

CSS

N.Nalini

AP(Sr)

SCOPE

VIT

Any **modification** in the design of websites was a very **difficult** and **boring** task , as it evolves **manually editing** every HTML page.



# What is CSS

- **C**ascading **S**tyle **S**heets
- Contains the rules for the **presentation** of HTML.



- CSS was introduced to keep the **presentation** information **separate** from **HTML** markup (content).

HTML + CSS + JavaScript  
Content + Style + Behavior

## CSS1

- Font properties such as typeface and emphasis
- Color of text, backgrounds, and other elements
- Text attributes such as spacing between words, letters, and lines of text
- Alignment of text, images, tables and other elements
- Margin, border, padding, and positioning for most elements
- Unique identification and generic classification of groups of attributes

## CSS2

- absolute, relative, and fixed positioning of elements and z-index,
- the concept of media types
- support for aural style sheets and bidirectional text
- new font properties such as shadows.

## CSS3

- Borders (border-radius, box-shadow)
- Backgrounds (multiple backgrounds)
- Color (HSL colors, HSLA colors, opacity, RGBA colors)
- media queries
- multi-column layout
- Web fonts

# Sources of Styles

## Author (developer) Styles

- **Inline Styles** - As inline attribute “style” inside HTML tags

```
<div style="font-weight: bold;">I am bold</div>
```

- **Embedded/internal Styles** - As embedded style tag with in HTML document.

```
<html>
  <head>
    <title>Welcome to Vendio!</title>
    <style>
      .footer {
        width:90%;
      }
    </style>
    -----
  </html>
```

- **Linked/external Styles** - Inside separate files with .css extension  
`<link rel="stylesheet" href="external.css" type="text/css" />`

# Sources of Styles(contd.)

- **User Style sheets**

This file contains the user created styles .

`[firefox profile folder]/ chrome/userContent-example.css` is the current user's style sheet file for the firefox.

- **Browser default style sheet**

This file contains default styles for all users of a browser

`[firefox folder]/res/html.css` is the default style sheet file for the firefox.

# Types of CSS selectors

Styles assigned to a HTML element

- Tag selector
- Group selector
- Class selector
- ID selector
- Pseudo-classes



# CSS Selectors

- ID based ( #)

## HTML

```
<div id="content">  
  Text  
</div>
```

## CSS

```
#content {  
  width: 200px;  
}
```

ID selectors should be used with **single** elements.

# Class based selector

- Class (.)

## HTML

```
<div class="big">
  Text
</div>
<div>
  <span class="big">some text </span>
</div>
```

## CSS

```
.big{
  width: 200px;
}
```

Class based styles can be used by **multiple** HTML elements.

# Tag based selectors

- Tag (Tag name)

## HTML

```
<div>
  Text
</div>
<div>
  <span>some text </span>
</div>
<span>some other text </span>
```

## CSS

```
DIV {
    width: 200px;
}
SPAN {
    font-size:130%;
}
```

# Grouping

- Multiple selectors can be grouped in a single style declaration by using , .

```
H1, P, .main {  
    font-weight:bold;  
}
```

# Universal selectors

Universal selectors are used to select any element.

```
* {  
  color: blue;  
}
```

# Attribute selectors

Attribute selectors selects elements based upon the attributes present in the HTML Tags and their value.

```
IMG[src="small.gif"] {  
    border: 1px solid #000;  
  
}
```

will work for

```

```

# CSS Pseudo-classes

A pseudo-class is used to define a special state of an element. `selector:pseudo-class { property: value }`

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

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

- `:link`
- `:visit`
- `:hover`
- `:active`
- `:focus`
- `:lang()`
- `:left`
- `:right`
- `:first`
- `:first-child`

## CSS Pseudo-elements

- `:first-letter`
- `:first-line`
- `:before`

```
<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>

<p><b><a href="default.asp" target=" blank">This is a link</a></b></p>
```

```
<!DOCTYPE html>
<html>
<head>
<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>
<p><b>Note:</b> For :first-child to work in IE8 and earlier, a DOCTYPE must be
declared.</p>

</body>
</html>
```

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

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

**Note:** For :first-child to work in IE8 and earlier, a DOCTYPE must be declared.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<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

**Note:** IE8 supports the :focus pseudo-class only if a !DOCTYPE is specified.


```
<!DOCTYPE html>
<html>
<head>
<style>
h1::before {
  content: url(smiley.gif);
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>The ::before pseudo-element inserts content before
the content of an element.</p>

<h1>This is a heading</h1>
<p><b>Note:</b> IE8 supports the content property
```

 **This is a heading**

The ::before pseudo-element inserts content b

 **This is a heading**

**Note:** IE8 supports the content property only i

```
<head>
<style>
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  padding: 12px 16px;
  z-index: 1;
}

.dropdown:hover .dropdown-content {
  display: block;
}
</style>
</head>
<body>

<h2>Hoverable Dropdown</h2>
<p>Move the mouse over the text below to open the dropdown content.</p>

<div class="dropdown">
  <span>Mouse over me</span>
  <div class="dropdown-content">
    <p>Hello World!</p>
  </div>
</div>

</body>
</html>
```

## Hoverable Dropdown

Move the mouse over the text below to open the dropdown content.

Mouse over me

# 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:
  - descendant selector (space)
  - child selector (>)
  - adjacent sibling selector (+)
  - general sibling selector (~)

# Descendant selectors

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

## HTML

```
<div class="abc">
  <section>
    <P>
      Hello there!
    </p>
  </section>
</div>
<div>XXX
</div>
```

## CSS

```
div.abc P {
  font-weight:bold;
}
```

# Child selectors

The child selector selects all elements that are the children of a specified element. Child selectors will not select all descendants, only direct children.

HTML

```
<div >
  <div class="abc">
    <P>
      Hello there!
    </p>
  <section> <P> xxx</P></section>
</div>
</div>
```

CSS

```
DIV.abc > P {
  font-weight:bold;
}
```

# General Sibling Selector

- The general (subsequent) 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:

```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>

<p>Paragraph 1.</p>

<div>
  <p>Paragraph 2.</p>
</div>

<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>

</body>
</html>
```

Paragraph 1.

Paragraph 2.

Paragraph 3.

Some code.

Paragraph 4.

# Adjacent sibling selectors

It will select the sibling immediately following an element.

```
DIV.abc + P {  
    font-weight: bold;  
}  
DIV#f + P {  
    color: blue;  
}
```

```
<div  
    class="abc">Message</div>  
<P>Hello there!</p>  
<P>Hello there!</p>  
  
<div id="f">Message</div>  
<P>Hello there!</p>  
<P>Hello there!</p>
```



# CSS Values

- **Words:** text-align:center;.
- **Numerical values:** Numerical values are usually followed by a unit type.  
font-size:12px;  
12 is the numerical value and px is the unit type pixels.
  - Absolute Values – in, pc, px, cm, mm, pt
  - Relative Values – em, ex, %
- **Color values:** color:#336699 or color#369 or rgb(255, 255, 255).

**Table 8-2** CSS length units

Unit	Name	Description
cm	Centimeters	Measures values in centimeters
em	Em space	Measures values according to width of the uppercase letter <i>M</i> for the font that is being used
ex	x-height	Measures values according to the height of the lowercase letter <i>x</i> for the font that is being used
in	Inches	Measures values in inches
mm	Millimeters	Measures values in millimeters
pc	Picas	Measures values in picas, which are equal to $\frac{1}{6}$ of an inch
pt	Points	Measures values in points, which are equal to $\frac{1}{72}$ of an inch
px	Pixels	Measures values in pixels

# Categories of CSS properties

- Positioning and layout handling related.
- Background related properties.
- Font and text related
- Links related.
- Lists related.
- Table related.

# CSS Text

- Text-indent
- Text-align
- Text-decoration
- Letter-spacing
- Text-transform
- Word-spacing

# Text and Table

<b>text-align</b>	start   end   left   right   center   justify
<b>text-align-last</b>	start   end   left   right   center   justify
<b>text-decoration</b>	none   underline   overline   line-through   blink
<b>text-emphasis</b>	none   [ [ accent   dot   circle   disc ] [ before   after ]? ]
<b>text-indent</b>	<i>length</i> %
<b>text-justify</b>	auto   inter-word   inter-ideograph   inter-cluster   distribute   kashida   tibetan
<b>text-outline</b>	none <i>color</i> <i>length</i>
<b>text-shadow</b>	none <i>color</i> <i>length</i>
<b>text-transform</b>	none   capitalize   uppercase   lowercase
<b>text-wrap</b>	normal   unrestricted   none   suppress

TABLE	
<b>border-collapse</b>	collapse   separate
<b>border-spacing</b>	<i>length length</i>
<b>caption-side</b>	top   bottom   left   right
<b>empty-cells</b>	show   hide
<b>table-layout</b>	auto   fixed

# **Text Decoration in CSS**

```
<style>
```

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

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

```
h3 {  
    text-decoration: underline;  
}
```

```
</style>
```

# Text Decoration in CSS

---

**This is heading 1**

~~**This is heading 2**~~

**This is heading 3**

# Font Properties

- font-style: *italic*;
- font-family: times, arial, serif, sans-serif, monospace;
- font-weight: (**bold**, **bolder**, lighter, or 100 –**900**;) )
- font-size: *size*;

font: style weight *size* family;



FONT	
<b>font</b>	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption   icon   menu   message-box   small- caption   status-bar
<b>font-family</b>	<i>family-name</i> <i>generic-family</i> inherit
<b>font-size</b>	xx-small   x-small   small   medium   large   x-large   xx-large   smaller   larger   inherit <i>length</i> %
<b>font-size-adjust</b>	none   inherit <i>number</i>
<b>font-stretch</b>	normal   wider   narrower   ultra-condensed   extra- condensed   condensed   semi-condensed   semi- expanded   expanded   extra-expanded   ultra- expanded   inherit
<b>font-style</b>	normal   italic   oblique   inherit
<b>font-variant</b>	normal   small-caps   inherit
<b>font-weight</b>	normal   bold   bolder   lighter   100   200   300   400   500   600   700   800   900   inherit

# Shorthand Font Property

- **font**

- Shorthand rule for setting multiple font properties at the same time

```
font:italic normal bold 12px/16px verdana
```

is equal to writing this:

```
font-style: italic;  
font-variant: normal;  
font-weight: bold;  
font-size: 12px;  
line-height: 16px;  
font-family: verdana;
```

# CSS Background

- **background-image:** url(../location/of/image.jpg)
- **background-repeat:** tile image in background
- **background-position:** *vertical* (top, center, bottom, or *size*) *horizontal* (left, center, right, or *size*)
- **background-attachment:** (*scroll* or *fixed*)

# Background

BACKGROUND	
<b>background</b>	<i>background-image</i> <i>background-position</i> <i>background-size</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-origin</i> <i>background-clip</i> <i>background-color</i>
<b>background-attachment</b>	scroll   fixed
<b>background-break</b>	bounding-box   each-box   continuous
<b>background-clip</b>	<i>length</i> % border-box   padding-box   content-box   no-clip
<b>background-color</b>	<i>color</i> transparent
<b>background-image</b>	<i>url</i> none
<b>background-origin</b>	border-box   padding-box   content-box
<b>background-position</b>	top left   top center   top right   center left   center center   center right   bottom left   bottom center   bottom right <i>x-% y-%</i> <i>x-pos y-pos</i>
<b>background-repeat</b>	repeat   repeat-x   repeat-y   no-repeat
<b>background-size</b>	<i>length</i> % auto   cover   contain

```
<style>      CSS Background
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>W3Schools background image example.</p>
<p>The background image is only showing once, but
    it is disturbing the reader!</p>
</body>
```

Note:img{float:left}

# Hello World!

W3Schools background image example.

The background image is only showing once, but it is disturbing the reader!



# Background Shorthand Property

- **background**: shorthand rule for setting background properties at the same time:

```
background: #FFF0C0 url("back.gif") no-repeat  
fixed top;
```

is equal to writing:

```
background-color: #FFF0C0;  
background-image: url("back.gif");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: top;
```

- Some browsers will not apply BOTH color and image for background if using shorthand rule

# CSS Lists

- **list-style-type:** *none, disc, circle, square*, (other types available)
- **list-style-position:** *inside* or *outside*
- **list-style-image:** url (../path/to/image.jpg)

**list-style:** type position image



# List

LIST & MARKERS	
<b>list-style</b>	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>
<b>list-style-image</b>	none <i>url</i>
<b>list-style-position</b>	Inside   outside
<b>list-style-type</b>	none   asterisks   box   check   circle   diamond   disc   hyphen   square   decimal   decimal-leading- zero   lower-roman   upper- roman   lower-alpha   up- per-alpha   lower-greek   lower-latin   upper-latin   hebrew   armenian   geor- gian   cjk-ideographic   hiragana   katakana   hira- gana-iroha   katakana-iroha   footnotes
<b>marker-offset</b>	auto <i>length</i>

# Border Properties

- **border-width:** (thin, medium, thick, or *size*)
- **border-style:** (none, hidden, dotted, dashed, solid, double, groove, ridge, inset, or outset)
- **border-color:** *color*
- **border(-top, -right, -left, -bottom):** width style color

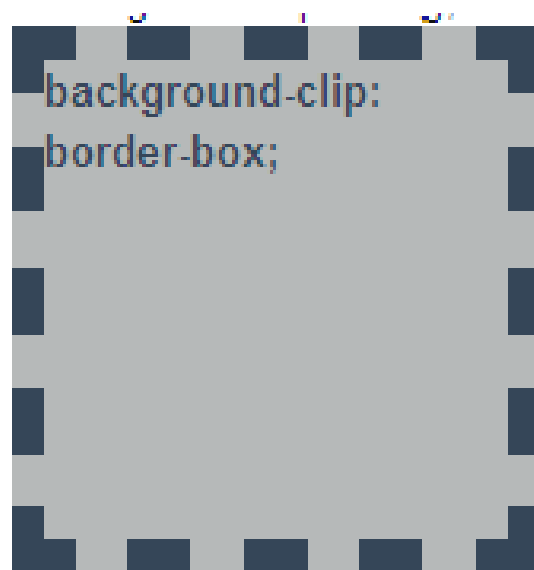
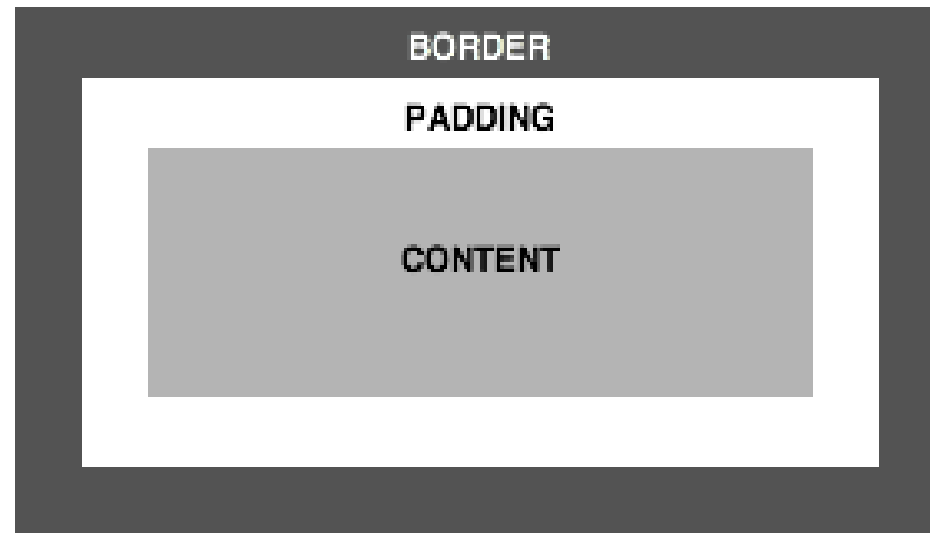


Figure 1: `background-clip: border-box;`

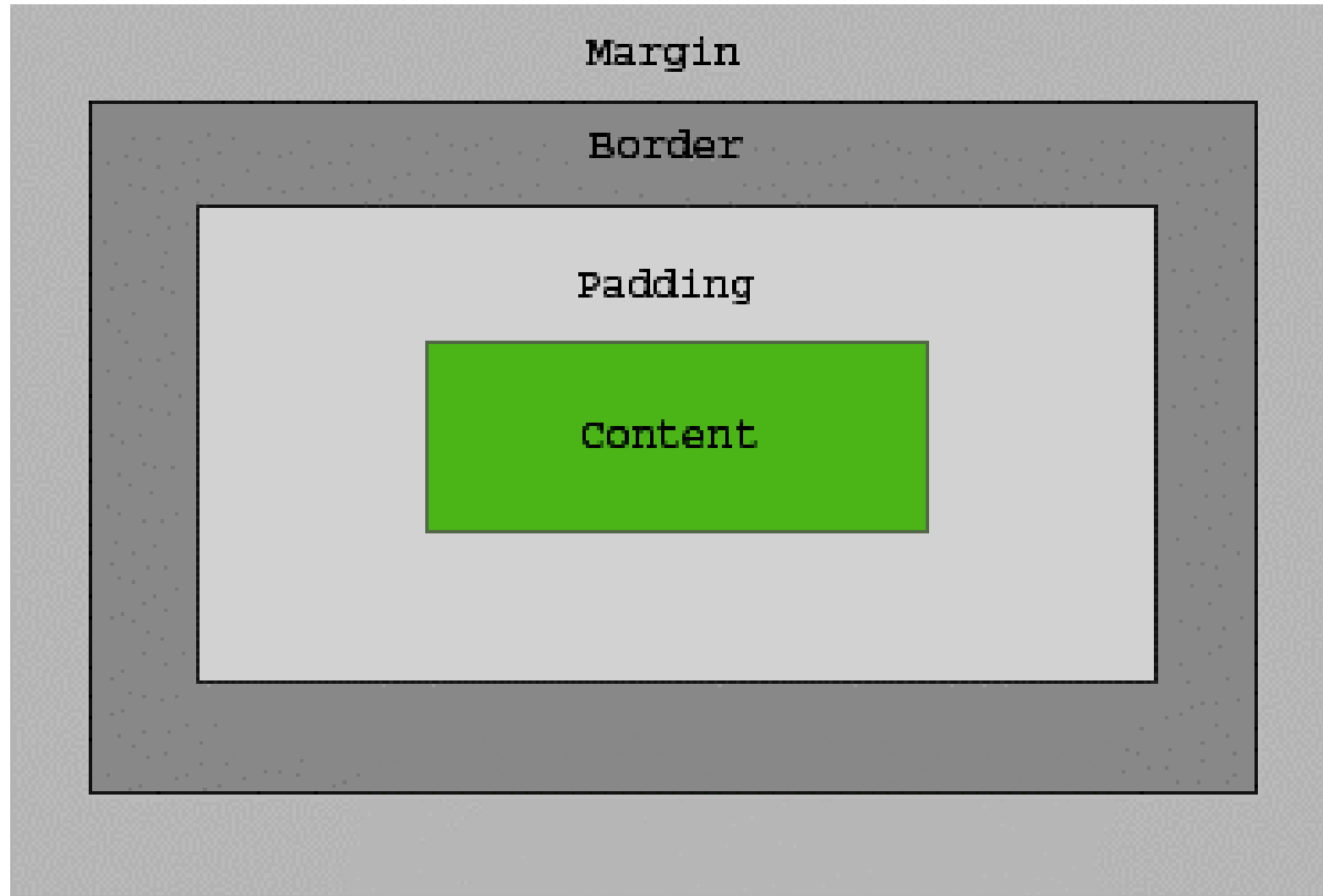


# Border Model

<b>border-bottom</b>	<i>border-bottom-width border-style border-color</i>
<b>border-bottom-color</b>	<i>border-color</i>
<b>border-bottom-style</b>	<i>border-style</i>
<b>border-bottom-width</b>	<i>thin   medium   thick length</i>
<b>border-collapse</b>	<i>collapse   separate</i>
<b>border-color</b>	<i>color</i>
<b>border-image</b>	<i>image [ number / % border-width stretch   repeat   round ] none</i>

<b>border-left</b>	<i>border-left-width border-style border-color</i>
<b>border-left-color</b>	<i>border-color</i>
<b>border-left-style</b>	<i>border-style</i>
<b>border-left-width</b>	<i>thin   medium   thick length</i>
<b>border-right</b>	<i>border-right-width border-style border-color</i>

# The Box Model



# Boxer Model

<b>margin</b>	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	<b>padding</b>	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>
<b>margin-bottom</b>	<i>auto</i> <i>length</i> <i>%</i>	<b>padding-bottom</b>	<i>length</i> <i>%</i>
<b>margin-left</b>	<i>auto</i> <i>length</i> <i>%</i>	<b>padding-left</b>	<i>length</i> <i>%</i>
<b>margin-right</b>	<i>auto</i> <i>length</i> <i>%</i>	<b>padding-right</b>	<i>length</i> <i>%</i>
<b>margin-top</b>	<i>auto</i> <i>length</i> <i>%</i>	<b>padding-top</b>	<i>length</i> <i>%</i>
		<b>marquee-direction</b>	<i>forward</i>   <i>reverse</i>
		<b>marquee-loop</b>	<i>infinite</i> <i>number</i>
		<b>marquee-play-count</b>	<i>infinite</i> <i>integer</i>
		<b>marquee-speed</b>	<i>slow</i>   <i>normal</i>   <i>fast</i>
		<b>marquee-style</b>	<i>scroll</i>   <i>slide</i>   <i>alternate</i>

# Cascading Inheritance

- Nested elements inherit the properties from the its parent



- If you specify a style for the `<body>` tag it will affect all content in your HTML page.
- If you want to override inherited settings, you need to specify a style in a more local element

```
body {  font-family: Verdana;
        font-size: 14px; }
```

```
body {  font-family: Verdana;
        font-size: 1.1em; }
.littletext { font-size: 8px; }

<body>
This text is larger.
<p class="littletext">This text is
smaller.</p>
```

# Inheritance

- Styles that relate to **text and appearance** are **inherited** by the descendant elements.
- Styles that relate to the appearance of boxes created by styling DIVs, paragraphs, and other elements, such as **borders, padding, margins** are **not inherited**.



# Cascade

The CSS cascade assigns a weight to each style rule. When several rules apply, the one with the greatest weight takes precedence.

Order of preference for various styles:

- Default browser style sheet (**weakest**)
- User style sheet
- Author style sheet
- Author embedded styles
- Author inline styles (**strongest**)

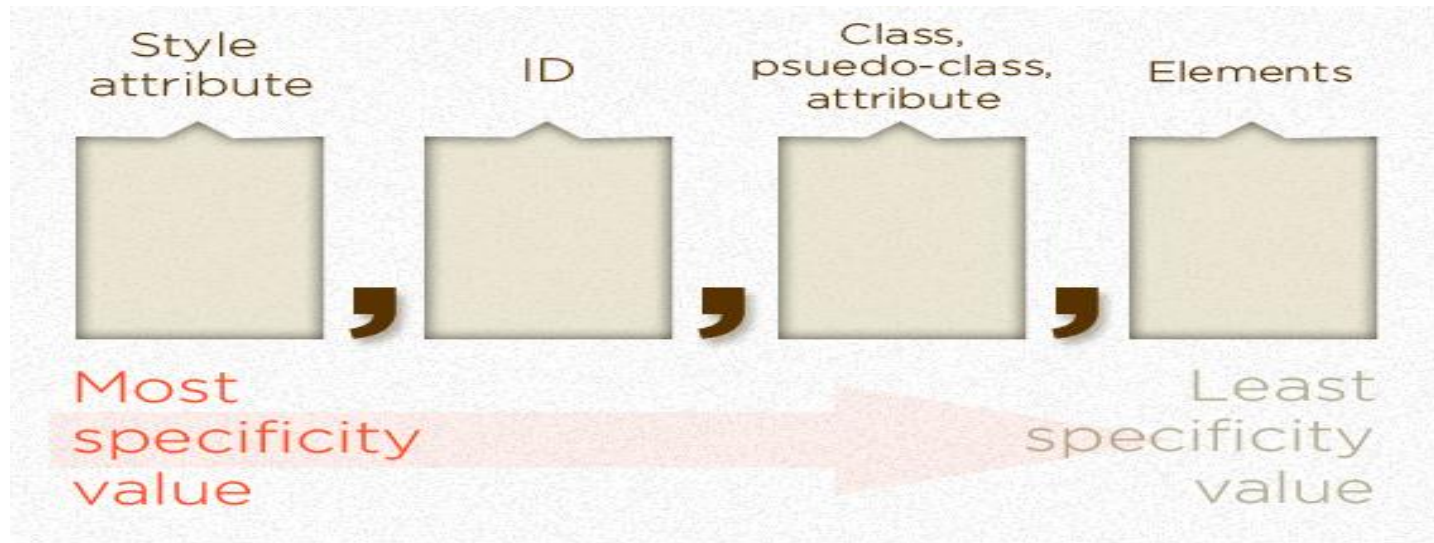


# CSS Specificity

If there are two or more conflicting CSS rules that point to the same element, the browser follows some rules to determine which one is most specific and therefore wins out.

Rule 1. Inline > Embedded > CSS File

Rule 2. ID > class > TAG



CSS selector	Priority/specificity (inline-id-class-element)
style=""	1000
#id #id2 {}	0200
#id .class {}	0110
div#id {}	0101
#id {}	0100
div.class .class2 {}	0021
div.class {}	0011
div p {}	0002
div {}	0001

#### Additional rules:

- if the declarations have the same priority, than the last declared have higher priority;
- selector \* has no weight;
- pseudo classes and attribute selector have same weight as classes;
- !important declarations are higher in priority than normal;
- if the priority is the same, than the order styles is important (last is more important);

# Animation

# Animation

- An animation lets an element gradually change from one style to another.
- You can change as many CSS properties you want, as many times you want.
- To use CSS animation, you must first specify some keyframes for the animation.
- Keyframes hold what styles the element will have at certain times.

# The @keyframes Rule

- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
- To get an animation to work, you must bind the animation to an element.

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  -webkit-animation-name: example; /* Safari 4.0 - 8.0 */
  -webkit-animation-duration: 4s; /* Safari 4.0 - 8.0 */
  animation-name: example;
  animation-duration: 4s;
}

/* Safari 4.0 - 8.0 */
@-webkit-keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* Standard syntax */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```



```
div {
  width: 100px;
  height: 100px;

  position: relative;
  /* Safari 4.0 - 8.0 */
  -webkit-animation-name: example;
  -webkit-animation-duration: 5s;
  -webkit-animation-timing-function: linear;
  -webkit-animation-delay: 2s;
  -webkit-animation-iteration-count: infinite;
  -webkit-animation-direction: alternate;
  /* Standard syntax */
  animation-name: example;
  animation-duration: 5s;
  animation-timing-function: linear;
  animation-delay: 2s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}

/* Safari 4.0 - 8.0 */
@-webkit-keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}

/* Standard syntax */
@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
```





Property	Description
<a href="#"><u>@keyframes</u></a>	Specifies the animation code
<a href="#"><u>animation</u></a>	A shorthand property for setting all the animation properties
<a href="#"><u>animation-delay</u></a>	Specifies a delay for the start of an animation
<a href="#"><u>animation-direction</u></a>	Specifies whether an animation should be played forwards, backwards or in alternate cycles(normal,reverse,alternate and alternate-reverse)
<a href="#"><u>animation-duration</u></a>	Specifies how long time an animation should take to complete one cycle
<a href="#"><u>animation-fill-mode</u></a>	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
<a href="#"><u>animation-iteration-count</u></a>	Specifies the number of times an animation should be played(number/infinite)
<a href="#"><u>animation-name</u></a>	Specifies the name of the @keyframes animation
<a href="#"><u>animation-play-state</u></a>	Specifies whether the animation is running or paused
<a href="#"><u>animation-timing-function</u></a>	Specifies the speed curve of the animation(linear,ease,ease-in/out,ease-in-out, cubic-Bezier(n,n,n,n))

# MULTIPLE COLUMN

## Property

## Description

[column-count](#)

Specifies the number of columns an element should be divided into

[column-fill](#)

Specifies how to fill columns

[column-gap](#)

Specifies the gap between the columns

[column-rule](#)

A shorthand property for setting all the column-rule-\* properties

[column-rule-color](#)

Specifies the color of the rule between columns

[column-rule-style](#)

Specifies the style of the rule between columns

[column-rule-width](#)

Specifies the width of the rule between columns

[column-span](#)

Specifies how many columns an element should span across

[column-width](#)

Specifies a suggested, optimal width for the columns

[columns](#)

A shorthand property for setting column-width and column-count

```
<html>
<head>
<style>
.newspaper {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;
  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
  -moz-column-gap: 40px; /* Firefox */
  column-gap: 40px;
  -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */
  -moz-column-rule-style: solid; /* Firefox */
  column-rule-style: solid;
}
</style>
</head>
<body>
<div class="newspaper">
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad
minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut
aliquip ex ea commodo consequat.
</div>
</body>
</html>
```

The column-rule-style property sets the style of the rule between the columns of the element:

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

# USER INTEREFACE

- **resize** – Specifies whether or not an element is resizable by the user

Values: None, horizontal, vertical and both

- **outline-offset** - adds space between an outline and the edge or border of an element.

# USER INTERFACE

```
<html>
<head>
<style>
div {
    border: 2px solid;
    width: 300px;
    resize: horizontal;
    overflow: auto;
    outline: 5px dashed blue;
    outline-offset: 5px;
}
</style>
</head>
<body>
<h1>The resize Property</h1>

<div>
<p>Let the user resize only the width of this div element.</p>
<p>To resize: Click and drag the bottom right corner of this div element.</p>
</div>

<p><b>Note:</b> Internet Explorer does not support the resize property.</p>

</body>
</html>
```

## The resize Property

Let the user resize only the width of this div element.

To resize: Click and drag the bottom right corner of this div element.

## The resize Property

Let the user resize only the width of this div element.

To resize: Click and drag the bottom right corner of this div element.

# CSS Media Types

- The @media rule, introduced in CSS2, made it possible to define different style rules for different media types.
- Examples: You could have one set of style rules for computer screens, one for printers, one for handheld devices, one for television-type devices, and so on.
- Unfortunately these media types never got a lot of support by devices, other than the print media type.



# 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 not|only *mediatype* and (*expressions*) {  
CSS-Code;  
}
- <link rel="stylesheet" media="*mediatype* and|not|only (*expressions*)" href="*print.css*">

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

- `@media screen and (min-width: 480px) {  
 body {  
 background-color: lightgreen;  
 }  
}`
- `@media screen and (min-width: 480px) {  
 #leftsidebar {width: 200px; float: left;}  
 #main {margin-left: 216px;}  
}`
- `/* On screens that are 992px or less, set the background  
color to blue */  
@media screen and (max-width: 992px) {  
 body {  
 background-color: blue;  
 font-size: 30px;  
 }  
}`



# References

- [www.w3schools.com](http://www.w3schools.com)
- [www.w3.org](http://www.w3.org)
- *World wide web*