



Cascading Style Sheets (CSS)

ONE LOVE. ONE FUTURE.

Content

Basic CSS

Advanced CSS

A decorative background consisting of a large, circular arrangement of red dots of varying sizes. The dots are more densely packed on the right side and become sparser towards the left, creating a sense of depth and movement.

Basic CSS

ONE LOVE. ONE FUTURE.

Motivation

Welcome to My Homepage

Use the menu to select different Stylesheets

Stylesheet 1

Stylesheet 2

Stylesheet 3

Stylesheet 4

No Stylesheet

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

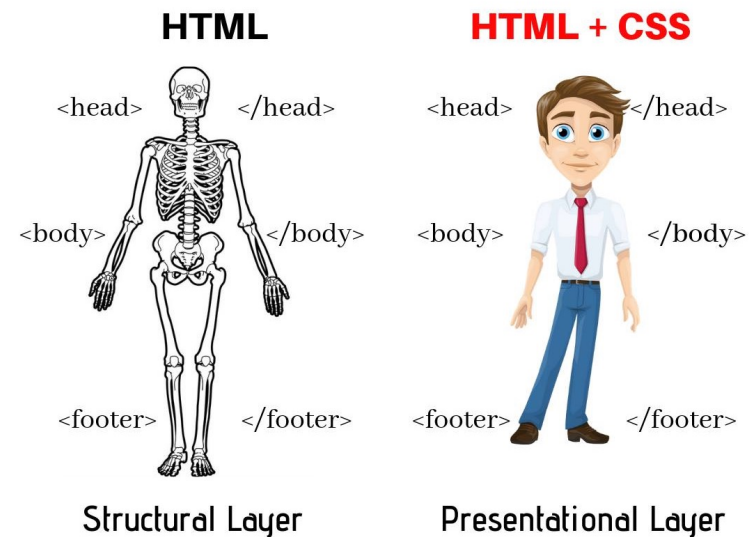
This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
[No Stylesheet](#).

https://www.w3schools.com/css/demo_default.htm

Content vs. Presentation

- Most HTML tags define content type, independence of presentation.
 - exceptions?
 - e.g. `` `` for bold text and `<i>` `</i>` for italicized text
- Style sheets associate presentation formats with HTML elements.
 - CSS1: published in 1996
 - CSS2: published in 1998
 - CSS3: divided into many separate modules

HTML Vs CSS



Content vs. Presentation (cont.)

- Style sheets can be used to specify how tables should be rendered, how lists should be presented, etc.
- HTML style sheets are known as *Cascading Style Sheets*, since can be defined at three different levels
 1. *inline* style sheets apply to the content of a single HTML element
 2. *document* style sheets apply to the whole BODY of a document
 3. *external* style sheets can be linked and applied to numerous documents
- *lower-level style sheets can **override** higher-level style sheets*

Selectors

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<h1>Hello world!</h1>
<p>Me too!</p>
<p>And me!</p>
</body>
</html>
```

Hello world!

Me too!

And me!

Inline Style Sheets

- Using the **style** attribute, we can specify presentation style for a *single* HTML element
- within tag, list sequence of **property:value** pairs separated by semi-colons

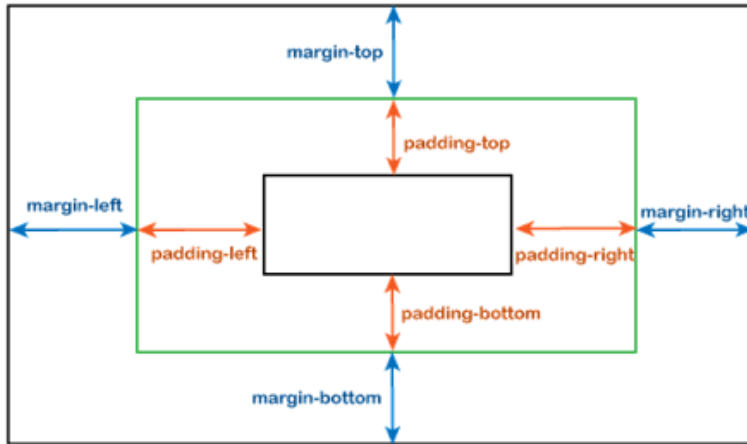
```
<body>
  <p style="font-family:Arial; text-align:right">
This is a right-justified paragraph in Arial, with some
  <span style="color:red">red text</span>.
  </p>

  <p>And
    <a style="color:red; text-decoration:none; font-
size:larger;" href="page01.html">LINK</a>
    is a formatted link.
  </p>
</body>
```

This is a right-justified paragraph in Arial, with some red text.

And LINK is a formatted link.

Inline Style Sheets (cont.)



Text

A. one thing
B. or another

- with this
- or that

```
<!DOCTYPE html>
<html>
<head>
<title>Inline Style Sheets</title>
</head>
<body>
<p style="margin-left:50px; margin-top:30px"> Text
</p>

<ol style="list-style-type:upper-alpha">
<li> one thing</li>
<li> or another</li>
  <ul style="list-style-type:square">
    <li> with this</li>
    <li> or that</li>
  </ul>
</ol>
</body>
</html>
```

Document Style Sheets

- Inline style sheets apply to *individual elements* in the page.
 - using inline style directives can lead to inconsistencies, as similar elements are formatted differently
 - inline definitions mix content & presentation
- As a general rule, inline style sheet directives should be used as sparingly as possible
- Alternatively, document style sheets allow for a cleaner separation of content and presentation.
 - style definitions are placed in the <head> of the page (within STYLE tags)
 - can apply to all elements, or a subclass of elements, throughout the page.

Document Style Sheets

Centered Title

This paragraph will have the first line indented, but subsequent lines will be flush.

This paragraph will not be indented.

The End

```
<!DOCTYPE html>
<html>
<head>
  <title>Document Style Sheets</title>
  <style type="text/css">
    h1 {color:blue;
        text-align:center}
    p.indented {text-indent:0.2in}
  </style>
</head>
<body>
  <h1> Centered Title </h1>
  <p class="indented">This paragraph
will have the first line indented, but
subsequent lines will be flush. </p>

  <p>This paragraph will not be
indented.</p>
</body>
</html>
```

Document Style Sheets (cont.)

Danh sách SV

Tên	MSSV
-----	------

Minh	20202020
------	----------

Mạnh	20212021
------	----------

```
<!DOCTYPE html>
<html>
<head>
  <title> Inline Style Sheets </title>
  <style type="text/css">
    table {font-family:Arial,sans-serif}
    caption {color:red;
              font-style:italic;
              text-decoration:underline}
    th {background-color:green}
  </style>
</head>
<body>
  <table>
    <caption> Danh sách SV </caption>
    <tr><th> Tên </th> <th> MSSV</th></tr>
    <tr><td> Minh </td> <td> 20202020 </td></tr>
    <tr><td> Mạnh </td> <td> 20212021 </td></tr>
  </table>
</body>
</html>
```

External Style Sheets

- modularity is key to the development and reuse of software
 - package and make available for reuse
 - central libraries make it possible to make a single change and propagate the changes
- external style sheets place the style definitions in separate files
 - multiple pages can link to the same style sheet, consistent look across a site
 - possible to make a single change and propagate automatically
 - represents the ultimate in content/representation separation

Modularity & Style Sheets

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

This is a heading

This is a paragraph.

```
<!DOCTYPE html>  
<html>  
<head>  
<link rel="stylesheet"  
href="mystyle.css">  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>
```

A decorative background consisting of a large, circular arrangement of red dots of varying sizes. The dots are more densely packed on the right side and become sparser towards the left, creating a sense of depth and movement.

Advanced CSS

ONE LOVE. ONE FUTURE.

Universal, Child, and Adjacent Selectors

- **Universal selectors**: set global styles for a page, or as a descendant of a selector to set styles of everything within something.

```
* {  
    margin: 0;  
    padding: 0;  
}  
#contact * {  
    display: block;  
}
```

Example: set the margin and padding on everything in a page to zero and everything within an element with the ID “contact” to be displayed as a block

Universal, Child, and Adjacent Selectors

- **Child selectors:** A greater-than symbol (“>”) can be used to specify something that is a child of something else, that is, something immediately nested within something.

```
#genus_examples > li { border: 1px solid red }
```

Example: set the border for all child of element has id=“genus_examples”

Universal, Child, and Adjacent Selectors

- **Adjacent selectors:** A plus sign (“+”) is used to target an adjacent sibling of an element, essentially, something immediately following something.

```
<h1>Clouded leopards</h1>  
<p>Clouded leopards are cats that  
belong to the genus Neofelis.</p>  
<p>There are two extant species:  
Neofelis nebulosa and Neofelis  
diardi.</p>
```

```
h1 + p { font-weight: bold }
```

Only the first paragraph, that following the heading, will be made bold.

Position

- specifies the type of positioning method used for an element
 - static: is not affected by top, bottom, left, and right.
 - relative: is positioned relative to its normal position.
 - fixed: stays in the same place even if the page is scrolled
 - absolute: is positioned relative to the nearest positioned ancestor (or document body if no ancestor exists)
 - sticky: is positioned based on the user's scroll position.

```
div.relative {  
  position: relative;  
  left: 20px;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}
```

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

This div element has position: relative;

This div element has position: absolute;

Position

```
div.static {  
  position: static;  
  width: 300px;  
  height: 100px;  
  left: 50px;  
  border: 3px solid red;  
}
```

```
div.relative {  
  position: relative;  
  width: 300px;  
  height: 100px;  
  left: 50px;  
  border: 3px solid blue;  
}
```

```
div.absolute {  
  position: absolute;  
  top: 20px;  
  width: 200px;  
  height: 50px;  
  border: 3px solid green;  
}
```

```
div.fixed {  
  position: fixed;  
  top: 50px;  
  left: 50px;  
  width: 200px;  
  height: 50px;  
  border: 3px solid yellow;  
}
```

```
div.sticky {  
  position: sticky;  
  top: 20px;  
  width: 200px;  
  height: 50px;  
  border: 3px solid black;  
}  
</style>  
</head>
```

```
<body>  
<div class="static">This div position: static;</div>  
<div class="relative">This div position: relative  
<div class="absolute">This div position: absolute;</div>  
<div class="fixed">This div position: fixed;</div>  
</div>  
<div class="sticky">This div position: sticky;</div>
```

This div position: static;

This div position: fixed;

This div position: relative

This div position: absolute;

This div position: sticky;

Overlap

```
.gray-box {  
  background: lightgray;  
  height: 60px;  
}
```

```
.green-box {  
  position: relative;  
  background: lightgreen;  
  left: 50px;  
  top: -20px;  
  height: 60px;  
}
```

```
<body>  
  <div class="gray-box">Gray box</div>  
  <div class="green-box">Green box</div>  
</body>
```

Gray box

Green box

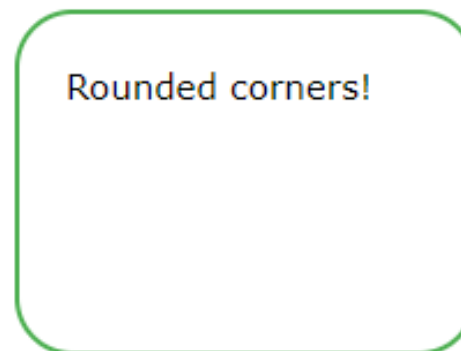
```
.green-box {  
  position: relative;  
  background: lightgreen;  
  left: 50px;  
  top: -20px;  
  height: 60px;  
  z-index: -1;  
}
```

Gray box

Rounded Corners

- With the CSS border-radius property, you can give any element “rounded corners”.
 - Rounded corners for an element with a border:

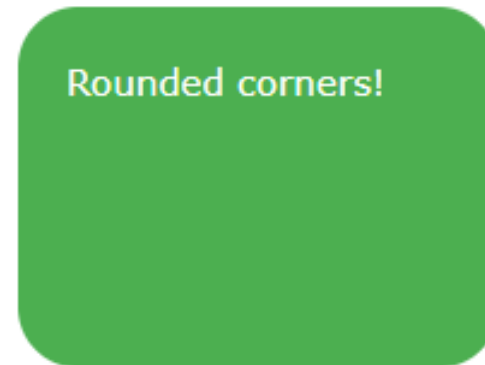
```
#rcorners2 {  
  border-radius: 25px;  
  border: 2px solid #73AD21;  
  padding: 20px;  
  width: 200px;  
  height: 150px;  
}
```



Rounded Corners

- With the CSS border-radius property, you can give any element “rounded corners”.
 - Rounded corners for an element with a specified background color:

```
#rcorners1 {  
  border-radius: 25px;  
  background: #73AD21;  
  padding: 20px;  
  width: 200px;  
  height: 150px;  
}
```



Responsive Web Design

- Design for multi-devices with different screen sizes, resolutions

- HTML5

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

- Media query: technique in CSS3

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

- Example

- https://www.w3schools.com/css/example_withoutviewport.htm
- https://www.w3schools.com/css/example_withviewport.htm

References

1. <https://www.w3schools.com/css/default.asp>
2. https://developer.mozilla.org/en-US/docs/Learn_web_development/Core

Exercises

- <https://www.freecodecamp.org/learn/2022/responsive-web-design/>
- Exercises:
 - Learn HTML by Building a Cat Photo App
 - Build a Personal Portfolio Webpage

