="value"] Selector

The link with target="_blank" gets a yellow background:

[w3schools.com](https://www.w3schools.com) [disney.com](http://www.disney.com) [wikipedia.org](http://www.wikipedia.org)

CSS [attribute]="value" Selector

The [attribute]="value" selector is used to select elements with the specified attribute starting with the specified value. The following example selects all elements with a class attribute value that begins with "top":

Note: The value has to be a whole word, either alone, like class="top", or followed by a hyphen(-), like class="top-text"!

```
<style>
[class|=top] {
  background: yellow;
}
</style>
</head>
<body>
```

```
<h2>CSS [attribute]="value" Selector</h2>
```

```
<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>
```

CSS [attribute]="value" Selector

Welcome

Hello world!

Are you learning CSS?

CSS [attribute^="value"] Selector

The [attribute^="value"] selector is used to select elements whose attribute value begins with a specified value. The following example selects all elements with a class attribute value that begins with "top":

Note: The value does not have to be a whole word!

```
<!DOCTYPE html>
<html>
<head>
<style>
[class^="top"] {
  background: yellow;
}
</style>
</head>
<body>
<h2>CSS [attribute^="value"] Selector</h2>

<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>
```

CSS [attribute^="value"] Selector

Welcome

Hello world!

Are you learning CSS?

CSS [attribute\$="value"] Selector

The [attribute\$="value"] selector is used to select elements whose attribute value ends with a specified value.

The following example selects all elements with a class attribute value that ends with "test":

Note: The value does not have to be a whole word!

```
<style>
[class$="test"] {
  background: yellow;
}
</style>
</head>
<body>
<h2>CSS [attribute$="value"] Selector</h2>
<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytest">This is some text in a paragraph.</p>

</body>
</html>
```

CSS [attribute\$="value"] Selector

The first div element.

The second div element.

The third div element.

This is some text in a paragraph.

CSS [attribute*="value"] Selector

The [attribute*="value"] selector is used to select elements whose attribute value contains a specified value.

The following example selects all elements with a class attribute value that contains "te":

Note: The value does not have to be a whole word!

```
<!DOCTYPE html>
<html>
<head>
<style>
[class*="te"] {
  background: yellow;
}
</style>
</head>
<body>
<h2>CSS [attribute*="value"] Selector</h2>
<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytest">This is some text in a paragraph.</p>
</body>
</html>
```

CSS [attribute*="value"] Selector

The first div element.

The second div element.

The third div element.

This is some text in a paragraph.

summary

All CSS Attribute Selectors

Selector	Example	Example description
<code>[<u>attribute</u>]</code>	<code>[target]</code>	Selects all elements with a target attribute
<code>[<u>attribute=</u><i>value</i>]</code>	<code>[target=_blank]</code>	Selects all elements with target="_blank"
<code>[<u>attribute~</u><i>value</i>]</code>	<code>[title~flower]</code>	Selects all elements with a title attribute containing the word "flower"
<code>[<u>attribute </u><i>value</i>]</code>	<code>[lang =en]</code>	Selects all elements with a lang attribute value starting with "en"
<code>[<u>attribute^</u><i>value</i>]</code>	<code>a[href^="https"]</code>	Selects every <a> element whose href attribute value begins with "https"
<code>[<u>attribute\$</u><i>value</i>]</code>	<code>a[href\$=".pdf"]</code>	Selects every <a> element whose href attribute value ends with ".pdf"
<code>[<u>attribute*</u><i>value</i>]</code>	<code>a[href*="w3schools"]</code>	Selects every <a> element whose href attribute value contains the substring "w3schools"

Summarize: Not including pseudo

Selectors	Description
h1, p, span etc.	element selector
.className	class selector
#idName	id selector
*	Universal selector (selects everything)
h1.className	select h1 with class 'className'
h1#idName	select h1 with id 'idName'
p span	descendant selector (select span which is inside p)
p > span	child selector ('span' which is direct descendant of 'p')
h1, h2, p	group selection (select h1, h2 and p)
span[my_id]	select 'span' with attribute 'my_id'
span[my_id=m_span]	select 'span' with attribute 'my_id=m_span'

CSS background

1.CSS background-color

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

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

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

We can add opacity as well where 0 means completely transparent and 1 means opaque

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

you can use an RGB color value with an **alpha** channel (RGBA) - which specifies the opacity for a color.

An RGBA color value is specified with: `rgba(red, green, blue, alpha)`. The *alpha* parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

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

```
<style>
h1 {
  background-color: green;
}
div {
  background-color: lightblue;
}
p {
  background-color: yellow;
}
</style>
</head>
<body>
<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>
```

CSS background-color example!

This is a text inside a div element.

This paragraph has its own background color.

We are still in the div element.

2.CSS background-image

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

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



Example

```
<!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>
```


3.CSS Background Repeat

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

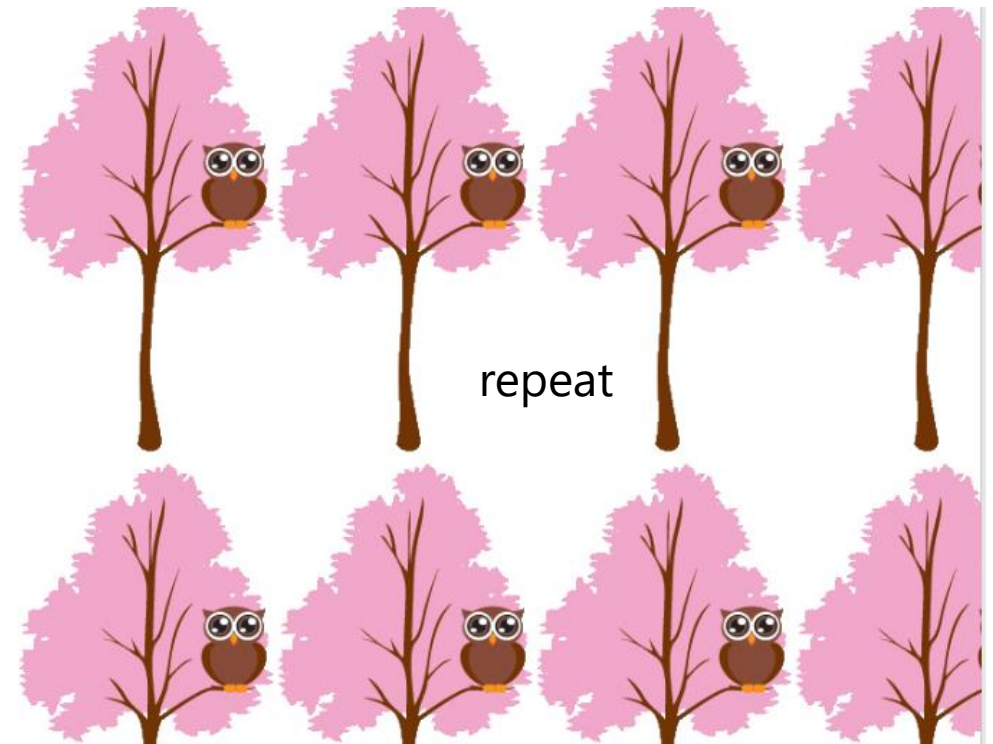


No repeat

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

Possible Values

```
background-repeat: repeat;  
background-repeat: no-repeat;  
background-repeat: repeat-x;  
background-repeat: repeat-y;
```



4.CSS Background position

```
body {  
    background-image: url("img_tree.png");  
    background-position: right top;  
}
```

Values(we can also combine them)

- 1.center
- 2.top
- 3.bottom
- 4.left
- 5.right

5.CSS background-attachment

```
body {  
    background-image: url("img_tree.png");  
    background-attachment: fixed;  
}
```

```
body {  
    background-image: url("img_tree.png");  
    background-attachment: scroll;  
}
```

Values(we can also combine them)


- 1.Fixed
- 2.scroll

CSS background - Shorthand property

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

Instead of writing:

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



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

The background-size property

Can be following:

- cover : fits & no empty space remains
- contain : fits & image is fully visible
- auto : display in original size
- {{width}} : set width & height will be set automatically
- {{width}} {{height}} : set width & height

```
body {  
  background-size: auto;  
}
```

Summarize

Background Properties

Property	Description	Possible Values	Examples
background-attachment	Declares the attachment of a background image (to scroll with the page content or be in a fixed position).	<i>fixed</i> <i>scroll</i>	<i>div { background-attachment:fixed; }</i> <i>div { background-attachment:scroll; }</i>
background-color	Declares the background color.	Valid color names, RGB values, hexadecimal notation.	<i>div { background-color:green; }</i> <i>div { color:#00FF00; }</i>
background-image	Declares the background	URL values.	<i>div { background-</i>

	image of an element.		<i>image:url(images/img.jpg); }</i> <i>body { background-image:url(img.jpg); }</i>
background-position	Declares the position of a background image.	<i>top left</i> <i>top center</i> <i>top right</i> <i>center left</i> <i>center center</i> <i>center right</i> <i>bottom left</i> <i>bottom center</i> <i>bottom right</i>	<i>div { background-position:10px 50px; }</i> <i>div { background-position:bottom right; }</i>
background-repeat	Declares how and/or if a background image repeats.	<i>repeat</i> <i>repeat-x</i> <i>repeat-y</i> <i>no-repeat</i>	<i>div { background-repeat:repeat-x; }</i> <i>div { background-repeat:no-repeat; }</i>
background	Used as a shorthand property to set all the background properties at once.	<p>Separate values by a space in the following order (those that are not defined will use inherited or default initial values):</p> <p><i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i></p>	<i>div { background:green url(image.jpg) no-repeat fixed center center; }</i> <i>div { background:url(image.jpg) fixed; }</i>

CSS Border

1.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).

```
<!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>
```


The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

CSS Border Width

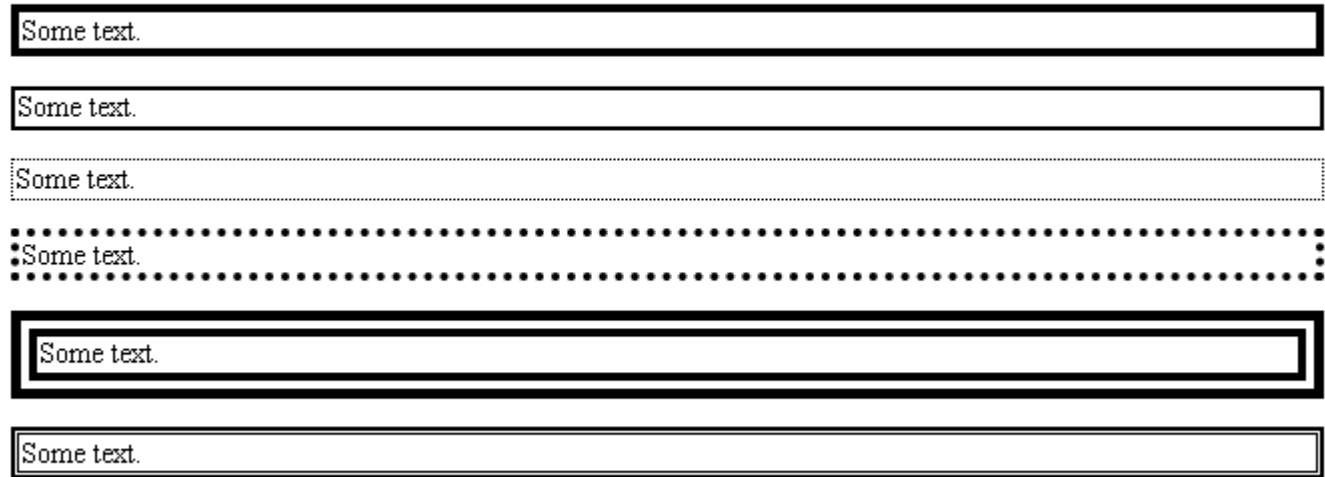
The border-width property specifies the width of the four borders.

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:

```
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;  
}
```

The border-width Property

This property specifies the width of the four borders:



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

```

<!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>

```

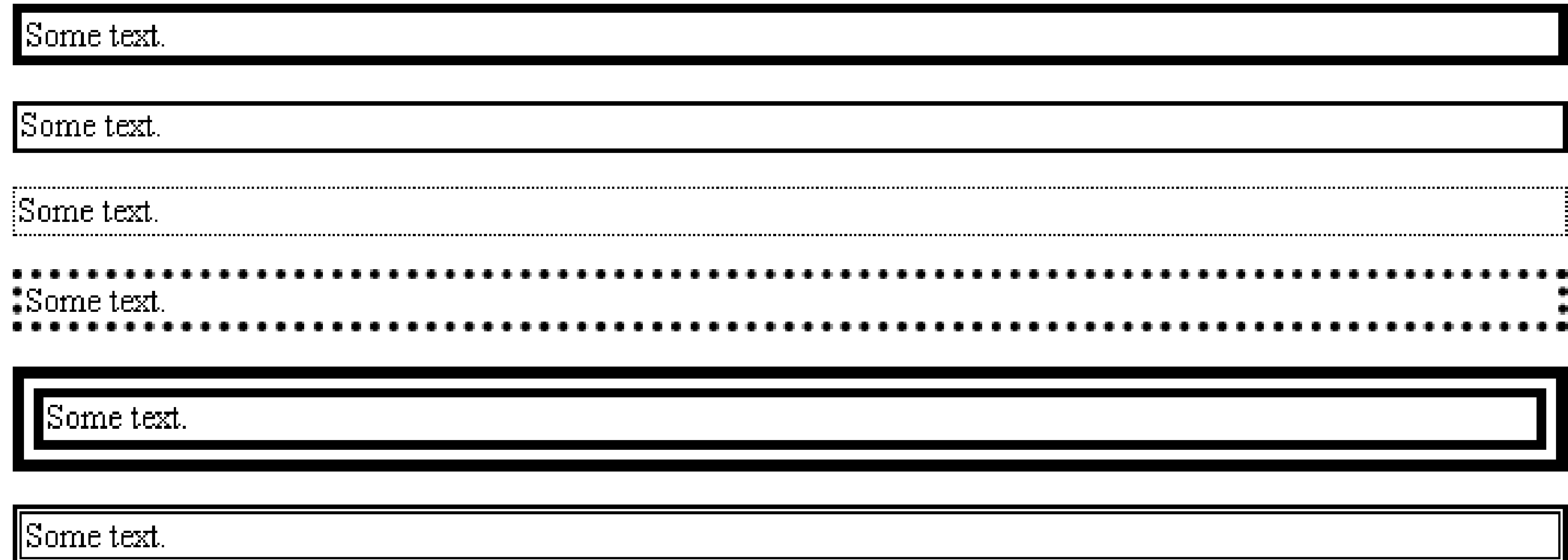
```

</body>
</html>

```

The border-width Property

This property specifies the width of the four borders:



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

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

```

<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>

```

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.

```

<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-color: red green blue yellow;
/* red top, green right, blue bottom and
yellow left */
}
</style>
</head>
<body>

```

```

<h2>The border-color Property</h2>
<p>The border-color property can have from
one to four values (for the top border, right
border, bottom border, and the left
border):</p>

```

```

<p class="one">A solid multicolor
border</p>

```

The border-color Property

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



A solid multicolor border

Specific Side Colors

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

CSS Border Sides

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>
```

Individual Border Sides



This produce same result

```
<style>
body {
  text-align: center;
}
/* Four values */
p.four {
  border-style: dotted solid
double dashed;
}
/* Three values */
p.three {
  border-style: dotted solid
double;
}
```

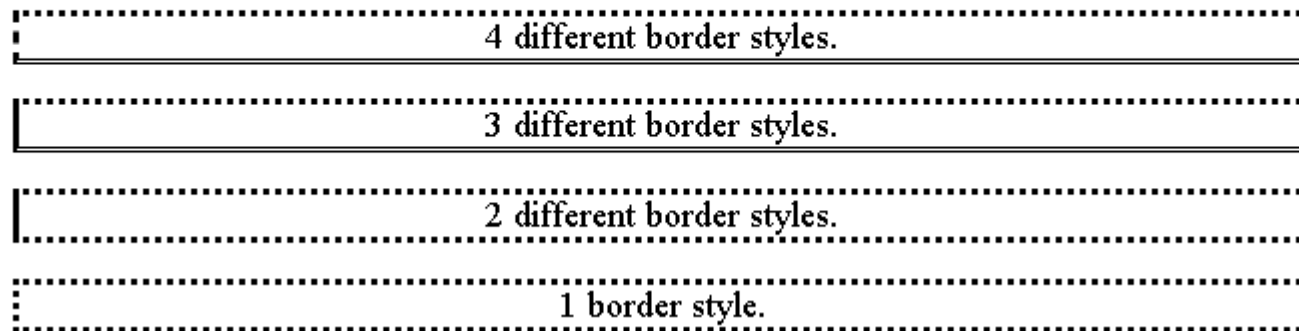
```
/* Two values */
p.two {
  border-style: dotted so
}
```

```
/* One value */
p.one {
  border-style: dotted;
}
```

```
</style>
</head>
```

```
<body>
<h2>Individual Border
Sides</h2>
<p class="four">4 different
border styles.</p>
<p class="three">3 different
border styles.</p>
<p class="two">2 different
border styles.</p>
<p class="one">1 border
style.</p>
```

Individual Border Sides



CSS Border - Shorthand Property

Like you saw in the previous page, there are many properties to consider when dealing with borders.

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

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

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

The border Property

```
<style>  
p {  
  border: 5px solid red;  
}  
</style>  
</head>  
<body>  
<h2>The border Property</h2>  
<p>This property is a shorthand property for border-width,  
border-style, and border-color.</p>  
</body>  
</html>
```

This property is a shorthand property for border-width, border-style, and border-color.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border-left: 6px solid red;
  background-color: lightgrey;
}
</style>
</head>
<body>
```

The border-left Property

This property is a shorthand property for border-left-width, border-left-style, and border-left-color.

```
<h2>The border-left Property</h2>
<p>This property is a shorthand property for border-left-width,
border-left-style, and border-left-color.</p>

</body>
</html>
```

CSS Rounded Borders

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

```
<style>
p.normal {
  border: 2px solid red;
}

p.round1 {
  border: 2px solid red;
  border-radius: 5px;
}

p.round2 {
  border: 2px solid red;
  border-radius: 8px;
}

p.round3 {
  border: 2px solid red;
  border-radius: 12px;
}
</style>
</head>
<body>
<h2>The border-radius
Property</h2>
<p>This property is used to add
rounded borders to an
element:</p>
<p class="normal">Normal
border</p>
<p class="round1">Round
border</p>
<p class="round2">Rounder
border</p>
<p class="round3">Roundest
border</p>
```

The border-radius Property

This property is used to add rounded borders to an element:

Normal border

Round border

Rounder border

Roundest border

Border Properties

Property	Description	Possible Values	Examples
border-top-color	Declares the color of the top border.	Valid color names, RGB values, hexadecimal notation, or the predefined value transparent .	<i>div { border-top-color:green; }</i> <i>div { border-top-color:#00FF00; }</i>
border-top-style	Declares the style of the top border.	<i>none</i> <i>hidden</i> <i>dotted</i> <i>dashed</i> <i>solid</i> <i>double</i> <i>groove</i> <i>ridge</i> <i>inset</i> <i>outset</i>	<i>div { border-top-style:solid; }</i> <i>div { border-top-style:inset; }</i>
border-top-width	Declares the width of the top border.	Lengths or the following predefined values: <i>thin</i> <i>medium</i> <i>thick</i>	<i>div { border-top-width:2px; }</i> <i>div { border-top-width:thin; }</i>
border-top	Used as a shorthand property to set all the border-top properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): <i>border-top-width</i> <i>border-top-style</i> <i>border-top-color</i>	<i>div { border-top:2px solid green; }</i> <i>div { border-top:thick double #00FF00; }</i>

Similarly for right, bottom and left

border-color	Declares the border color of all four borders at once.	<p>Valid color names, RGB values, hexadecimal notation, or the predefined value transparent.</p> <p>Separate the color for each border by a space, declaring the colors for the borders in the following order:</p> <p>border-top-color border-right-color border-bottom-color border-left-color</p> <p>Undeclared values work as further shorthand notation. If only one color value is declared, all four borders will use that color. If two colors are declared, the top and bottom borders will use the first color while the right and left borders will use the second color. If three colors are declared, the top border will use the first color, the right and left borders will use the second color, and the bottom border will use the third color.</p>	<pre>div { border-color:green red blue olive; }</pre> <pre>div { border-color:green; }</pre> <pre>div { border-color:green red; }</pre> <pre>div { border-color:green red blue; }</pre>
border-style	Declares the border style of all four borders at once.	<p><i>none</i> <i>hidden</i> <i>dotted</i> <i>dashed</i> <i>solid</i> <i>double</i> <i>groove</i> <i>ridge</i> <i>inset</i> <i>outset</i></p>	<pre>div { border-style:solid dotted dashed double; }</pre> <pre>div { border-style:solid; }</pre> <pre>div { border-style:solid dotted; }</pre> <pre>div { border-style:solid dotted dashed; }</pre>
border-width	Declares the width of all four borders at once.	<p>Lengths or the following predefined values:</p> <p><i>thin</i> <i>medium</i> <i>thick</i></p>	<pre>div { border-width:1px 3px 5px 2px; }</pre> <pre>div { border-width:thin; }</pre>
border	Used as a shorthand to declare the border properties when all four borders will	Separate values by a space in the following order (those that are not defined will use inherited or default initial values):	<pre>div { border:1px double green; }</pre> <pre>div { border:thin</pre>
	have the same appearance.	<p><i>border-width</i> <i>border-style</i> <i>border-color</i></p>	<pre>solid #00FFFF; }</pre>

CSS Margins

The CSS margin properties are used to create space around elements, outside of any defined borders. With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

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

margin-top

margin-right

margin-bottom

margin-left

All the margin properties can have the following values:

auto - the browser calculates the margin

length - specifies a margin in px, pt, cm, etc.

% - specifies a margin in % of the width of the containing element

inherit - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed.

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 25px 50px 75px 100px;
  background-color: lightblue;
}
</style>
</head>
<body>
```

The margin shorthand property - 4 values

This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.

```
<h2>The margin shorthand property - 4 values</h2>
```

```
<div>This div element has a top margin of 25px, a right margin
of 50px, a bottom margin of 75px, and a left margin of
100px.</div>
```

```
<hr>
```

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



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

Margin - Shorthand Property

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

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

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

So, here is how it works:

If the margin property has four values:

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

If the margin property has three values:

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

If the margin property has two values:

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

If the margin property has one value:

margin: 25px;
all four margins are 25px

The auto Value

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

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

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

The inherit Value

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width:300px;
  margin: auto;
  border: 1px solid red;
}
</style>
</head>
<body>

  <h2>Use of margin:auto</h2>

  <p>You can set the margin property to auto to horizontally
  center the element within its container. The element will then
  take up the specified width, and the remaining space will be
  split equally between the left and right margins:</p>

  <div>
    This div will be horizontally centered because it has margin:
    auto;
  </div>

</body>
</html>
```

Use of margin:auto

You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:

This div will be horizontally centered because it has margin: auto;

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  border: 1px solid
```

```
red;
```

```
  margin-left: 100px;
```

```
}
```

```
p.ex1 {
```

```
  margin-left: inherit;
```

```
}
```

```
</style>
```

```
</head> <body>
```

```
<h2>Use of the inherit value</h2>
```

```
<p>Let the left margin be inherited from the parent  
element:</p>
```

```
<div>
```

```
<p class="ex1">This paragraph has an inherited left margin  
(from the div element).</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

Use of the inherit value

Let the left margin be inherited from the parent element:

This paragraph has an inherited left margin (from the div element).

Summarize

All CSS Margin Properties

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

CSS Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

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

Hello World

RGB Value

In CSS, a color can be specified as an RGB value, using this formula:

`rgb(red, green, blue)`

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.

For example, `rgb(255, 0, 0)` is displayed as red, because red is set to its highest value (255) and the others are set to 0.

HEX Value

A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.

In CSS, a color can be specified using a hexadecimal value in the form:

`#rrggbb`

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

HSL Value

In CSS, a color can be specified using hue, saturation, and lightness (HSL) in the form:

`hsl(hue, saturation, lightness)`

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

RGBA Value

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with:

`rgba(red, green, blue, alpha)`

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

<code>rgb(0, 0, 0)</code>	<code>rgb(60, 60, 60)</code>	<code>#ff0000</code>	<code>#0000ff</code>
<code>rgb(120, 120, 120)</code>	<code>rgb(180, 180, 180)</code>	<code>#3cb371</code>	<code>#ee82ee</code>
<code>rgb(240, 240, 240)</code>	<code>rgb(255, 255, 255)</code>	<code>#ffa500</code>	<code>#6a5acd</code>

<code>rgba(255, 99, 71, 0)</code>	<code>rgba(255, 99, 71, 0.2)</code>	<code>hsl(0, 100%, 50%)</code>	<code>hsl(240, 100%, 50%)</code>
<code>rgba(255, 99, 71, 0.4)</code>	<code>rgba(255, 99, 71, 0.6)</code>	<code>hsl(147, 50%, 47%)</code>	<code>hsl(300, 76%, 72%)</code>
<code>rgba(255, 99, 71, 0.8)</code>	<code>rgba(255, 99, 71, 1)</code>	<code>hsl(39, 100%, 50%)</code>	<code>hsl(248, 53%, 58%)</code>

CSS padding

Padding is used to create space around an element's content, inside of any defined borders.

Padding - Individual Sides

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

padding-top

padding-right

padding-bottom

padding-left

All the padding properties can have the following values:

length - specifies a padding in px, pt, cm, etc.

% - specifies a padding in % of the width of the containing element

inherit - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

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

```
<!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>
```

Using individual padding properties

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.

Padding - Shorthand Property

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

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

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

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

If the padding property has four values:

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

If the padding property has three values:

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

If the padding property has two values:

padding: 25px 50px;
top and bottom paddings are 25px
right and left paddings are 50px

If the padding property has one value:

padding: 25px;
all four paddings are 25px

All CSS Padding Properties

Property	Description
<u>padding</u>	A shorthand property for setting all the padding properties in one declaration
<u>padding-bottom</u>	Sets the bottom padding of an element
<u>padding-left</u>	Sets the left padding of an element
<u>padding-right</u>	Sets the right padding of an element
<u>padding-top</u>	Sets the top padding of an element

CSS Setting height and width

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

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

CSS height and width Values

The height and width properties may have the following values:

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

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

Note: Remember that the height and width properties do not include padding, borders, or margins! They set the height/width of the area inside the padding, border, and margin of the element!

Setting max-width

The max-width property is used to set the maximum width of an element. The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width). The problem with the <div> above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.

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

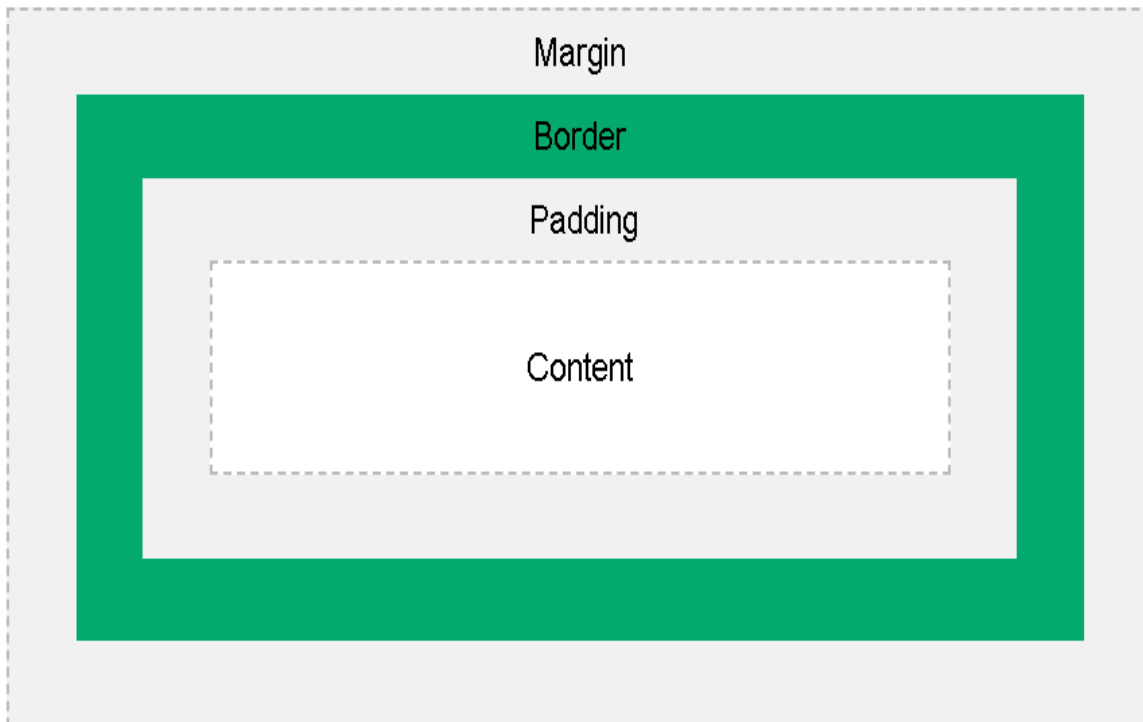
Tip: Drag the browser window to smaller than 500px wide, to see the difference between the two divs!

```
div {  
  max-width: 400px;  
  min-width: 100px;  
  max-height: 600px;  
  min-height: 400px;  
}
```

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

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



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

Explanation of the different parts:

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

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

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

Width and Height of an Element

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

<code>div {</code>	
<code>width: 320px;</code>	320px (width)
<code>padding: 10px;</code>	+ 20px (left + right padding)
<code>border: 5px solid gray;</code>	+ 10px (left + right border)
<code>margin: 0;</code>	+ 0px (left + right margin)
<code>}</code>	= 350px

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS Outline

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

CSS has the following outline properties:

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

Note: Outline differs from borders! Unlike border, the outline is drawn outside the element's border, and may overlap other content. Also, the outline is NOT a part of the element's dimensions; the element's total width and height is not affected by the width of the outline.

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

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

CSS Text

1.Text Color

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

a color name - like "red"

a HEX value - like "#ff0000"

an RGB value - like "rgb(255,0,0)"

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.

Text Color and Background Color

```
body {  
    color: blue;  
}
```

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

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

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

2.CSS Text Alignment

Text Alignment

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

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

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

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

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

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

```
h2 {  
  text-align: left;  
}
```

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

Text Direction

The direction and unicode-bidi properties(used for reverse the letter) can be used to change the text direction of an element:

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

```

<!DOCTYPE html>
<html>
<head>
<style>
img.a {
  vertical-align: baseline;
}
img.b {
  vertical-align: text-top;
}
img.c {
  vertical-align: text-bottom;
}
img.d {
  vertical-align: sub;
}
img.e {
  vertical-align: super;
}
</style>
</head>
<body>
<h1>The vertical-align
Property</h1>
<h2>vertical-align: baseline
(default):</h2>
<p>An  image with a
default alignment.</p>
<h2>vertical-align: text-
top:</h2>
<p>An  image with a text-
top alignment.</p>

```

```

<h2>vertical-align: text-bottom:</h2>
<p>An  image with a text-
bottom alignment.</p>

```

```

<h2>vertical-align: sub:</h2>
<p>An  image with a sub
alignment.</p>

```

```

<h2>vertical-align: sup:</h2>
<p>An  image with a super
alignment.</p>
</body>
</html>

```

The vertical-align Property

vertical-align: baseline (default):



An image with a default alignment.

vertical-align: text-top:



An image with a text-top alignment.

vertical-align: text-bottom:



An image with a text-bottom alignment.

vertical-align: sub:



An image with a sub alignment.

vertical-align: sup:



An image with a super alignment.

3.CSS Text Decoration

Text Decoration

The text-decoration property is used to set or remove decorations from text.

The value text-decoration: none; is often used to remove underlines from links:

This is heading 1

~~This is heading 2~~

This is heading 3

Note: It is not recommended to underline text that is not a link, as this often confuses the reader.

```
<style>
h1 {
  text-decoration: overline;
}

h2 {
  text-decoration: line-through;
}

h3 {
  text-decoration: underline;
}
</style>
</head>
<body>
```

```
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

4.Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

THIS IS SOME TEXT.

this is some text.

This Is Some Text.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.uppercase {
  text-transform: uppercase;
}

p.lowercase {
  text-transform: lowercase;
}
p.capitalize {
  text-transform: capitalize;
}
</style>
</head>
<body>
<p class="uppercase">This is some
text.</p>
<p class="lowercase">This is some
text.</p>
<p class="capitalize">This is some
text.</p>
```

5.CSS Text Spacing

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

Text Indentation

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

Line Height

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

```
h1 {  
  letter-spacing: 3px;  
}  
h2 {  
  letter-spacing: -3px;  
}
```

```
p.small {  
  line-height: 0.8;  
}  
p.big {  
  line-height: 1.8;  
}
```

Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text.

Word Spacing

The word-spacing property is used to specify the space between the words in a text.

```
h1 {  
  word-spacing: 10px;  
}  
h2 {  
  word-spacing: -5px;  
}
```

White Space

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

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


6.Text Shadow

The text-shadow property adds shadow to text.

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

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 1px 1px;
}
</style>
</head>
<body>
<h1>Text-shadow effect!</h1>
</body>
</html>
```

Text-shadow effect!

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

We can add color to it

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

Text-shadow effect!

We can add color to it

```
h1 {
  text-shadow: 2px 2px 5px blue;
}
```

Text-shadow effect!

```
p{
    white-space: nowrap;
width: 200px;
border: 1px solid #000000;
overflow: hidden;
text-overflow:clip;
}
```

This is effect of text overflow y

```
p{
    white-space: nowrap;
width: 200px;
border: 1px solid #000000;
overflow: hidden;
text-overflow:ellipsis;
}
```

This is effect of text overflo...

Overflow : possible values auto, hidden, scroll, visible

visible
This is effect of text overflow yes overflow

Scroll
This is effect of text overfl...

Writing mode:
vertical-rl

Simanta kasaju

Summarize

Property	Description
<u>color</u>	Sets the color of text
<u>direction</u>	Specifies the text direction/writing direction
<u>letter-spacing</u>	Increases or decreases the space between characters in a text
<u>line-height</u>	Sets the line height
<u>text-align</u>	Specifies the horizontal alignment of text
<u>text-decoration</u>	Specifies the decoration added to text
<u>text-indent</u>	Specifies the indentation of the first line in a text-block
<u>text-shadow</u>	Specifies the shadow effect added to text
<u>text-transform</u>	Controls the capitalization of text
<u>text-overflow</u>	Specifies how overflowed content that is not displayed should be signaled to the user
<u>unicode-bidi</u>	Used together with the <u>direction</u> property to set or return whether the text should be overridden to support multiple languages in the same document
<u>vertical-align</u>	Sets the vertical alignment of an element
<u>white-space</u>	Specifies how white-space inside an element is handled
<u>word-spacing</u>	Increases or decreases the space between words in a text

CSS Fonts

Font family

Choosing the right font has a huge impact on how the readers experience a website. The right font can create a strong identity for your brand. Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

Generic Font Families

In CSS there are five generic font families:

Serif fonts have a small stroke at the edges of each letter.

They create a sense of formality and elegance.

Sans-serif fonts have clean lines (no small strokes attached).

They create a modern and minimalistic look.

Monospace fonts - here all the letters have the same fixed width.

They create a mechanical look.

Cursive fonts imitate human handwriting.

Fantasy fonts are decorative/playful fonts.

All the different font names belong to one of the generic font families.

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

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

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

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	<i>Brush Script MT</i> <i>Lucida Handwriting</i>
Fantasy	Copperplate Papyrus

Fallback Fonts

However, there are no 100% completely web safe fonts. There is always a chance that a font is not found or is not installed properly. Therefore, it is very important to always use fallback fonts.

This means that you should add a list of similar "backup fonts" in the font-family property. If the first font does not work, the browser will try the next one, and the next one, and so on. Always end the list with a generic font family name.

Best Web Safe Fonts for HTML and CSS

The following list are the best web safe fonts for HTML and CSS:

Arial (sans-serif)

Verdana (sans-serif)

Helvetica (sans-serif)

Tahoma (sans-serif)

Trebuchet MS (sans-serif)

Times New Roman (serif)

Georgia (serif)

Garamond (serif)

Courier New (monospace)

Brush Script MT (cursive)

```
p {  
font-family: Tahoma, Verdana, sans-serif;  
}
```


Font Style

The font-style property is mostly used to specify italic text.

This property has three values:

normal - The text is shown normally

italic - The text is shown in italics

oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.normal {  
  font-weight: normal;  
}  
p.light {  
  font-weight: lighter;  
}  
p.thick {  
  font-weight: bold;  
}  
p.thicker {  
  font-weight: 900;  
}
```

Font Variant

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

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

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

Font Weight

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

Font Size

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

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

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs. The font-size value can be an absolute, or relative size.

Absolute size:

Sets the text to a specified size

Does not allow a user to change the text size in all browsers (bad for accessibility reasons)

Absolute size is useful when the physical size of the output is known

Relative size:

Sets the size relative to surrounding elements

Allows a user to change the text size in browsers

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

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

Set Font Size With Pixels

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

Set Font Size With Em

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

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.

The size can be calculated from pixels to em using this formula: $\text{pixels}/16=\text{em}$

Trirong Font

```
<link rel="stylesheet" href="https://fonts.
googleapis.com/css?family=Trirong">
```

Audiowide Font

```
<link rel="stylesheet" href="https://font
s.googleapis.com/css?family=Audiowide">
```

Responsive Font Size

The text size can be set with a vw unit, which means the "viewport width".

That way the text size will follow the size of the browser window:

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

Multiple fonts

```
<link rel="stylesheet" href="https://font
s.googleapis.com/css?family=Audiowide|Sof
ia|Trirong">
```

CSS Links

The four links states are:

a:link - a normal, unvisited link

a:visited - a link the user has visited

a:hover - a link when the user mouses over it

a:active - a link the moment it is clicked

```
color: blue;  
text-decoration: underline;  
text-decoration: none;  
background-color: lightgreen;
```

Link Buttons

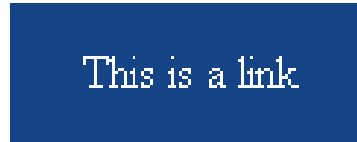
This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes/buttons:

```
<style>
a:link, a:visited {
  background-color: #144386;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}
</style>
```

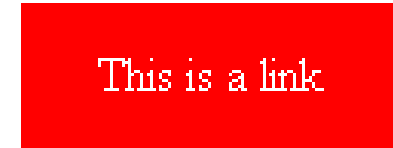
Link Button

A link styled as a button:



Link Button

A link styled as a button:



```
<!DOCTYPE html>
<html>
<body>
<p>Mouse over the words to change the cursor.</p>
<span style="cursor:auto">auto</span><br>
<span style="cursor:crosshair">crosshair</span><br>
<span style="cursor:default">default</span><br>
<span style="cursor:e-resize">e-resize</span><br>
<span style="cursor:help">help</span><br>
<span style="cursor:move">move</span><br>
<span style="cursor:n-resize">n-resize</span><br>
<span style="cursor:ne-resize">ne-resize</span><br>
<span style="cursor:nw-resize">nw-resize</span><br>
<span style="cursor:pointer">pointer</span><br>
<span style="cursor:progress">progress</span><br>
<span style="cursor:s-resize">s-resize</span><br>
<span style="cursor:se-resize">se-resize</span><br>
<span style="cursor:sw-resize">sw-resize</span><br>
<span style="cursor:text">text</span><br>
<span style="cursor:w-resize">w-resize</span><br>
<span style="cursor:wait">wait</span><br>
</body>
</html>
```

CSS Lists

In HTML, there are two main types of lists:

unordered lists () - the list items are marked with bullets

ordered lists () - the list items are marked with numbers or letters

```
list-style-type: circle;  
list-style-type: square;  
list-style-type: upper-roman;  
list-style-type: lower-alpha;  
list-style-type: none;  
list-style-type: disc;  
list-style-type: decimal;
```

Unordered Lists:

- Coffee
 - Tea
 - Coca Cola
-
- Coffee
 - Tea
 - Coca Cola

Ordered Lists:

1. Coffee
 2. Tea
 3. Coca Cola
-
- I. Coffee
 - II. Tea
 - III. Coca Cola

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

Position The List Item Markers

The list-style-position property specifies the position of the list-item markers (bullet points).

"**list-style-position: outside;**" means that the bullet points will be outside the list item. The start of each line of a list item will be aligned vertically. This is default:

- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola

- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola

"**list-style-position: inside;**" means that the bullet points will be inside the list item. As it is part of the list item, it will be part of the text and push the text at the start:

List - Shorthand property

The list-style property is a shorthand property. It is used to set all the list properties in one declaration:

```
list-style: square inside url("sqpurple.gif");
```

When using the shorthand property, the order of the property values are:

list-style-type (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)

list-style-position (specifies whether the list-item markers should appear inside or outside the content flow)

list-style-image (specifies an image as the list item marker)

Styling List With Colors

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

Anything added to the or tag, affects the entire list, while properties added to the tag will affect the individual list items:

```
<style>
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;
}
</style>
</head>
<body>

<h1>Styling Lists With Colors:</h1>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
```

Styling Lists With Colors:

- 
1. Coffee
 2. Tea
 3. Coca Cola

- 
- Coffee
 - Tea
 - Coca Cola

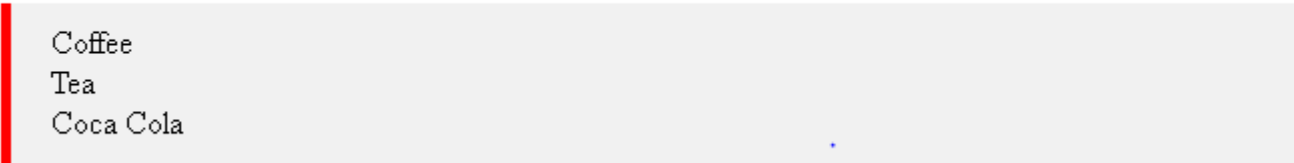
```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

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

```
<style>
ul {
  border-left: 5px solid red;
  background-color: #f1f1f1;
  list-style-type: none;
  padding: 10px 20px;
}
</style>
</head>
<body>
```

```
<p>List with a red left border:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

List with a red left border:



- Coffee
- Tea
- Coca Cola

summary

List Properties

Property	Description	Possible Values	Examples
list-style-type	Declares the type of list marker used.	<i>disc</i> <i>circle</i> <i>square</i> <i>decimal</i> <i>decimal-leading-zero</i> <i>lower-roman</i> <i>upper-roman</i> <i>lower-alpha</i> <i>upper-alpha</i> <i>lower-greek</i> <i>lower-latin</i> <i>upper-latin</i>	<i>ol { list-style-type:upper-roman; }</i> <i>ul { list-style-type:square; }</i>
list-style-position	Declares the position of the list marker.	<i>inside</i> <i>outside</i>	<i>ol { list-style-position:inside; }</i> <i>ul { list-style-position:outside; }</i>
list-style-image	Declares an image to be used as the list marker.	URL values.	<i>ul { list-style-image:url(image.jpg); }</i>
list-style	Shorthand property to declare three list properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): <i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>	<i>ul { list-style:disc inside url(image.gif); }</i> <i>ol { list-style:upper-roman outside; }</i>
marker-offset	Declares the marker offset for elements with a value of marker set for the display property.	Lengths and the predefined value auto .	<i>li:before { display:marker; marker-offset:5px; }</i>

CSS Tables

Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for <table>, <th>, and <td> elements:

```
table, th, td {  
  border: 1px solid black;  
}
```

Add a border to a table:

Firstname	Lastname
Simanta	Kasaju
Puran	Kasaju

Full-Width Table

The table above might seem small in some cases. If you need a table that should span the entire screen (full-width), add width: 100% to the <table> element:

```
table {  
  width: 100%;  
}
```

```
table, th, td {  
  border: 1px solid red;  
}
```

Add a border to a table:

Firstname	Lastname
Simanta	Kasaju
Puran	Kasaju

Collapse Table Borders

The border-collapse property sets whether the table borders should be collapsed into a single border:

```
table {  
  border-collapse: collapse;  
}
```

CSS Table Size

Table Width and Height

The width and height of a table are defined by the width and height properties.

CSS Table Alignment

Horizontal Alignment

The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.

By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned to center-align the content of <td> elements as well, use text-align: center:

```
text-align: center;
```

Vertical Alignment

The vertical-align property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.

By default, the vertical alignment of the content in a table is middle (for both <th> and <td> elements).

```
vertical-align: bottom;
```

CSS Table Style

Table Padding

To control the space between the border and the content in a table, use the padding property on <td> and <th> elements:

Horizontal Dividers

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  padding: 8px;
  text-align: left;
  border-bottom: 1px solid #ddd;
}
</style>
</head>
<body>
```

```
<h2>Bordered Table Dividers</h2>
<p>Add the border-bottom property
to th and td for horizontal
dividers:</p>
```

```
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Gender</th>
  </tr>
  <tr>
    <td>Puran</td>
    <td>Kasaju</td>
    <td>Male</td>
  </tr>
```

```
<tr>
  <td>Shashi</td>
  <td>Kasaju</td>
  <td>Male</td>
</tr><tr>
  <td>Salina</td>
  <td>Kasaju</td>
  <td>Female</td>
</tr>
<tr>
  <td>Prashant</td>
  <td>Kasaju</td>
  <td>Male</td>
</tr>
<tr>
  <td>Simanta</td>
  <td>Kasaju</td>
  <td>Male</td>
</tr>
</table>
</body>
</html>
```

Bordered Table Dividers

Add the border-bottom property to th and td for horizontal dividers:

Firstname	Lastname	Gender
Puran	Kasaju	Male
Shashi	Kasaju	Female
Salina	Kasaju	Female
Prashant	Kasaju	Male
Simanta	Kasaju	Male

Hoverable Table

Use the :hover selector on <tr> to highlight table rows on mouse over:

Just add this to previous example

```
tr:hover {background-color:green;}
```

Hoverable Table

Move the mouse over the table rows to see the effect.

Firstname	Lastname	Gender
Puran	Kasaju	Male
Shashi	Kasaju	Female
Salina	Kasaju	Female
Prashant	Kasaju	Male
Simanta	Kasaju	Male


```
th:hover{color:yellow;}
tr.a:hover {background-color:seagreen}
tr.b:hover {background-color:rgb(151, 201, 172)}
tr.c:hover {background-color:rgb(188, 226, 48)}
tr.d:hover {background-color:rgb(221, 9, 150)}
tr.e:hover {background-color:rgb(9, 187, 231)}
tr.f:hover {background-color:rgb(9, 187, 231)}
```

Hoverable Table

Move the mouse over the table rows to see the effect.

Firstname	Lastname	Gender
Puran	Kasaju	Male
Shashi	Kasaju	Female
Salina	Kasaju	Female
Prashant	Kasaju	Male
Simanta	Kasaju	Male

Striped Tables

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

```
tr:nth-child(even){background-color: #111111;
color:white}
```

```
th {
background-color: #04AA6D;
color: white;
}
```

Striped Table

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

Firstname	Lastname	Gender
Puran	Kasaju	Male
Shashi	Kasaju	Female
Salina	Kasaju	Female
Prashant	Kasaju	Male
Simanta	Kasaju	Male

Striped Table

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

Firstname	Lastname	Gender
Puran	Kasaju	Male
Shashi	Kasaju	Female
Salina	Kasaju	Female
Prashant	Kasaju	Male
Simanta	Kasaju	Male

style="overflow-x: auto;"

Table Properties

Property	Description	Possible Values	Examples
border-collapse	Declares the way borders are displayed.	<i>collapse</i> <i>separate</i>	<i>table { border-collapse:collapse; }</i> <i>table { border-collapse:separate; }</i>
border-spacing	Declares the distance separating borders (if border-collapse is separate).	Lengths for the horizontal and vertical spacing, separated by a space. If one length is value is declared, that length is used for both the horizontal and vertical spacing. If two lengths are declared, the first one is used for horizontal spacing and the second one is used for vertical spacing.	<i>table { border-spacing:5px; }</i> <i>table { border-spacing:5px 10px; }</i>
caption-side	Declares where the table caption is displayed in relation to the table.	<i>top</i> <i>bottom</i> <i>left</i> <i>right</i>	<i>caption { caption-side:top; }</i> <i>caption { caption-side:right; }</i>
empty-cells	Declares the way empty cells are displayed (if border-collapse is separate).	<i>show</i> <i>hide</i>	<i>table { empty-cells:show; }</i> <i>table { empty-cells:hide; }</i>
table-layout	Declares the type of table layout.	<i>auto</i> <i>fixed</i>	<i>table { table-layout:auto; }</i> <i>table { table-layout:fixed; }</i>

CSS Layout - The display Property

The display property is the most important CSS property for controlling layout.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

Examples of block-level elements:

```
<div>
<h1> - <h6>
<p>
<form>
<header>
<footer>
<section>
```

Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline `` element inside a paragraph.

Examples of inline elements:

```
<span>
<a>
<img>
```

```
visibility:hidden;
display: none;
display: block;
display: inline;
```

CSS Layout - width and max-width

Using max-width instead, in this situation, will improve the browser's handling of small windows. This is important when making a site usable on small devices:

```
div.ex1 {  
  width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

```
div.ex2 {  
  max-width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

Tip: Resize the browser window to less than 500px wide, to see the difference between the two divs!

CSS Layout - The position Property

The position Property

The position property specifies the type of positioning method used for an element.

There are five different position values:

static

relative

fixed

absolute

sticky

Elements are then positioned using the top, bottom, left, and right properties.

However, these properties will not work unless the position property is set first.

They also work differently depending on the position value.

```
div.sticky{  
    position: sticky;  
    bottom: 0;  
    right: 0;  
    top:0;  
    background-color:red;  
  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```

```
div.fixed {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```


CSS Layout - float and clear

The CSS float property specifies how an element should float.

The CSS clear property specifies what elements can float beside the cleared element and on which side.

The float Property

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The float property can have one of the following values:

left - The element floats to the left of its container

right - The element floats to the right of its container

none - The element does not float (will be displayed just where it occurs in the text). This is default

inherit - The element inherits the float value of its parent

In its simplest use, the float property can be used to wrap text around images.

```
img {  
    float: right;  
    float: left;  
    float: none;  
}
```

When we use the float property, and we want the next element below (not on right or left), we will have to use the clear property.

The clear property specifies what should happen with the element that is next to a floating element.

The clear property can have one of the following values:

none - The element is not pushed below left or right floated elements. This is default

left - The element is pushed below left floated elements

right - The element is pushed below right floated elements

both - The element is pushed below both left and right floated elements

inherit - The element inherits the clear value from its parent

When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left. Your floated element will continue to float, but the cleared element will appear below it on the web page.

```
<!DOCTYPE html>
<html>
<head>
<style>
.div1 {
  float: left;
  padding: 10px;
  border: 3px solid #73AD21;
}

.div2 {
  padding: 10px;
  border: 3px solid red;
}

.div3 {
  float: left;
  padding: 10px;
  border: 3px solid #73AD21;
}
```

```
.div4 {
  padding: 10px;
  border: 3px solid red;
  clear: left;
}
</style>
</head>
<body>
<h2>Without clear</h2>
<div class="div1">div1</div>
<div class="div2">div2 - Notice that div2 is after div1 in the
HTML code. However, since div1 floats to the left, the text in
div2 flows around div1.</div>
<br><br>

<h2>With clear</h2>
<div class="div3">div3</div>
<div class="div4">div4 - Here, clear: left; moves div4 down
below the floating div3. The value "left" clears elements floated
to the left. You can also clear "right" and "both".</div>
</body>
</html>
```

Without clear

div1 div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

With clear

div3

div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

The clearfix Hack

If a floated element is taller than the containing element, it will "overflow" outside of its container.

We can then add a clearfix hack to solve this problem:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 3px solid
#4CAF50;
  padding: 5px;
}
.img1 {
  float: right;
}
.img2 {
  float: right;
}
.clearfix {
  overflow: auto;
}
```

```
</style>
</head>
<body>
```

```
<h2>Without Clearfix</h2>
```

```
<p>This image is floated to the
right. It is also taller than the
element containing it, so it
overflows outside of its
container:</p>
```

```
<div>
  
  Sleeping baby
</div>
```

```
<h2 style="clear:right">With
Clearfix</h2>
```

```
<p>We can fix this by adding a
clearfix class with overflow:
auto; to the containing
element:</p>
```

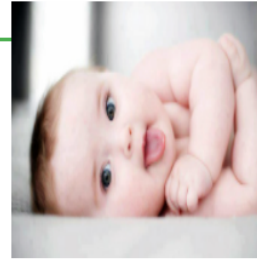
```
< div class="clearfix">
  
  Sleeping baby
</div>
```

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

Without Clearfix

This image is floated to the right. It is also taller than the element containing it, so it overflows outside of its container:

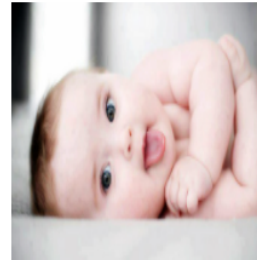
Sleeping baby



With Clearfix

We can fix this by adding a clearfix class with `overflow: auto;` to the containing element:

Sleeping baby



We can do this by using after

```
.clearfix::after {  
  content: "";  
  clear: both;  
  display: table;  
}
```

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #111;
}

```

```

li a:hover {
  background-color: blue;
}

```



```

.active {
  background-color: red;
}

```

```

</style>

```

```

</head>

```

```

<body>

```



```

li {
  float: left;
}

li a {
  display: inline-block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

```

```

<ul>

```

```

  <li><a href="#home" class="active">Home</a></li>

```

```

  <li><a href="#Department">News</a></li>

```

```

  <li><a href="#Gallery">Gallery</a></li>

```

```

  <li><a href="#Academics">Academic</a></li>

```

```

  <li><a href="#Download">Download</a></li>

```

```

  <li><a href="#contact">Contact</a></li>

```

```

</ul>

```

```

</body>

```

```

</html>

```

All CSS Float Properties

Property	Description
<u>box-sizing</u>	Defines how the width and height of an element are calculated: should they include padding and borders, or not
<u>clear</u>	Specifies what should happen with the element that is next to a floating element
<u>float</u>	Specifies whether an element should float to the left, right, or not at all
<u>overflow</u>	Specifies what happens if content overflows an element's box
<u>overflow-x</u>	Specifies what to do with the left/right edges of the content if it overflows the element's content area
<u>overflow-y</u>	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area

CSS Navigation Bar

Navigation Bars

Having easy-to-use navigation is important for any web site.

With CSS you can transform boring HTML menus into good-looking navigation bars.

Navigation Bar = List of Links

A navigation bar needs standard HTML as a base.

In our examples we will build the navigation bar from a standard HTML list.

A navigation bar is basically a list of links, so using the `` and `` elements makes perfect sense:

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: #555;
  color: white;
}
</style>
</head>
<body>

<h2>Vertical Navigation Bar</h2>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>

```

Vertical Navigation Bar

Home

News

Contact

About

Vertical Navigation Bar

Home

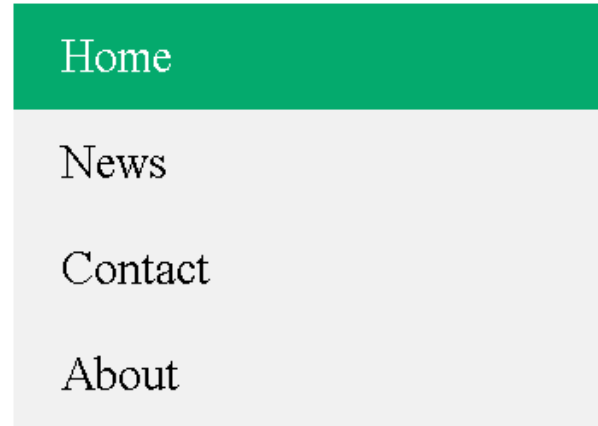
News

Contact

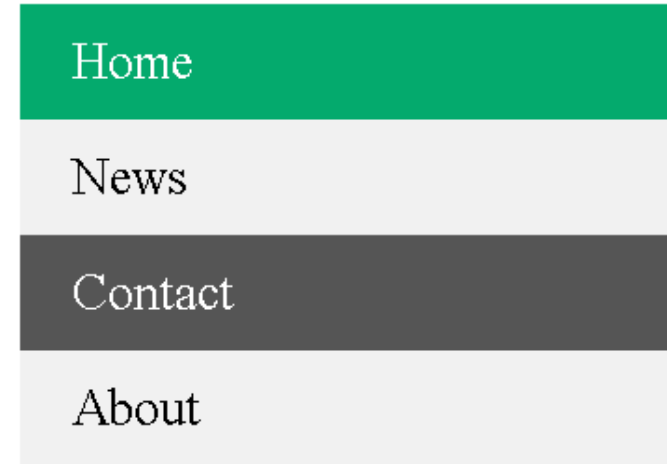
About

```
li a.active {  
  background-color: #04AA6D;  
  color: white;  
}
```

Vertical Navigation Bar



Vertical Navigation Bar



Add this on ul and li
`border-bottom: 1px solid #555;`

Vertical Navigation Bar



<pre> <!DOCTYPE html> <html> <head> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <style> body {margin: 0;} ul.topnav { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; } ul.topnav li {float: left;} </pre>	<pre> ul.topnav li a { display: block; color: white; text-align: center; padding: 14px 16px; text-decoration: none; } ul.topnav li a.active {background-color: #04AA6D;} ul.topnav li.right {float: right;} @media screen and (max-width: 600px) { ul.topnav li.right, ul.topnav li {float: none;} } </style> </head> <body> <ul class="topnav"> Home News Contact <li class="right">About </pre>	<pre> <div style="padding:0 16px;"> <h2>Responsive Topnav Example</h2> <p>This example use media queries to stack the topnav vertically when the screen size is 600px or less.</p> <p>You will learn more about media queries and responsive web design later in our CSS Tutorial.</p> <h4>Resize the browser window to see the effect.</h4> </div> </body> </html> </pre>
---	--	---

<!DOCTYPE html>	</style>
<html>	</head>
<head>	<body>
<style>	
.dropdown {	<h2>Dropdown Image</h2>
display: inline-block;	<p>Move the mouse over the image below to open the
}	dropdown content.</p>
.dropdown-content {	<div class="dropdown">
display: none;	
	<div class="dropdown-content">
background-color: #f9f9f9;	
min-width: 160px;	<div class="desc">Cute baby</div>
	</div>
}	</div>
.dropdown:hover .dropdown-	
content {	</body>
display: block;	</html>
}	
.desc {	
padding: 15px;	
text-align: center;	
}	

<pre> <!DOCTYPE html> <html> <head> <style> .dropbtn { background-color: #4CAF50; color: white; padding: 16px; font-size: 16px; border: none; cursor: pointer; } .dropdown { position: relative; display: inline-block; } .dropdown-content { display: none; background-color: #f9f9f9; min-width: 160px; </pre>	<pre> } .dropdown-content a { color: black; padding: 12px 16px; text-decoration: none; display: block; } .dropdown-content a:hover {background-color: #f1f1f1} .dropdown:hover .dropdown-content { display: block; } .dropdown:hover .dropbtn { background-color: #3e8e41; } </style> </pre>	<pre> </head> <body> <h2>Dropdown Menu</h2> <p>Move the mouse over the button to open the dropdown menu.</p> <div class="dropdown"> <button class="dropbtn">Dropdown</button> <div class="dropdown-content"> Link 1 Link 2 Link 3 </div> </div> </body> </html> </pre>
---	--	---

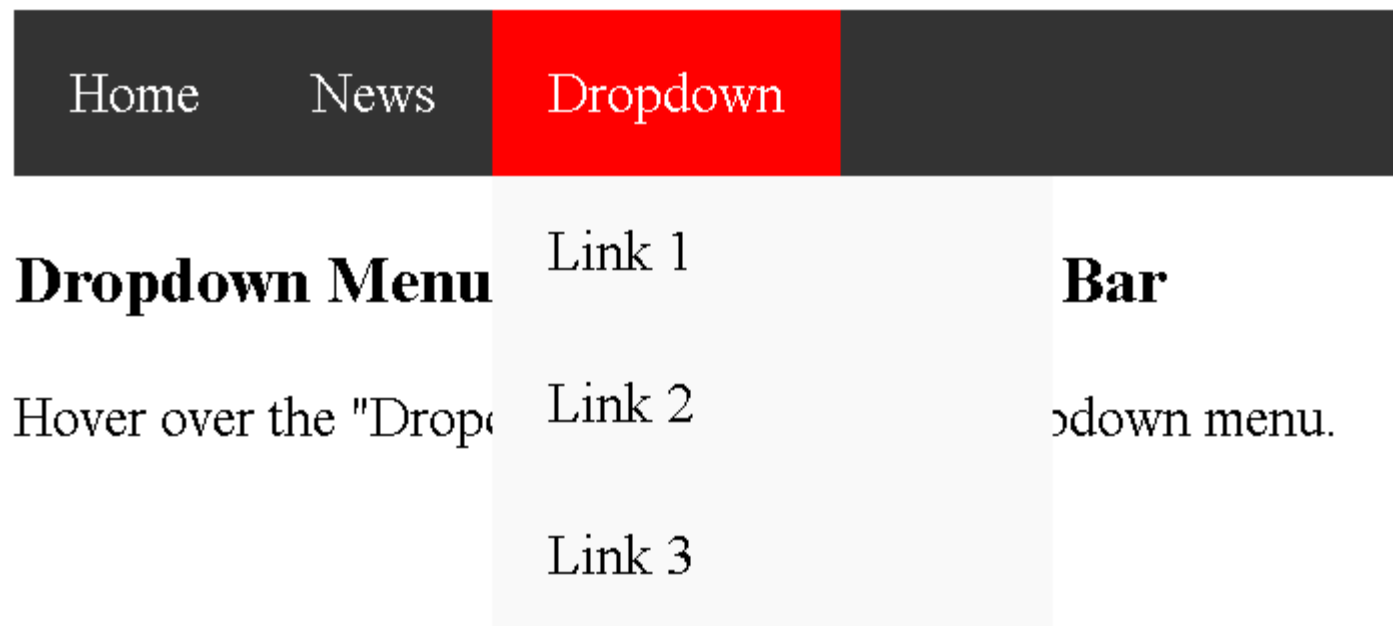


Image	div.desc { padding: 15px; text-align: center; }	 <div class="desc">Add a description of the image here</div> </div> <div class="gallery"> <div class="desc">Add a description of the image here</div> </div>
<!DOCTYPE html> <html> <head> <style> div.gallery { margin: 5px; border: 1px solid #ccc; float: left; width: 180px; }	</style> </head> <body> <div class="gallery"> width="600" height="400"> <div class="desc">Add a description of the image here</div> </div> <div class="gallery"> width="600" height="400">	
div.gallery: hover { border: 1px solid #777; }		
div.gallery img { width: 100%; height: auto; }		</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

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  opacity: 0.2;
}
img:hover {
  opacity: 1.0;
}
</style>
</head>
<body>
<h1>Image Transparency</h1>
<p>The opacity property is often used together with the :hover
selector to change the opacity on mouse-over:</p>



</body>
</html>
```

Image Transparency

The opacity property is often used together with the :hover selector to change the opacity on mouse-over:



```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  border-radius: 20px;
}
</style>
</head>
<body>
```

```
<h2>Rounded Images</h2>
```

```
<p>Use the border-radius property to create rounded
images:</p>
```

```

```

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

Rounded Images

Use the border-radius property to create rounded images:



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img {  
  width: 33%;  
  height: auto;  
  float: left;  
  max-width: 235px;  
}
```

```
.blur {filter: blur(2px);}
```

```
.brightness {filter: brightness(150%);}
```

```
.contrast {filter: contrast(180%);}
```

```
.grayscale {filter: grayscale(500%);}
```

```
.huerotate {filter: hue-  
rotate(180deg);}
```

```
.invert {filter: invert(100%);}
```

```
.opacity {filter: opacity(50%);}
```

```
.saturate {filter: saturate(7);}
```

```
.sepia {filter: sepia(100%);}
```

```
.shadow {filter: drop-shadow(8px  
8px 10px green);}
```

```
</style>
```

```
</head>
```

```
<body>
```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```
</body>
```

```
</html>
```




CSS Units

Absolute Lengths

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Relative Lengths

Relative length units specify a length relative to another length property. Relative length units scales better between different rendering mediums.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

CSS Forms

```

<!DOCTYPE html>
<html>
<style>
input[type=text], select {
  width: 50%;
  padding: 12px 20px;
  margin: 8px 5px;
  display: block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
input[type=submit] {
  width: 20%;
  display: block;
  background-color: black;
  color: white;
  padding: 5px 5px;
  margin: 8px 50px;
  border: none;
  border-radius: 20px;
  cursor: pointer;
  text-align: center;
}
input[type=submit]:hover {
  background-color: red;}
div {
  border-radius: 5px;
  background-color: rgba(8, 214, 214, 0.133);
  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="Nepal">Nepal</option>
  <option
value="America">America</option>
  <option
value="australia">Australia</option>
  <option value="canada">Canada</option>
  <option value="usa">USA</option>
</select>
<label for ="gender" >Gender:</label>
<input type="radio" name="gender"
value="Male">Male
  <input type="radio" name="gender"
value="Female">Female
  <br>
  <br>
  <label for ="Hobby" >Hobbies:</label>
  <input type="checkbox" value="singing">
singing
  <input type="checkbox"
value="Dancing">Dancing
  <input type="checkbox"
value="Vlogging">Vlogging

```

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

</body>
</html>
```

Using CSS to style an HTML Form

First Name

Last Name

Country

Gender: ☐ Male ☐ Female

Hobbies: ☐ singing ☐ Dancing ☐ Vlogging

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 20%;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  font-size: 16px;
  background-color: white;
  background-image: url('capture.png');
  background-size: 30px;
  background-position: 3px 1px;
  background-repeat: no-repeat;
  padding: 12px 20px 12px 40px;
}
</style>
</head>
<body>

<p>Input with icon:</p>

<form>
  <input type="text" name="search" placeholder="Search..">
</form>

</body>
</html>
```

Input with icon:



Color text field

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: none;
  background-color: #a8c6c9;
  color: black;
}
</style>
</head>
<body>

<p>Colored text fields:</p>
```

```
<form>
  <label for="fname">First Name</label>
  <input type="text" id="fname" name="fname"
value="Simanta">
  <label for="lname">Last Name</label>
  <input type="text" id="lname" name="lname"
value="Kasaju">
</form>

</body>
</html>
```

Colored text fields:

First Name

Simanta

Last Name

Kasaju

Focus

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: 1px solid #555;
  outline: none;
}

input[type=text]:focus {
  background-color: lightblue;
}
</style>
</head>
<body>
```

<p>In this example, we use the :focus selector to add a background color to the text field when it gets focused (clicked on):</p>

```
<form>
  <label for="fname">First Name</label>
  <input type="text" id="fname" name="fname"
value="Simanta">
  <label for="lname">Last Name</label>
  <input type="text" id="lname" name="lname"
value="Kasaju">
</form>

</body>
</html>
```

First Name

Simanta|

Last Name

Kasaju

CSS3 Introduced Media Queries

Media queries in CSS3 extended the CSS2 media types idea: Instead of looking for a type of device, they look at the capability of the device.

Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as iPhone and Android phones).

Media Query Syntax

A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

CSS3 Media Types

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud


```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  color: black;
  background-image: url(pic/4.jpg);
  background-size: cover;
  background-repeat: no-repeat;
}

@media screen and (min-width: 500px) {
  body {

    color: red;
    background-image: url(pic/3.jpg);
    background-repeat: no-repeat;
    background-size: cover;
  }
}

```

```

</style>
</head>
<body>
<h1>Resize the browser window to see the effect!</h1>
</body>
</html>

```



When above 500



upto 500

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1">
<style>
.topnav {
  overflow: hidden;
  background-color:#112233;
}
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

```

```

.active{
  background-color: red;
}
@media screen and (max-width: 600px) {
  .topnav a {
    float: none;
    width: 100%;
  }
}
</div>
</style>
</body>
</head>
</html>
<body>
<h2>Responsive navigation menu</h2>
<p>Resize the browser window to see the
effect: When the screen is less than 600px, the
navigation menu will be displayed vertically
instead of horizontally.</p>
<div class="topnav">
  <a href="#" class="active">Home</a>
  <a href="#">Gallery</a>
  <a href="#">Contact</a>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    a{
      text-decoration: none;
      color: black;
      padding: 2px;
    }
    a:hover{
      background-color: #00f000;
    }
    .active{
      background-color: red;
    }
  </style>

```

```

</head>
<body>
  <div>
    <a href="">&laquo;</a>
    <a href="1">1</a>
    <a href="2">2</a>
    <a href="3" class="active">3</a>
    <a href="4">4</a>
    <a href="5">5</a>
    <a href="">&laquo;</a>
  </div>
</body>
</html>

```



```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport"
content="width=device-width, initial-
scale=1.0">
<style>
    body{
        background-image: url(map.png);
        background-position: center;
        background-size: cover;
    }
    * {
        box-sizing: border-box;
    }

    .row::after {
        content: "";
        clear: both;
        display: block;
    }

```

```

[class*="col-"] {
    float: left;
    padding: 15px;
}

.header {
    background-color: #ff0000;
    color: #ffffff;
    padding: 15px;
    text-align: center;
}

.menu ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

```

```

.menu li {
    padding: 8px;
    border-radius: 20px;
    margin-bottom: 5px;
    background-color:
#093509;
    color: #03eeee;
    text-align: center;
}

.menu li:hover {
    background-color:
#ff0156;
}

```

```

.aside {
  background-color: #00ffea;
  padding: 10px;
  color: #112233;
  text-align: center;
  font-size: 20px;
}

.col-3 {width: 25%;}
.col-6 {width: 50%;}

.footer {
  color: #ffffff;
  text-align: center;
  font-size: 12px;
  padding: 15px;
}

p.copyright {
  color: #fff;
  line-height: 40px;
  font-size: 0.7em;
  background-color: red;
}

@media only screen and (max-width: 768px) {
  /* For mobile phones: */
  [class*="col-"] {
    width: 100%;
  }
}
</style>
</head>
<body>

<div class="header">
  <h1>Nepal</h1>

```

```

<marquee>हिमालको सेतो फेटा, पहाडको
हरियो भोटा र तराईको पहुँलो कछाड फेरेर
बसेको मेरो देश मेरो मुटुको टुक्रा हो ।
सिन्धुमा हराएको विन्दु जस्तै विश्वको
भूमण्डलमा सानो देखिए पनि मेरो देश मेरो
प्राकृतिक दृश्यले एकदमै सुन्दर र पवित्र छ ।
पूर्व मेची र पश्चिममा कालीको रेखा कोरेर
उत्तरतर्फ हिमश्रृङ्खलाको प्राकृतिक
पर्खाल लगाई स्वयम् प्रकृति नै मेरो
देशको रक्षामा खटिएकी छिन् । यस देशको
चार किल्लाभित्र प्रकृति रमाउँदै नाचेकी छिन्
</marquee>

</div>

```

```
<div class="row">
  <div class="col-3 menu">
    <ul>
      <li>Places</li>
      <li>Food</li>
      <li>Temples</li>
      <li>Culture</li>
      <li>Festivals</li>
    </ul>
  </div>
  <div class="col-6">
    <h1>Nepal</h1>
    <marquee> मेरो देश नेपाल</marquee>
    <p>
```

 <div style="background-color: sandybrown; padding: 5px;"> is a
small, landlocked country, situated between Tibet (China) to
 its north, and India to its east, west and south. At latitudes between
26 and 30

 degrees north and longitudes between 80 and 88 degrees east, the
country covers an

 area of 147,181sq.km and stretches approximately 145-241 km
 </div>

```
</p>
```

```
</div>
```

```

<div class="col-3 right">
  <div class="aside">
    <h2>What is for?</h2>
    <p>Nepal is known as the famous ultimate outdoor-sports
destination, offering a wide range of sports like rock climbing,
canyoning, ziplining etc.</p>
    <h2>special about nepal?</h2>
    <p>Nepal is the home to eight of the highest peak of the
world among the top ten highest peaks. The country contains
beautiful and amazing mountain ranges which are the reason it
is termed as a Himalayan country. </p>
    <h2>Reasons to Visit Nepal?</h2>
    <p>

```

```

<ul>
  <li>Trek to Everest Base Camp.</li>
  <li>Learn the culture of Kathmandu
</li>
  <li>Admire the Boudhanath Stupa</li>
</ul></p>
</div>
</div>
</div>
<div class="footer">
  <p class="copyright">©2021 ,Simanta Kasaju, All
Rights Reserved.</p>
</div>
</body>
</html>

```

Nepal

हिमालको सेतो फेरा, पहाडको हरियो भीटा र तराईको पहेँलो कछाड फेरेर बसेको मेरो देश मेरो मुटुको टुक्रा हो । सिन्धुमा हराएको विन्दु जस्तै विश्वको भूमण्डलमा सानो देखिए पनि मेरो देश मेरो प्राकृतिक दृश्यले एकदमै सुन्दर र पवित्र छ । पूर्व मेची र पश्चिम



Places

Food

Temples

Culture

Festivals

Nepal Karnali

मेरो देश नेपाल

is a small, landlocked country, situated between Tibet (China) to its north, and India to its east, west and south. At latitudes between 26 and 30 degrees north and longitudes between 80 and 88 degrees east, the country covers an area of 147,181sq.km and stretches approximately 145-241 km

Bheri

Rapti

Dhawalagiri

Gandaki

Bagmati

Lumbini

Narayani

Janakpur

Sagarmatha

What is for?

Nepal is known as the famous ultimate outdoor-sports destination, offering a wide range of sports like rock climbing, canyoning, ziplining etc.

special about nepal?

Nepal is the home to eight of the highest peak of the world among the top ten highest peaks. The country contains beautiful and amazing mountain ranges which are the reason it is termed as a Himalayan country.

Reasons to Visit Nepal?

- Trek to Everest Base Camp.
- Learn the culture of Kathmandu
- Admire the Boudhanath Stupa

Nepal

हिमालको सेतो फेरा, पहाडको हरियो :



Places

Food

Temples

Culture

Festivals

Nepal

मेरो देश

Nepal is a small, landlocked country, situated between Tibet (China) to its north, and India to its east, west and south. At latitudes between 26 and 30 degrees north and longitudes between 80 and 88 degrees east, the country covers an area of 147,181sq.km and stretches approximately 145-241 km

CSS Flexbox

CSS3 Flexible boxes also known as CSS Flexbox, is a new layout mode in CSS3.

The CSS3 flexbox is used to make the elements behave predictably when they are used with different screen sizes and different display devices. It provides a more efficient way to layout, align and distribute space among items in the container.

It is mainly used to make CSS3 capable to change its items width and height to best fit for all available spaces. It is preferred over block model.

The CSS3 flexbox contains flex containers and flex items.

Flex container:The flex container specifies the properties of the parent. It is declared by setting the display property of an element to either flex or inline-flex.

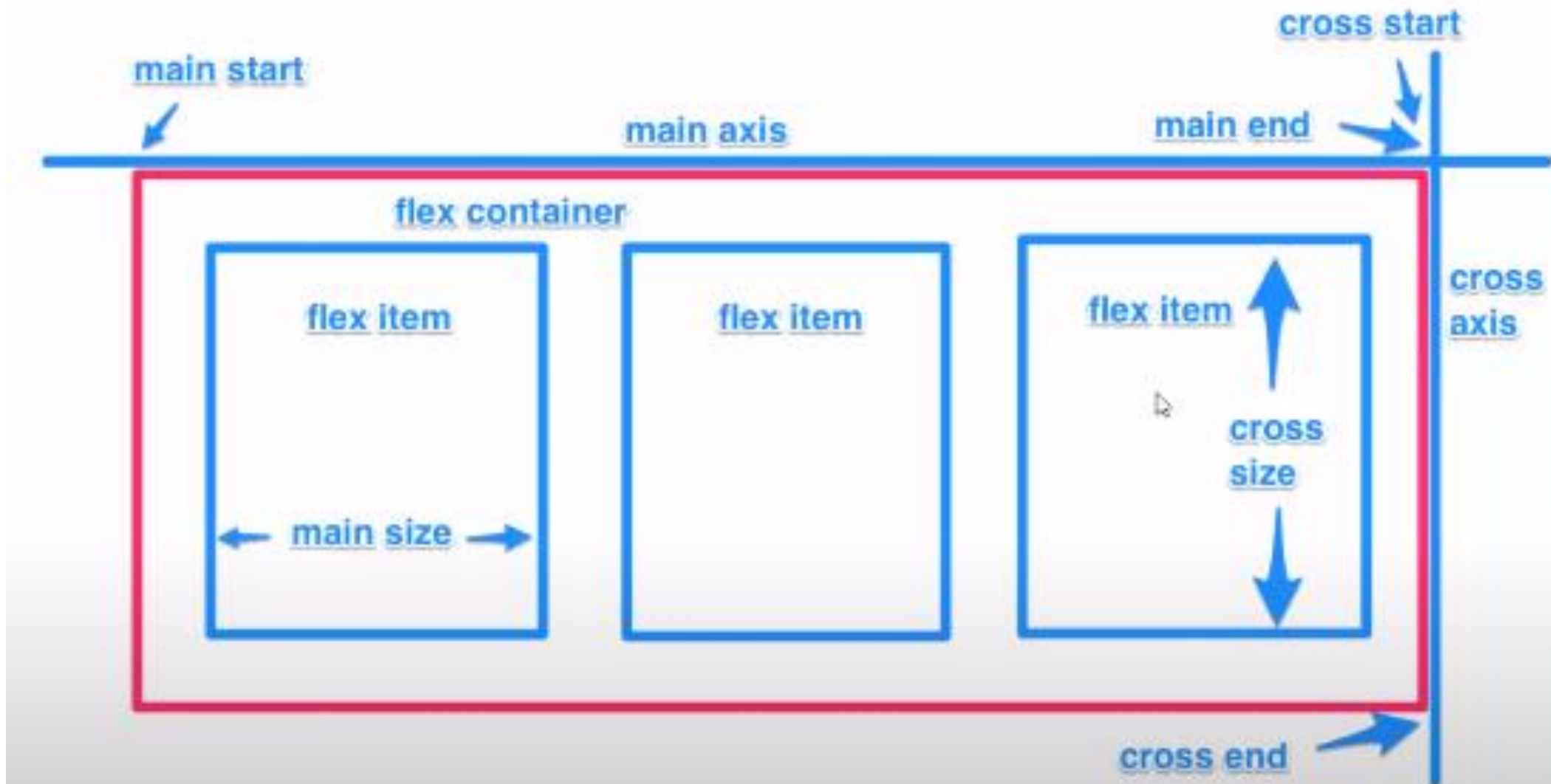
Flex items:The flex items specify properties of the children. There may be one or more flex items inside a flex container.

CSS Flexbox Layout Module

Before the Flexbox Layout module, there were four layout modes:

- Block, for sections in a webpage
- Inline, for text
- Table, for two-dimensional table data
- Positioned, for explicit position of an element

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.



The flex container becomes flexible by setting the display property to flex:

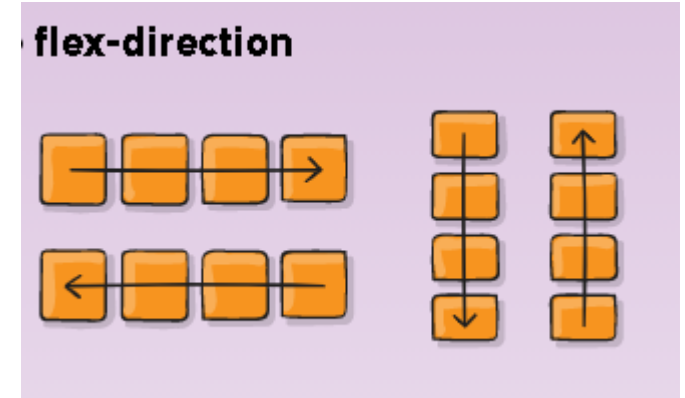
```
.flex-container {  
  display: flex;  
}
```

The flex container properties are:

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

flex-direction Property

`flex-direction: column;` Vertically from top to bottom
`flex-direction: column-reverse;` Vertically from bottom to top
`flex-direction: row;` Horizontally from left to right
`flex-direction: row-reverse;` Horizontally from right to left



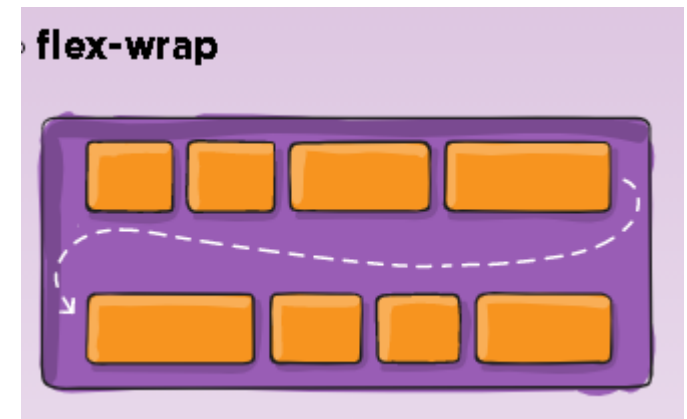
flex-wrap Property

`flex-wrap: wrap;`
`flex-wrap: nowrap;`
`flex-wrap: wrap-reverse;`

flex-flow Property

The flex-flow property is a shorthand property for setting both the flex-direction and flex-wrap properties.

`flex-flow: row wrap;`



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.flex-container {
```

```
  display: flex;
```

```
  background-color:gold;
```

```
  flex-direction: row;
```

```
}
```

```
.flex-container > div {
```

```
  background-color:black;
```

```
  margin: 10px;
```

```
  color:chartreuse;
```

```
  padding: 20px;
```

```
  font-size: 30px;
```

```
  width: 200px;
```

```
  height: 100px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="flex-container">
```

```
  <div>S <pre>1</pre></div>
```

```
  <div>I <pre>2</pre></div>
```

```
  <div>M <pre>3</pre> </div>
```

```
  <div>A <pre>4</pre></div>
```

```
  <div>N <pre>5</pre></div>
```

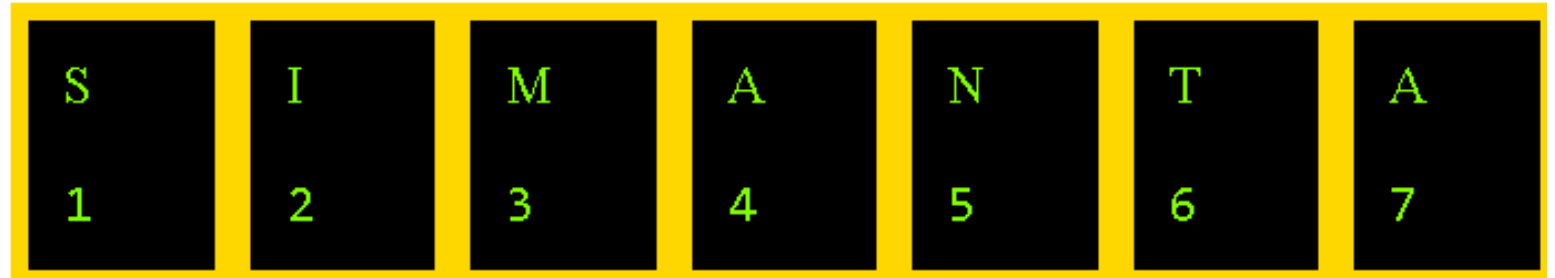
```
  <div>T <pre>6</pre></div>
```

```
  <div>A <pre>7</pre></div>
```

```
</div>
```

```
</body>
```

```
</html>
```



S
1
I
2
M
3
A
4
N
5
T
6
A
7

column

A
7
T
6
N
5
A
4
M
3
I
2
S
1

column-reverse

A	T	N	A	M	I	S
7	6	5	4	3	2	1

row-reverse

justify-content Property

`justify-content: center;` The center value aligns the flex items at the center of the container:

`justify-content: flex-start;` The flex-start value aligns the flex items at the beginning of the container (this is default):

`justify-content: flex-end;` The flex-end value aligns the flex items at the end of the container:

`justify-content: space-around;` The space-around value displays the flex items with space before, between, and after the lines:

`justify-content: space-between;` The space-between value displays the flex items with space between the lines:

justify-content

flex-start



flex-end



center



space-between



space-around



space-evenly



align-items Property

`align-items: center;`

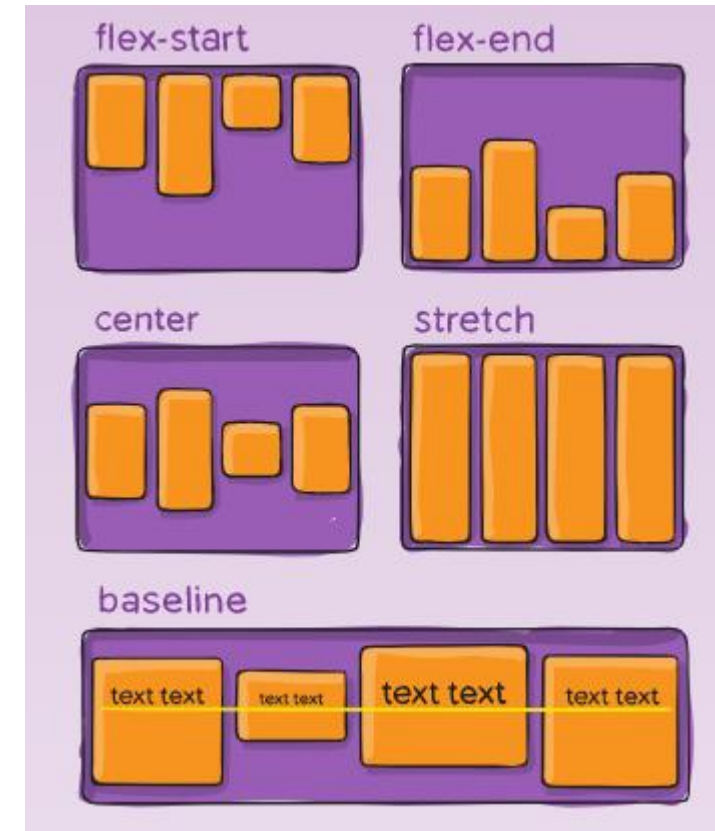
`align-items: flex-start;` The flex-start value aligns the flex items at the top of the container:

`align-items: flex-end;` The flex-end value aligns the flex items at the bottom of the container:

`align-items: stretch;` The stretch value stretches the flex items to fill the container (this is default):

`align-items: baseline;`

The baseline value aligns the flex items such as their baselines aligns:



align-content Property

`align-content: space-between;` The space-between value displays the flex lines with equal space between them:

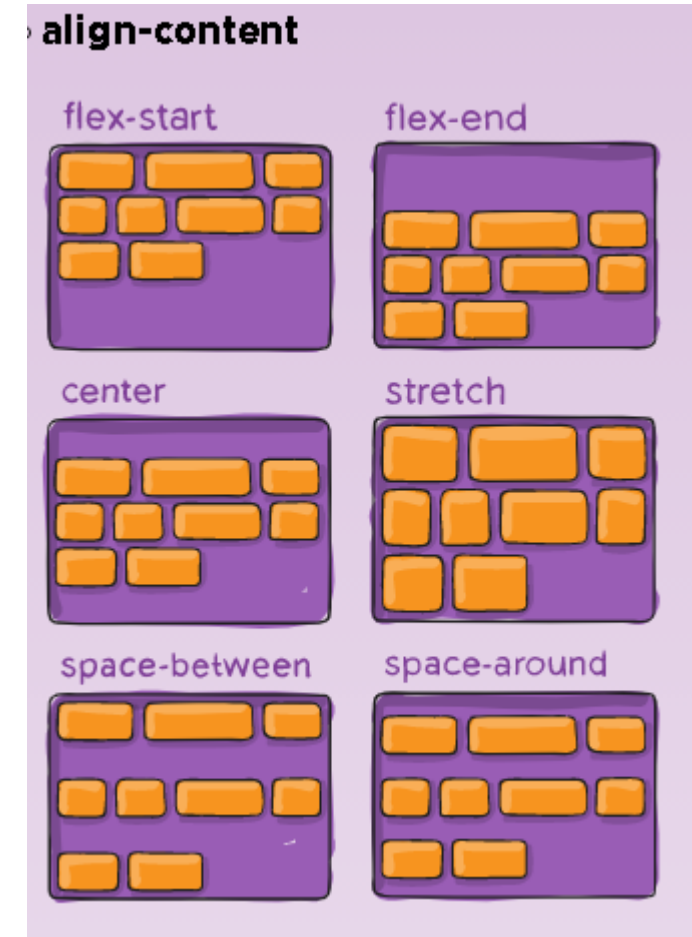
`align-content: space-around;` The space-around value displays the flex lines with space before, between, and after them:

`align-content: stretch;` The stretch value stretches the flex lines to take up the remaining space (this is default):

`align-content: center;` The center value displays display the flex lines in the middle of the container:

`align-content: flex-start;` The flex-start value displays the flex lines at the start of the container:

`align-content: flex-end;` The flex-end value displays the flex lines at the end of the container:



The CSS Flexbox Container Properties

The following table lists all the CSS Flexbox Container properties:

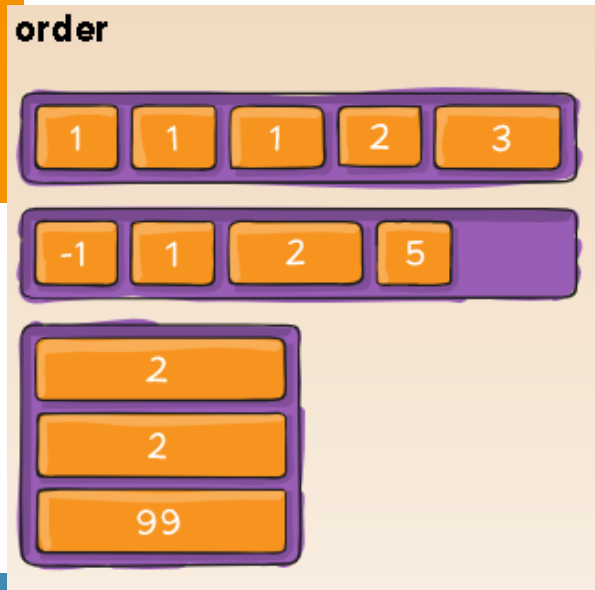
Property	Description
<u>align-content</u>	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
<u>align-items</u>	Vertically aligns the flex items when the items do not use all available space on the cross-axis
<u>display</u>	Specifies the type of box used for an HTML element
<u>flex-direction</u>	Specifies the direction of the flexible items inside a flex container
<u>flex-flow</u>	A shorthand property for flex-direction and flex-wrap
<u>flex-wrap</u>	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
<u>justify-content</u>	Horizontally aligns the flex items when the items do not use all available space on the main-axis

CSS Flex Items

Child Elements (Items)

The direct child elements of a flex container automatically becomes flexible (flex) items.

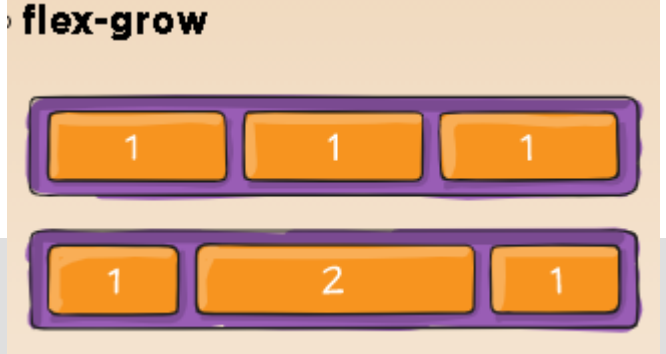
order
flex-grow
flex-shrink
flex-basis
flex
align-self



order
The order value must be a number, default value is 0. Higher the number of order appears at last.

flex-grow

The flex-grow property specifies how much a flex item will grow relative to the rest of the flex items.



flex-shrink Property

The flex-shrink property specifies how much a flex item will shrink relative to the rest of the flex items.

flex-basis Property

The flex-basis property specifies the initial length of a flex item.

The CSS Flexbox Items Properties

The following table lists all the CSS Flexbox Items properties:

Property	Description
<u>align-self</u>	Specifies the alignment for a flex item (overrides the flex container's align-items property)
<u>flex</u>	A shorthand property for the flex-grow, flex-shrink, and the flex-basis properties
<u>flex-basis</u>	Specifies the initial length of a flex item
<u>flex-grow</u>	Specifies how much a flex item will grow relative to the rest of the flex items inside the same container
<u>flex-shrink</u>	Specifies how much a flex item will shrink relative to the rest of the flex items inside the same container
<u>order</u>	Specifies the order of the flex items inside the same container

End of CSS