CSS
Cascanding Style Sheet

CSS

- Cascading style sheets (CSS) provides a simple way to style the content on your web pages
- As like as HTML, its pretty easy
- After you write a few lines of HTML page, add a little CSS and you immediately see the results

```
<html>
<head>
  <title>Ex of CSS</title>
</head>
<body>
<p>This is just HTML</p>
</body>
</html>
```

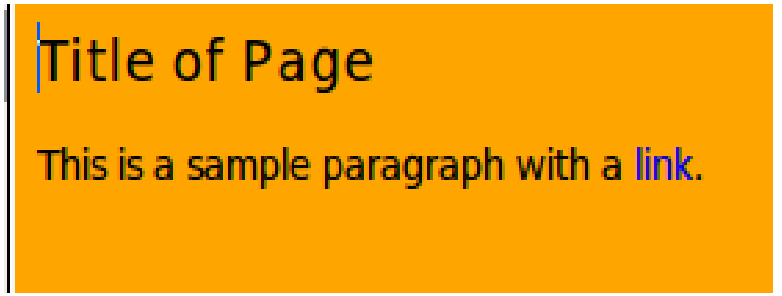
- Then just make simple modify like :

```
<p style="font-family: Tahoma;font-size: 200%">Hello, this is a bit modify of HTML

</p>
```

CSS with HTML

```
<html>
<head>
  <title>Ex of CSS</title>
  <style type="text/css">
  <!--
  body {
   font-family: Tahoma;
  }
  h1 {
   font-size: 120%;
  }
  a {
   text-decoration: none;
  }
  p {
   font-size: 90%;
  }
  -->
  </style>
</head>
<body bgcolor="orange">
<h1>Title of Page</h1>
<p>This is a sample paragraph with a <a href="http://abc.com">link</a>.</p>
</body>
</html>
```



- CSS contain rules with two parts: selectors and properties
- A selector identifies what portion of your web page gets styled. Within a selector are one or more properties and their values
- The property tells the browser what to change and the value lets the browser know what that change should be

Basic of CSS

- Selector Properties Value
- selector { property: value; }
- selector {
  property: value;
  }

- selector {
  property: value;
  property: value, value, value;
  property: value value value value;
  }
- selector, selector {
  property: value;
  }

Using Different Selectors to Apply Styles

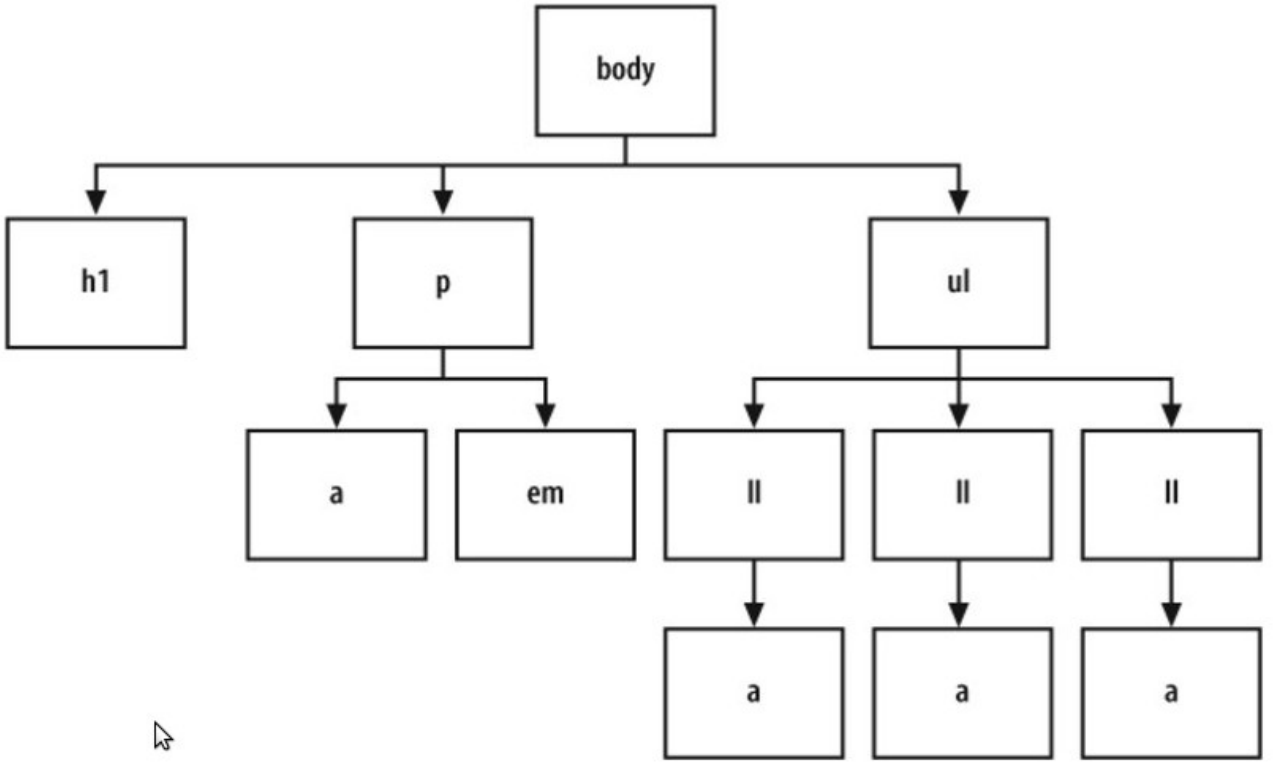
- May be you want to use selectors to apply unique styles to different parts of a web page.
-

```
<html>
<head>
  <title>Ex of CSS</title>
  <style type="text/css">
  <!--
  *{
   font-family: verdana, arial, sans-serif;
  }
   h1 {
   font-size: 120%;
  }
  #navigation {
   border: 1px solid black;
   padding: 40px;
  }
  li a {
   text-decoration: none;
  }
  p {
   font-size: 90%;
  }
  -->
  </style>
</head>
<body style="background-color: orange;">
<h1>Title of Page</h1>
<p>This is a sample paragraph with a <a href="http://abc.com">link</a>.
<em class="warning"></em></p>
<ul id="navigation">
  <li><a href="http://abcd.com">abcd</a></li>
  <li><a href="http://abcde.com">abcde</a></li>
  <li><a href="http://abcdef.com">abcdef</a></li>
</ul>
</body>
</html>
```

Title of Page

This is a sample paragraph with a link.

- abcd
- abcde
- abcdef



Type selectors

- h1 {
  font-size: 120%;
  }
- p {
  color: blue;
  }

Class selectors

- When you want to apply the same CSS rule many times to different elements, use the class selector.
- For example, class selectors can be used to identify warnings with red color in a paragraph, as well as in a list item
- .warning {

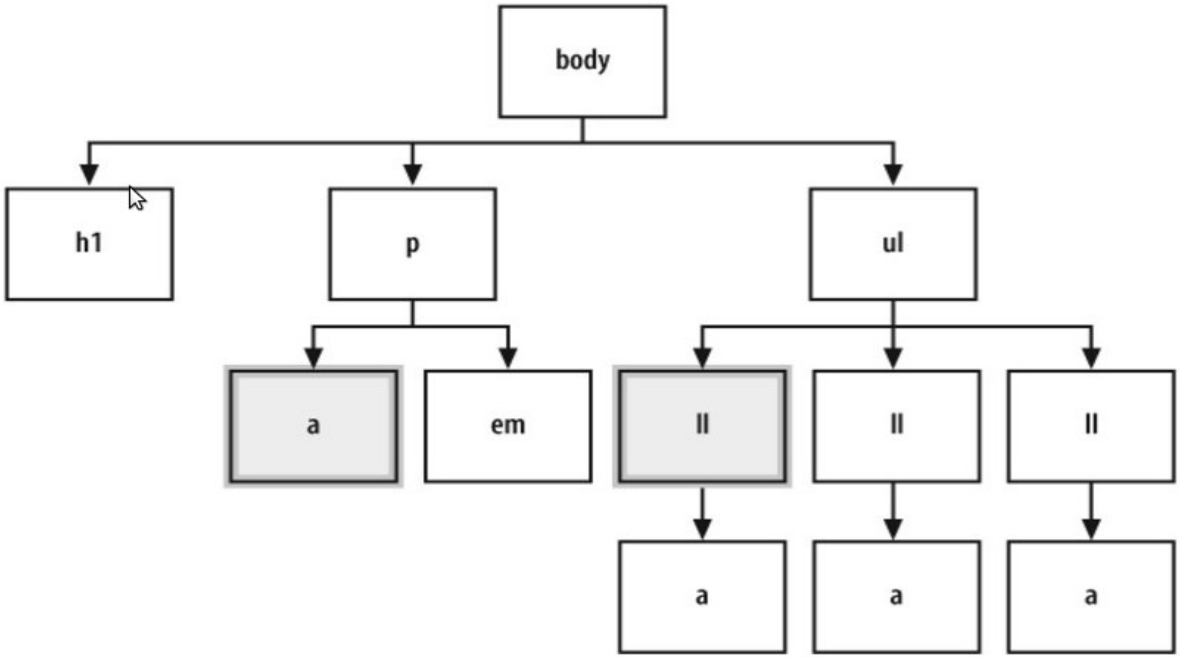
  font-weight: bold;

  }

Title of Page

This is a sample paragraph with a link. ***using class "warning"***

- abcd
- abcde
- abcdef



ID selectors

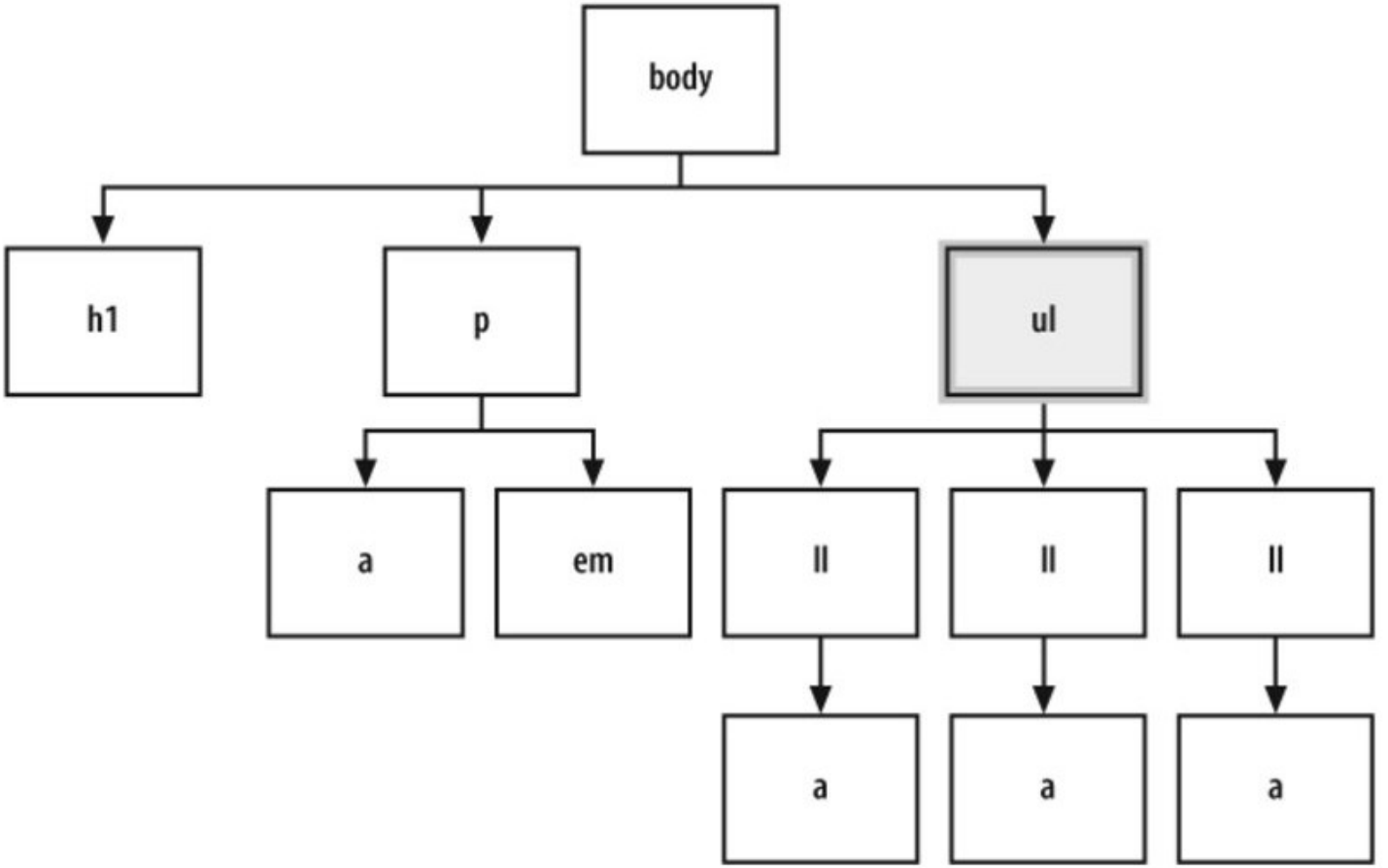
- ID selectors resemble class selectors except that according to the specification they appear only once in the document
- #navigation {

  border: 1px solid black;

  padding: 40px;

  }

```
<ul id="navigation">
 <li class="warning"><a href="http://csscookbook.com">Apples</a></li>
 <li><a href="http://csscookbook.com">Bananas</a></li>
 <li><a href="http://csscookbook.com">Cherries</a></li>
</ul>
```



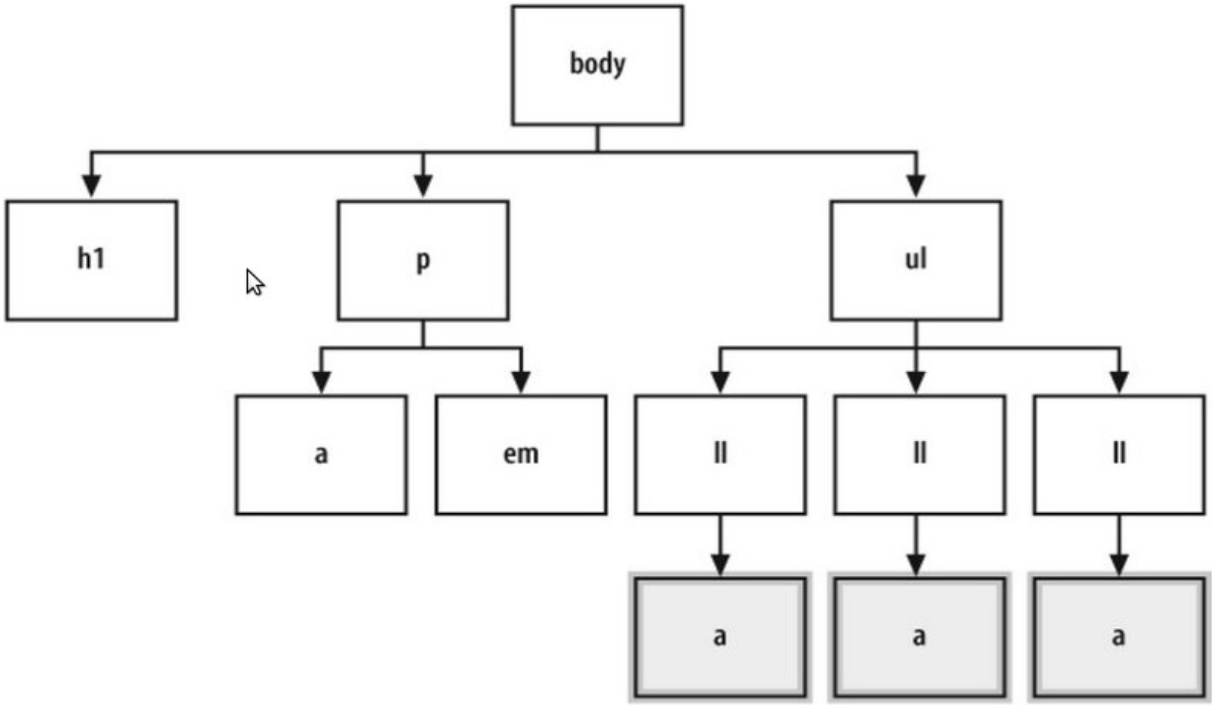
Descendant selectors

- Descendant selectors come next in line and override the type and class selector styles
- They typically have two elements with the second element being a descendant of the first
- li a {

  text-decoration: none;

  }

```
<<ul id="navigation">
 <li class="warning"><a href="http://....">Apples</a></li>
 <li><a href="http://....">Bananas</a></li>
 <li><a href="http://....">Cherries</a></li>
</ul>
```



Child selectors

- A child selector means that an element is styled if it is the direct descendant of its parent element
- A child selector is signified by right-angled bracket often set between two type selectors
- p > strong {

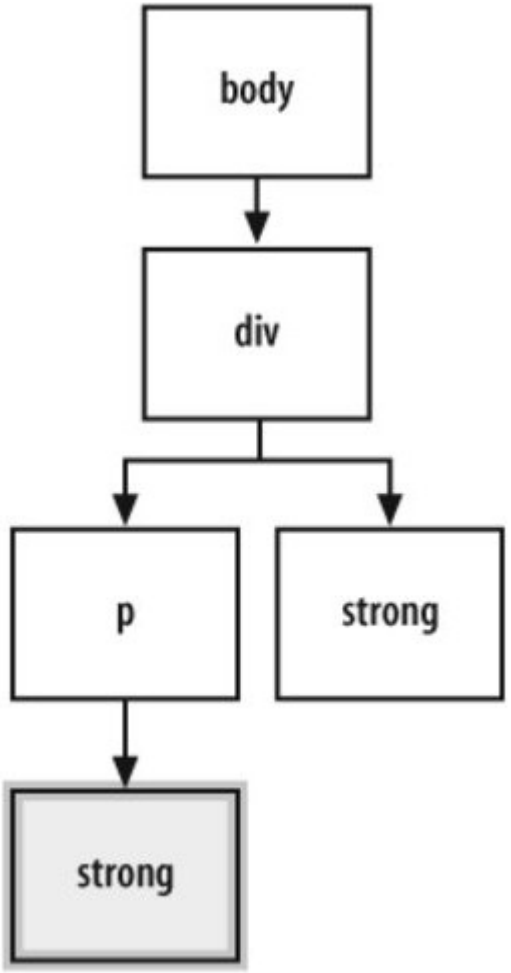
  text-decoration: underline;

  }

```
<div>
 <p>Nothing happens to this part of the sentence because this
<strong>strong</strong> isn't the direct child of div.</p>
 However, this <strong>strong</strong> is the child of div.
Therefore, it receives the style dictated in the CSS rule.
</div>
```

Nothing happens to this part of the sentence because this **strong** isn't the direct child of div.

However, this **strong** is the child of div. Therefore, it receives the div > strong style.

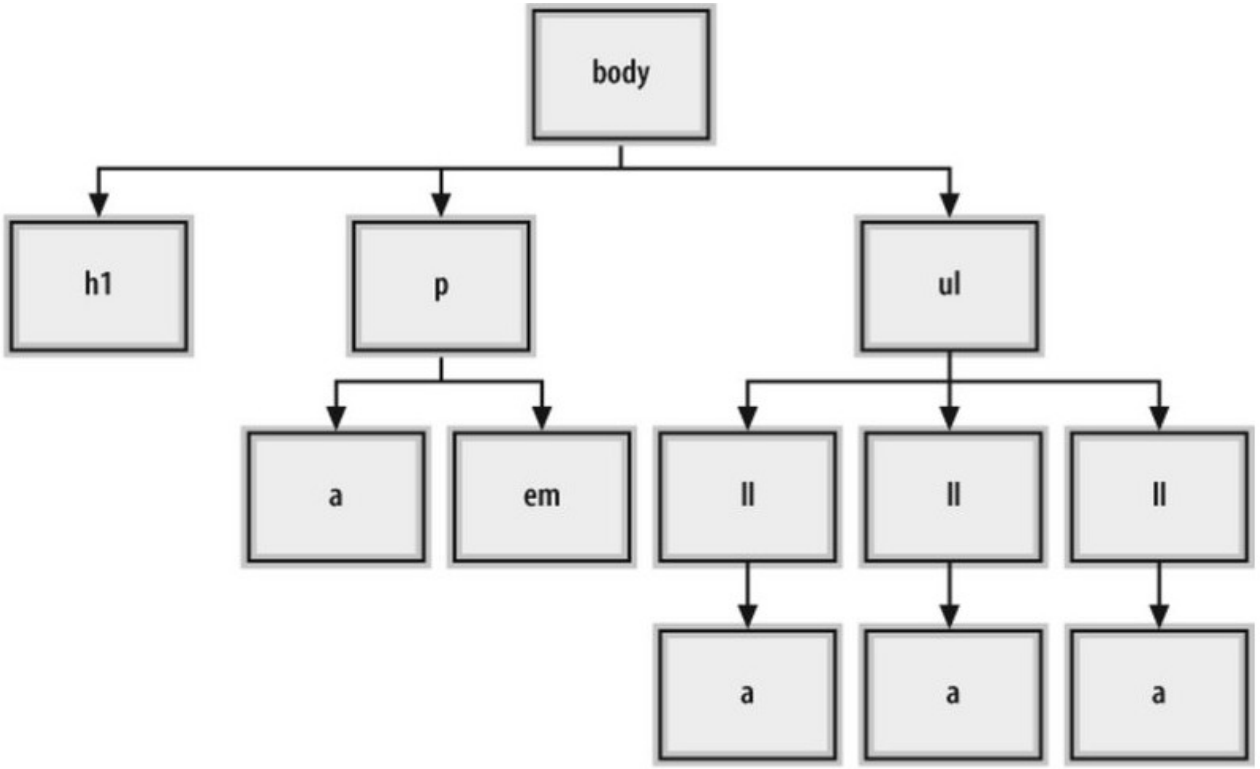

body
div
p
strong
strong

Universal selectors

- Universal selectors are represented with an asterisk (*) and apply to all elements
- * {

  font-family: Verdana, Arial, sans-serif;

  }
- Every element containing HTML text would be styled with a Verdana, Arial, or some other sans-serif font

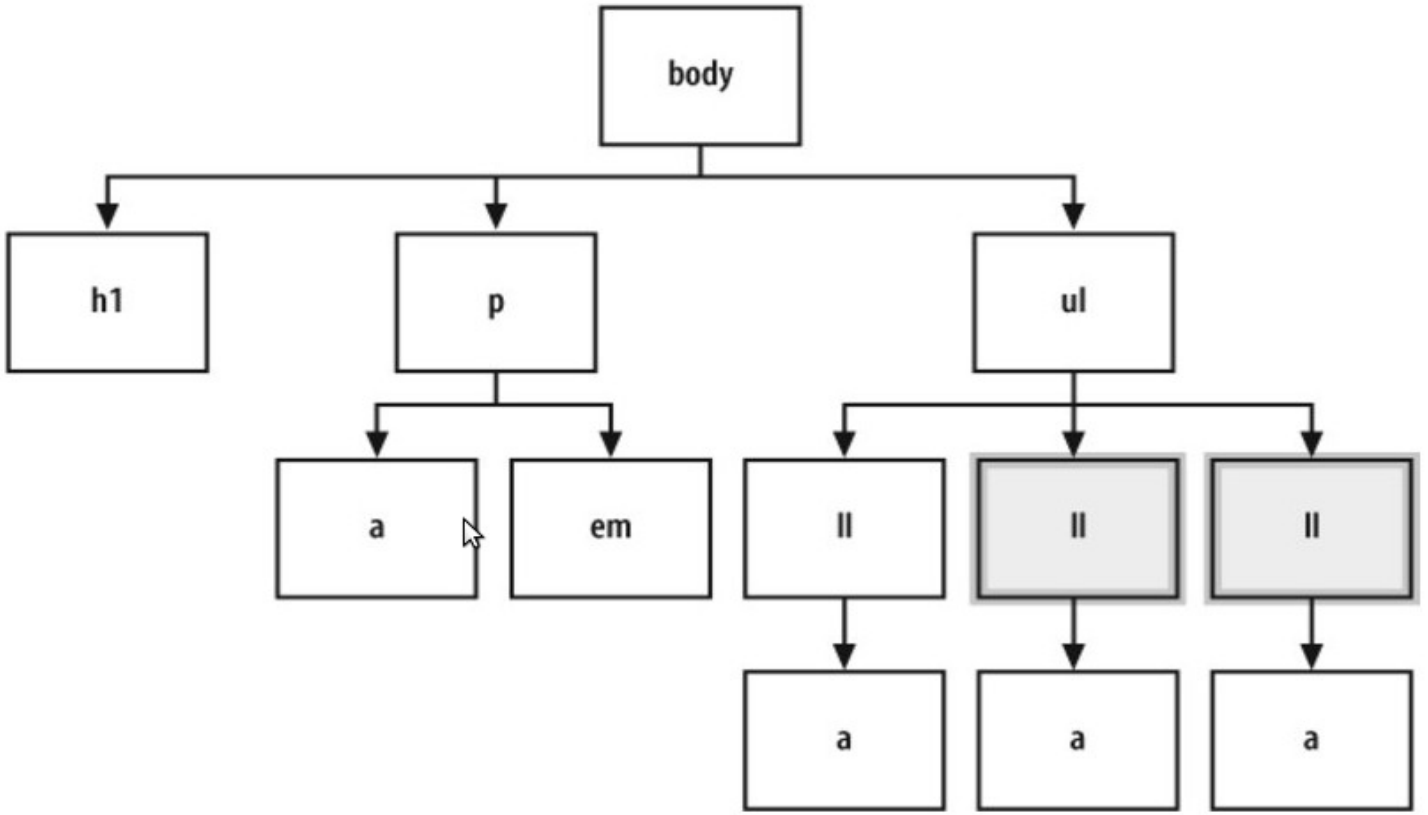

body
h1
p
ul
a
em
li
li
li
a
a
a

Adjacent sibling selectors

- Adjacent siblings describe the relationship between two elements that are placed side-by-side within the flow of a web page's markup
- li + li {

  font-size: 200%;

  }


body
h1
p
ul
a
em
li
li
li
a
a
a

Attribute selectors

- Attribute selectors have four ways to find an element that has a matching attribute
- [attribute] - Search for matches based on the attribute.

  a[href] {

  text-decoration: none;

  }
- Whenever href attribute appears within an a element in the HTML, the link won't have an underline (because the default the link always use underline)

- [attribute=val] - Search for matches based on the value.

  a[href="abc.com"] {

  text-decoration: none;

  }

- Whenever a link that points to abc.com appears in the HTML, the link won't have an underline.

- [attribute~=val] - Search for matches that contain the space-separated attribute somewhere in the value.

  a[title~="uns"] {

  text-decoration: none;

  }

- Whenever "uns" appears in the title attribute of an anchor element, the link won't have an underline

- [attribute|=val] - Search for matches that contain the attribute with a hyphen.

  a[href|="informatcs"] {

  text-decoration: none;

  }

- Also, whenever "informatics-" appears in the href attribute of an anchor element, the link won't have an underline.

Pseudo-classes

- You may want to add style to items that aren't based on elements' name, attributes, or content

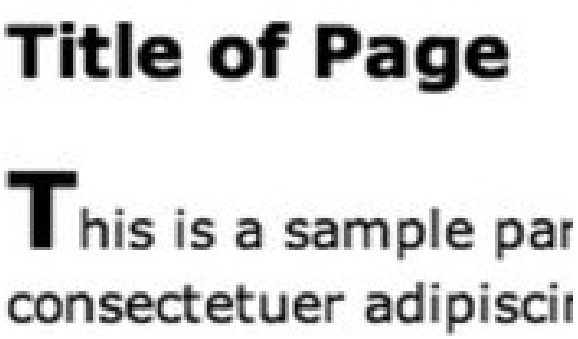
- ```
  a:link {
  color: blue;
  }
  a:visited {
  color: purple;
  }
  a:hover {
  color: red;
  }
  a:active {
  color: gray;
  }
  ```

In this setup, a basic link appears in blue. As soon as the mouse pointer hovers over the link, it changes to red. During the clicking of the link, the link appears gray. When returning to the page with the link after visiting, the link appears purple.
  ```

Pseudo-elements

- With most selectors, a developer makes use of elements and their arrangement within a web document to style a document
- However, sometimes a developer can style an item within a web document that's not marked up by elements through the use of pseudo-elements
- Pseudo-elements consist of :first-letter, :first-line, :before, and :after.

- p:first-letter {
  font-size: 200%;
  font-weight: bold;
  }
-



- p:first-letter {
  first-line: 200%;
  font-weight: bold;
  }

- :active
- This applies to an element during the period in which it is activated
- The most common example of this is clicking on a hyperlink in an HTML document: during the time that the mouse button is held down, the link is active
- a:active {color: red;}

  *:active {background: blue;}

- :after
- This allows the author to insert generated content at the end of an element's content
- a.external:after {content: " " url(/icons/globe.gif);}

  p:after {content: " | ";}

- :before
- This allows the author to insert generated content at the beginning of an element's content
- a[href]:before {content: "[LINK] ";}

  p:before {content: attr(class);}

- :first-child
- With this pseudo-class, an element is matched only when it is the first child of another element
- body *:first-child {font-weight: bold;}

  p:first-child {font-size: 125%;}

- :first-letter
- This is used to style the first letter of an element
- h1:first-letter {font-size: 166%;}

  a:first-letter {text-decoration: underline;}

- :first-line
- This is used to style the first line of text in an element, no matter how many or how few words may appear in that line
- p.lead:first-line {font-weight: bold;}

- :focus
- This applies to an element during the period in which it has focus
- One example from HTML is an input box that has the text-input cursor within it
- a:focus {outline: 1px dotted red;}

  input:focus {background: yellow;}

- :hover
- This applies to an element during the period in which it is "hovered"
- Hovering is defined as the user designating an element without activating it
- a[href]:hover {text-decoration: underline;}

  p:hover {background: yellow;}

- :lang
- This matches elements based on their human language encoding
- Such language information must be contained within or otherwise associated with the document; it cannot be assigned from CSS
- html:lang(en) {background: silver;}

  *:lang(fr) {quotes: '« ' ' »';}

- :link
- This applies to a link to a URI that has not been visited
- a:link {color: blue;}

  *:link {text-decoration: underline;}

- :visited
- This applies to a link to a URI that has been visited
- a:visited {color: purple;}

  *:visited {color: gray;}

Saving Time with Inheritance

- Inheritance is the process by which CSS properties applied to one tag are passed on to nested tags
- For example, a <p> tag is always nested inside of the <body> tag, so properties applied to the <body> tag get inherited by the <p> tag

Determining When to Use Class and ID Selectors

- You want to determine the best use for class and ID selectors.
- Use class selectors when you need to apply a style multiple times within a document and ID selectors for one-time only appearances of a style within a document.
- (#)...., ID selector
- (.)... , class

```
<!--
body {
margin: 0;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: .75em;
padding: 0;
}
#banner {
margin-top: 0;
margin-bottom: 0;
background-color: #900;
border-bottom: solid 1px #000;
padding: 5px 5px 5px 10px;
line-height: 75%;
color: #fff;
}
#sub_banner {
background-color: #ccc;
border-bottom: solid 1px #999;
font-size: .8em;
font-style: italic;
padding: 3px 0 3px 10px;
}
#content {
position: absolute;
margin-left: 18%;
width: 40%;
top: 100px;
padding: 5px;
```

```
#nav1 {
position: absolute;
width: 30%;
left: 60%;
top: 100px;
padding: 5px;
}
#nav2 {
position: absolute;
padding: 5px 5px 5px 10px;
top: 100px;
width: 15%;
}
#footer {
text-align: center;
padding-top: 7em;
}
.warning {
font-weight: bold;
color: red;
}
.title {
font-size: 120%;
}
.content {
font-family: Verdana, Arial, sans-serif;
margin-left: 20px;
margin-right: 20px;
}
.footer {
font-size: 75%;
}
```

- And apply to the HTML

```
<body style="background-color: orange;">
<div id="header">
<h1>Ex of CSS</h1>
<h2>Showcase of CSS Web Sites</h2>
</div>
<div id="content">
<h3>Page Title</h3>
<p class="title">Content Item Title</p>
<p class="content">Content goes here.</p>
</div>
<div id="navigation">
<h3>List</h3>
<a href="http://a.com/">Submit a site</a><br>
<a href="http://b.com/">CSS resources</a><br>
<a href="http://c.com/">RSS</a><br>
<h3>CSS Cookbook Stuff</h3>
<a href="http://d.com/">Home</a><br>
<a href="http://e.com/">About</a><br>
<a href="http://f.com/">Blog</a><br>
<a href="http://g.com/">Services</a><br>
</div>
<div id="blipverts">
<h3>Ads go here.</h3>
</div>
<div id="siteinfo">
<p class="footer">Copyright 2006</p>
</div>
</body>
```


Ex of CSS
Showcase of CSS Web Sites
List
Submit a site
CSS resources
RSS
CSS Content
Home
About
Blog
Services
Ads go here.
Copyright 2010
Page Title
Content Item Title
Content goes here.

- The ID selectors identify unique attributes that have one instance in the document tree, whereas class selectors can be used frequently throughout the web page
- Typically, web developers will use ID selectors to mark off unique sections of a web page
- Notice that the page is divided into the following sections: header, content, navigation, blipverts, siteinfo

Understanding CSS Properties

- To learn more about CSS properties
- Properties fall between the brackets and their values
- selector {

  property: value;

  }

- li {

  list-style-type: square;

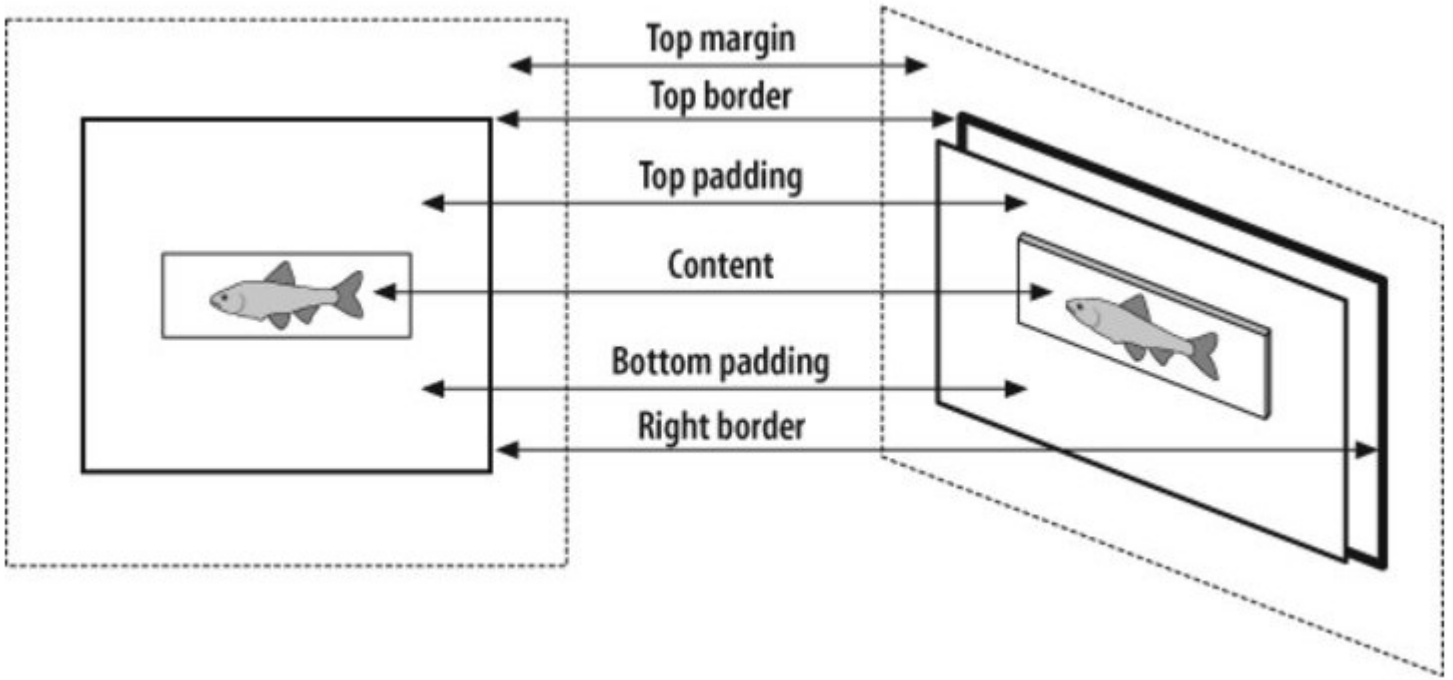
  }
- Any time li appears in the document, the bullet appears as a square rather than a traditional bullet
- Selectors identify what should be styled within a web document, while properties and selectors identify the what and how that portion of the web document should be modified

- For example, the color property means the element's color will change, but not what color

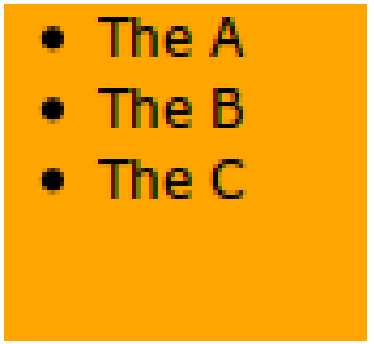
Property	Value
font-weight	bold
border-color	Color name or color hexadecimal HTML value (e.g., #000000 for black and #ffffff for white)
border-style	solid dotted dashed double
text-align	left center right justify

Understanding the Box Model

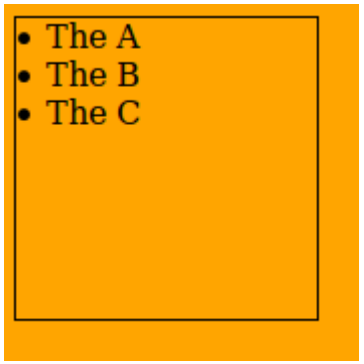
- To better understand the box model and how margins, borders, and padding work around content.
- Every block level element, like a p or div element, contains a top, right, bottom, and left edge
- These sides of block elements are composed of three layers surrounding the content

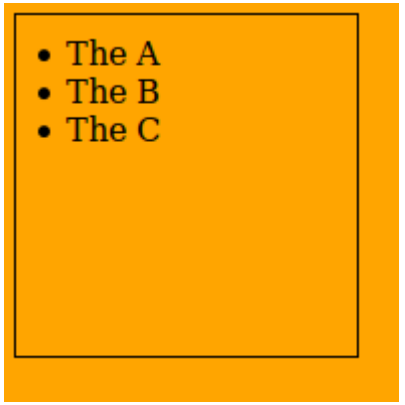

Top margin
Top border
Top padding
Content
Bottom padding
Right border

- div {
  height: 150px;
  width: 150px;
  }



- Add border: thin solid #000000;
  then u can see a border



- 

and padding:10px

- div {

  border: 5px double #000000;

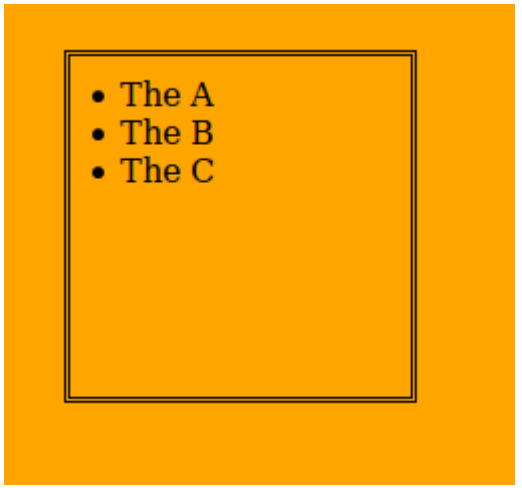
  height: 150px;

  width: 150px;

  padding: 10px;

  margin: 25px;

  }



Associating Styles to a Web Page

- To know about the different ways of adding styles to a web page
- You can apply styles in three ways: external, internal, and inline
- A unique web page may have its own style sheet so styles only affect the page and not all web pages. Define internal styles within the style tags

- <style>
  <!--
  ....
  -->
  </style>

- External style sheets are stored in a separate file, which become associated with the HTML file through linking
- All web pages link to the external style sheet that contains nothing but CSS styles. If you want to change the font color on all pages linked to this style sheet, just update the external style sheet. Link to the style sheet with the link tag.

- In the web page, add the following line between the head tags to link to the external style sheet
- <link href="screen.css" rel="stylesheet" type="text/css" />
- Inline styles work similarly to font with the style information applied to a specific tag within a web page. Designers rarely apply inline styles

How to Use Different Types of Style Sheets

- To provide style sheets for different media types
- Ex, make 3 external css such as print.css, screen.css and projection.css
- Then link all as different media, ex below:
- <link rel="stylesheet" type="text/css" href="/css/print.css" media="print" />
- <link rel="stylesheet" type="text/css" href="/css/screen.css" media="screen" />
- <link rel="stylesheet" type="text/css" href="/css/projection.css" media="projection" />

- Then how to use, you could use the @media rule instead to specific the different media rules within the same style sheet
-

```
<style type="text/css">
<!--
@media one {
 body { font: 10pt times, georgia, serif }
}

@media two {
 body { font: 12pt verdana, arial, sans-serif}
}

@media three {
 body { font-size: 14pt }
}

@media one, two, three {
 body { line-height: 120% }
}
-->
</style>
```

Organizing the Contents of a Style Sheet

- To know how effectively to organize contents within a style sheet for easier management
- Managing CSS can be accomplished by grouping common visual elements of a web page together
  - Elements (h1h6, p, a, list, links, images)
  - Typography
  - Page layout (header, content, navigation, global navigation, subnavigation, sidebar, footer)
  - Form tags (form, fieldset, label, legend)
  - Content (post, events, news)

- Manage CSS files by placing them in their own directory
  - /.../css/one.css
  - /.../css/two.css
- Place all css in one directory or hierarchy directory, it make easier to remember and manage

Absolute and Relative Positioning

- Use the position property
- Absolute → exact, relative → follow the flow

```
.absolute {
 position: absolute;
 bottom: 50px;
 left: 100px;
}
```

```
.relative {
 position: relative;
 top: 100px;
 left: 20px;
}
```

shfshdfbndbvnbxvbdh vjxvjxjbvjxcbjbxbjxcvvbbvchvb xnvxhjvhsdjkvhjxhv.....

shfshdfbndbvnbxvbdh vjxvjxjbvjxcbjbxbjxcvvbbvc

Name	Values	Initial value	Applies to(Default: all)	Inherited?	Percentages(Default: N/A)	Media groups
'background-attachment'	scroll | fixed | inherit	scroll		no		visual
'background-color'	<color> | transparent | inherit	transparent		no		visual
'background-image'	<uri> | none | inherit	none		no		visual
'background-position'	[[<percentage> | <length> | left | center | right] [<percentage> | <length> | top | center | bottom]?] | [[left | center | right] || [top | center | bottom]] | inherit	0% 0%		no	refer to the size of the box itself	visual
'background-repeat'	repeat | repeat-x | repeat-y | no-repeat | inherit	repeat		no		visual
'background'	['background-color' || 'background-image' || 'background-repeat' || 'background-attachment' || 'background-position'] | inherit	see individual properties		no	allowed on 'background-position'	visual
'border-collapse'	collapse | separate | inherit	separate	'table' and 'inline-table' elements	yes		visual
'border-color'	[<color> | transparent]{1,4} | inherit	see individual properties		no		visual
'border-spacing'	<length> <length>? | inherit	0	'table' and 'inline-table' elements	yes		visual
'border-style'	<border-style>{1,4} | inherit	see individual properties		no		visual

'border-top' 'border-right' 'border-bottom' 'border-left'	[<border-width> || <border-style> || >'border-top-color'] | inherit	see individual properties		no		visual
'border-top-color' 'border-right-color' 'border-bottom-color' 'border-left-color'	<color> | transparent | inherit	the value of the 'color' property		no		visual
'border-top-style' 'border-right-style' 'border-bottom-style' 'border-left-style'	<border-style> | inherit	none		no		visual
'border-top-width' 'border-right-width' 'border-bottom-width' 'border-left-width'	<border-width> | inherit	medium		no		visual
'border-width'	<border-width>{1,4} | inherit	see individual properties		no		visual
'border'	[<border-width> || <border-style> || >'border-top-color'] | inherit	see individual properties		no		visual
'bottom'	<length> | <percentage> | auto | inherit	auto	positioned elements	no	refer to height of containing block	visual
'caption-side'	top | bottom | inherit	top	'table-caption' elements	yes		visual
'clear'	none | left | right | both | inherit	none	block-level elements	no		visual

'clip'	<shape> | auto | inherit	auto	absolutely positioned elements	no		visual
'color'	<color> | inherit	depends on user agent		yes		visual
'content'	normal | none | [<string> | <uri> | <counter> | attr(<identifier>) | open-quote | close-quote | no-open-quote | no-close-quote]+ | inherit	normal	:before and :after pseudo-elements	no		all
'counter-increment'	[<identifier> <integer>?]+ | none | inherit	none		no		all
'counter-reset'	[<identifier> <integer>?]+ | none | inherit	none		no		all
'cursor'	[[<uri> ,]* [auto | crosshair | default | pointer | move | e-resize | ne-resize | nw-resize | n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help | progress]] | inherit	auto		yes		visual, interactive
'direction'	ltr | rtl | inherit	ltr	all elements, but see prose	yes		visual
'display'	inline | block | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none | inherit	inline		no		all
'empty-cells'	show | hide | inherit	show	'table-cell' elements	yes		visual

'float'	left | right | none | inherit	none	all, but see 9.7	no		visual
'font-family'	[[<family-name> | <generic-family>] [, <family-name>| <generic-family>]*] | inherit	depends on user agent		yes		visual
'font-size'	<absolute-size> | <relative-size> | <length> | <percentage> | inherit	medium		yes	refer to parent element's font size	visual
'font-style'	normal | italic | oblique | inherit	normal		yes		visual
'font-variant'	normal | small-caps | inherit	normal		yes		visual
'font-weight'	normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit	normal		yes		visual
'font'	[['font-style' || 'font-variant' || 'font-weight']? 'font-size' [/ 'line-height']? 'font-family'] | caption | icon | menu | message-box | small-caption | status-bar | inherit	see individual properties		yes	see individual properties	visual
'height'	<length> | <percentage> | auto | inherit	auto	all elements but non-replaced inline elements, table columns, and column groups	no	Allowed; percentage is calculated with respect to the height of the generated box's containing block. If the height of the containing block is not specified explicitly (i.e., it depends on the content height), the value is interpreted like "auto."	visual
'left'	<length> | <percentage> | auto | inherit	auto	positioned elements	no	refer to width of containing block	visual
'letter-spacing'	normal | <length> | inherit	normal		yes		visual
'line-height'	normal | <number> | <length> | <percentage> | inherit	normal		yes	refer to the font size of the element itself	visual

'list-style-image'	<uri> | none | inherit	none	elements with 'display: list-item'	yes		visual
'list-style-position'	inside | outside | inherit	outside	elements with 'display: list-item'	yes		visual
'list-style-type'	disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian | georgian | lower-alpha | upper-alpha | none | inherit	disc	elements with 'display: list-item'	yes		visual
'list-style'	['list-style-type' || 'list-style-position' || 'list-style-image'] | inherit	see individual properties	elements with 'display: list-item'	yes		visual
'margin-right' 'margin-left'	<margin-width> | inherit	0	all elements except elements with table display types other than table and inline-table	no	refer to width of containing block	visual
'margin-top' 'margin-bottom'	<margin-width> | inherit	0	all elements except elements with table display types other than table and inline-table	no	refer to width of containing block	visual
'margin'	<margin-width>{1,4} | inherit	see individual properties	all elements except elements with table display types other than table and inline-table	no	refer to width of containing block	visual

'max-height'	<length> | <percentage> | none | inherit	none	all elements but non-replaced inline elements, table columns, and column groups	no	Allowed; percentage is calculated with respect to the height of the generated box's containing block. If the height of the containing block is not specified explicitly (i.e., it depends on the content height), the value is interpreted like "auto."	visual
'max-width'	<length> | <percentage> | none | inherit	none	all elements but non-replaced inline elements, table rows, and row groups	no	refer to width of containing block	visual
'min-height'	<length> | <percentage> | inherit	0	all elements but non-replaced inline elements, table columns, and column groups	no	Allowed; percentage is calculated with respect to the height of the generated box's containing block. If the height of the containing block is not specified explicitly (i.e., it depends on the content height), the value is interpreted like "auto." Allowed; percentage is calculated with respect to the height of the generated box's containing block. If the height of the containing block is not specified explicitly (i.e., it depends on the content height), the value is interpreted like "auto."	visual
'min-width'	<length> | <percentage> | inherit	0	all elements but non-replaced inline elements, table rows, and row groups	no	refer to width of containing block	visual
'orphans'	<integer> | inherit	2	block-level elements	yes		visual, paged
'outline-color'	<color> | invert | inherit	invert		no		visual, interactive
'outline-style'	<border-style> | inherit	none		no		visual, interactive
'outline-width'	<border-width> | inherit	medium		no		visual, interactive

'outline'	['outline-color' || 'outline-style' || 'outline-width'] | inherit	see individual properties		no		visual, interactive
'overflow'	visible | hidden | scroll | auto | inherit	visible	non-replaced block-level elements, table cells, and inline-block elements	no		visual
'padding-top' 'padding-right' 'padding-bottom' 'padding-left'	<padding-width> | inherit	0	all elements except elements with table display types other than table, inline-table, and table-cell	no	refer to width of containing block	visual
'padding'	<padding-width>{1,4} | inherit	see individual properties	all elements except elements with table display types other than table, inline-table, and table-cell	no	refer to width of containing block	visual
'page-break-after'	auto | always | avoid | left | right | inherit	auto	block-level elements	no		visual, paged
'page-break-before'	auto | always | avoid | left | right | inherit	auto	block-level elements	no		visual, paged
'page-break-inside'	avoid | auto | inherit	auto	block-level elements	yes		visual, paged
'position'	static | relative | absolute | fixed | inherit	static		no		visual
'quotes'	[<string> <string>]+ | none | inherit	depends on user agent		yes		visual
'right'	<length> | <percentage> | auto | inherit	auto	positioned elements	no	refer to width of containing block	visual

'table-layout'	auto | fixed | inherit	auto	'table' and 'inline-table' elements	no		visual
'text-align'	left | right | center | justify | inherit	'left' if 'direction' is 'ltr'; 'right' if 'direction' is 'rtl'	block-level elements, table cells and inline blocks	yes		visual
'text-decoration'	none | [underline || overline || line-through || blink] | inherit	none		Allowed; percentage is calculated with respect to the height of the generated box's containing block. If the height of the containing block is not specified explicitly (i.e., it depends on the content height), the value is interpreted like "auto."		visual
'text-indent'	<length> | <percentage> | inherit	0	block-level elements, table cells and inline blocks	yes	refer to width of containing block	visual
'text-transform'	capitalize | uppercase | lowercase | none | inherit	none		yes		visual
'top'	<length> | <percentage> | auto | inherit	auto	positioned elements	no	refer to height of containing block	visual
'unicode-bidi'	normal | embed | bidi-override | inherit	normal	all elements, but see prose	no		visual
'vertical-align'	baseline | sub | super | top | text-top | middle | bottom | text-bottom | <percentage> | <length> | inherit	baseline	inline-level and 'table-cell' elements	no	refer to the 'line-height' of the element itself	visual
'visibility'	visible | hidden | collapse | inherit	visible		yes		visual

'white-space'	normal | pre | nowrap | pre-wrap | pre-line | inherit	normal		yes		visual
'widows'	<integer> | inherit	2	block-level elements	yes		visual, paged
'width'	<length> | <percentage> | auto | inherit	auto	all elements but non-replaced inline elements, table rows, and row groups	no	refer to width of containing block	visual
'word-spacing'	normal | <length> | inherit	normal		yes		visual
'z-index'	auto | <integer> | inherit	auto	positioned elements	no		Visual

Ext

'background-position-x'	<length> | <percentage> | left | center | right	0%		no	yes	visual
'background-position-y'	<length> | <percentage> | top | center | bottom	0%		no	yes	visual
'filter'	See http://tinyurl.com/c8vpf	n/a		no		filter properties
'ime-mode'	auto | active | inactive | disabled	auto		yes		visual
'layout-grid'	mode | type | line | char	both loose none none		yes		visual
'layout-grid-char'	<length> | <percentage> | none | auto	none		no	yes	visual
'layout-grid-line'	<length> | <percentage> | none | auto	none		no	yes	visual
'layout-grid-mode'	both | none | line | char	both		yes		visual
'layout-grid-type'	loose | strict | fixed	loose		yes		visual
'line-break'	normal | strict	normal		yes		visual
'overflow-x'	visible | scroll | hidden | auto	visible (except for textarea, then initial value is hidden)		no		visual
'overflow-y'	visible | scroll | hidden | auto	visible (except for textarea, then initial value is auto)				visual
'scrollbar-3dlight-color'	<color>	default color	element with scroll bar	yes		visual
'scrollbar-arrow-color'	<color>	default color	element with scroll bar	yes		visual
'scrollbar-base-color'	<color>	default color	element with scroll bar	yes		visual
'scrollbar-darkshadow-color'	<color>	default color	element with scroll bar	yes		visual
'scrollbar-face-color'	<color>	default color	element with scroll bar	yes		visual
'scrollbar-highlight-color'	<color>	default color	element with scroll bar	yes		visual

'scrollbar-shadow-color'	<color>	default color	element with scroll bar	yes		visual
'text-autospace'	none | ideograph-alpha | ideograph-numeric | ideograph-parenthesis | ideograph-space	none		no		visual
'text-justify'	auto | distribute | distribute-all-lines | distribute-center-last | inter-cluster | inter-ideograph | inter-word | kashida | newspaper	auto		yes		visual
'text-kashida-space'	<percentage> | inherit	0%		yes		visual
'text-underline-position'	above | below | auto | auto-pos	auto		yes		visual
'word-break'	normal | break-all | keep-all	normal		yes		visual
'word-wrap'	normal | break-word	normal		yes		visual
'writing-mode'	lr-tb | tb-rl	lr-tb		no		visual
'zoom'	normal | <number> | <percentage>	normal		no	yes	visual