

Internet Programming

Cascading Style Sheets

Dan Brandon, Ph.D., PMP

Separation of Style from Content

- Cascading Style Sheets (CSS) allows the content (text, images, etc.) and structure (headers, links, etc.) to be separated from the style (spacing, fonts, colors, etc.)
- This makes changing the style easier
- More importantly it makes applying a consistent style to multiple pages easier

CSS Standards

- CSS1 is the “original standard” in 1996
- CSS2 was the next release
- CSS3 is the current standard
- CSS continues to evolve
- Compatibility charts available online →



```
body {  
  font: x-small  
  background: #fff;  
  color: black;  
  margin: 0;  
  padding: 0;
```

Backgrounds and borders

[show page contents](#)

The backgrounds and borders.

This is the desktop table. See also the [mobile table](#).

Last major update on 27 February 2013.

[Back to the index.](#)

[CSS3 backgrounds and borders module.](#)

Backgrounds

Contents of this table



See also the [key](#) to my compatibility tables.

Selector	IE7 and lower	IE8	IE9	IE10	FF 15 Win	FF 15 Mac	Safari 5.0.2 Mac	Chrome 25 Win	Chrome 25 Mac	Yandex 4.5 Mac	Opera 12.14 Win	Opera 12.14 Mac
background-attachment States whether the background image scrolls with the document, the element, or not at all.	incorrect root	incomplete	yes		incomplete		yes		yes		yes	
Incomplete browser does not support local. Incorrect scroll means local.												
background-clip States whether the background covers the border-box (default), padding-box or content-box.	no		yes		yes		yes		yes		yes	
background-color		yes			yes		yes		yes		yes	
background-image	incomplete		yes		yes		yes		yes		yes	
• IE8 and lower do not support multiple background images.												
background-origin States whether the background originates in the border-box, padding-box (default) or content-box.	no		yes		yes		yes		yes		yes	
background-position States how the background image is positioned.	almost		yes		yes		yes		yes		yes	
IE7-9 and 6 not tested. • IE7 accepts the top: 75px, which should not work.												
background-repeat: no-repeat, repeat-x, repeat-y States how the background is repeated. Note the new values space and round.	yes		yes		yes		yes		yes		yes	
background-repeat: space, round States how the background is repeated. Note the new values space and round.	no		yes		no		buggy		buggy		yes	
Buggy Browser treats space and round as no-repeat.												
background-size States the size of the background image, in pixels, percenta, or with two special values.	no		yes		yes		yes		yes		yes	
Selector	IE7 and lower	IE8	IE9	IE10	FF 15 Win	FF 15 Mac	Safari 5.0.2 Mac	Chrome 25 Win	Chrome 25 Mac	Yandex 4.5 Mac	Opera 12.14 Win	Opera 12.14 Mac

Other

Contents of this table

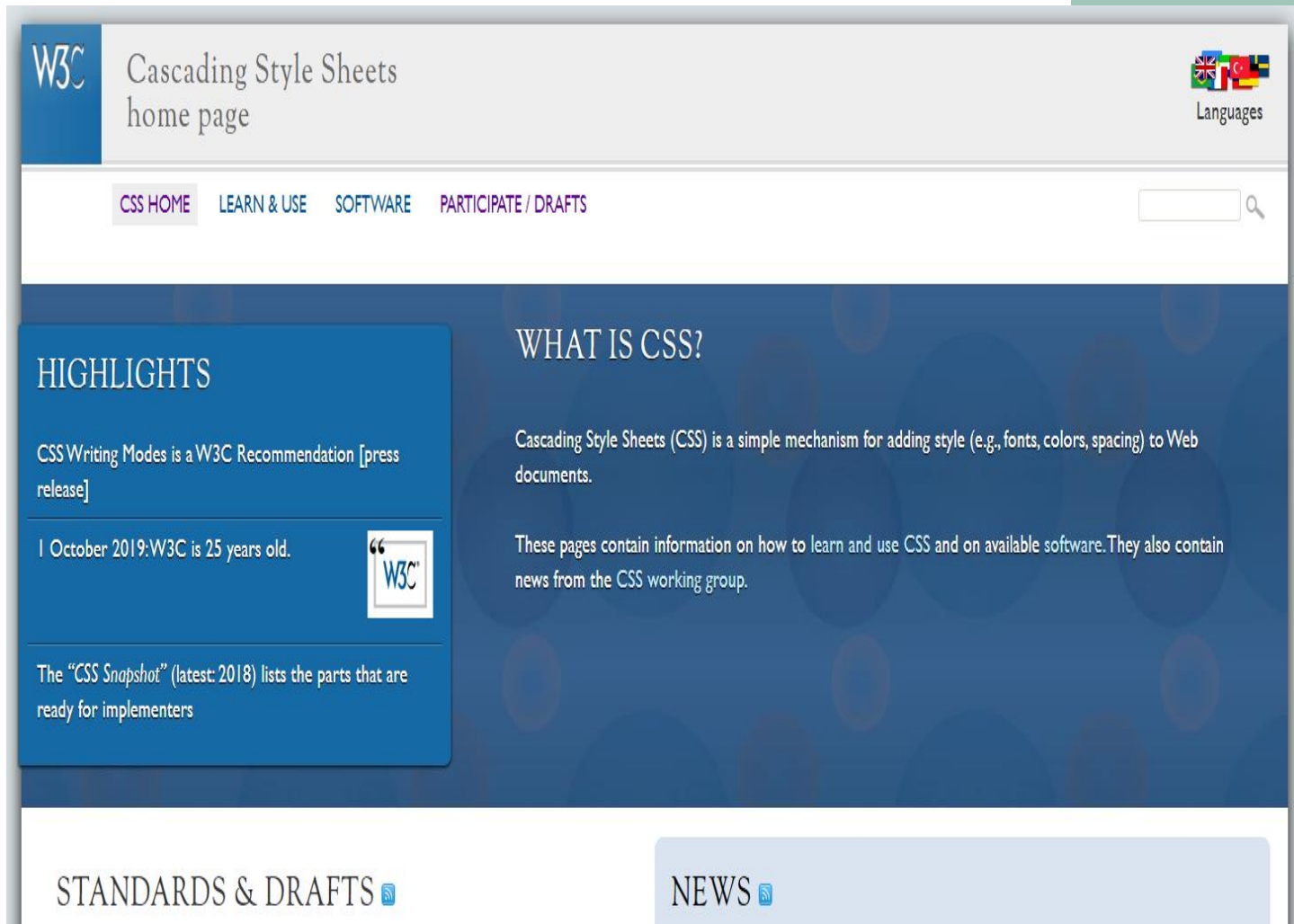


See also the [key](#) to my compatibility tables.

Selector	IE7 and lower	IE8	IE9	IE10	FF 15 Win	FF 15 Mac	Safari 5.0.2 Mac	Chrome 25 Win	Chrome 25 Mac	Yandex 4.5 Mac	Opera 12.14 Win	Opera 12.14 Mac
border-radius	no		yes		yes		yes		yes		yes	
box-decoration-break		no			no		no		~webkit-		incomplete	

W3C CSS Home

[<http://www.w3.org/Style/CSS/>]



Completed work	Status	Upcoming	
CSS Snapshot 2018	NOTE		
CSS Snapshot 2017	NOTE		
CSS Snapshot 2015	NOTE		
CSS Snapshot 2010	NOTE		
CSS Snapshot 2007	NOTE		
CSS Color Level 3	REC	REC	
CSS Namespaces	REC	REC	
Selectors Level 3	REC	REC	
CSS Level 2 Revision 1	REC	REC	
Media Queries	REC	REC	
CSS Style Attributes	REC	REC	
CSS Fonts Level 3	REC	REC	
CSS Writing Modes Level 3	REC	REC	
CSS Basic User Interface Level 3	REC	REC	
CSS Containment Level 1	REC	REC	

Stable drafts	Status	Upcoming	
CSS Backgrounds and Borders Level 3	CR	PR	
CSS Conditional Rules Level 3	CR	CR	
CSS Multi-column Layout Level 1	WD	CR	
CSS Values and Units Level 3	CR	PR	
CSS Flexible Box Layout Level 1	CR	PR	
CSS Cascading and Inheritance Level 3	CR	PR	
CSS Counter Styles Level 3	CR	PR	

[More »](#)

Abbreviation	Full name
FPWD	First Public Working Draft
WD	Working Draft
CR	Candidate Recommendation
PR	Proposed Recommendation
REC	Recommendation
SPSD	Superseded Recommendation

CSS Rules

- Each CSS rule consists of a **selector** and **declarations**
- The declarations have a property and a value, separated by a **colon**
 - selector {property:value; property:value}
- **body {color:red}**
 - The selector is body
 - The declaration property is color
 - The declaration value is red
- Multiple declarations for a selector are separated by **semicolons**

Internal and External Style Sheets

- Style specifications can be internal to an HTML document (“internal”) or referenced via a URL link (“external”)
- The “STYLE” tag can be used to define style specifications either as a separate tag or as an attribute of another tag
- As an attribute of another tag, it is called an **“inline” style**:
 - `<p style="font-size: 24pt; color: red">...text...</p>`
 - Note ; separated list of declarations inside of quotes

Inline Style

```
<HTML>
<HEAD><TITLE>Inline Style</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
// -->
</SCRIPT>
</HEAD>
<BODY>
Hello World !
<P STYLE="Font-size: 18pt; color: red">
Hello World !
<P STYLE="Font-size: 24pt; color: blue">
Hello World !
</BODY></HTML>
```



Hello World !

Hello World !

Hello World !

Font Size Specification

- Specify font size in either an absolute or relative mode:
 - Absolute:
 - font-size: 18pt (can also use: px, in, cm, %, em)
 - Note: em is a default font height size, ex is default font x width size
 - Relative:
 - font-size: xx-small
 - Values: xx-small, x-small, small, smaller, medium, large, larger, x-large, xx-large

CSS Units

- When you use specify lengths in CSS, you should generally use the relative units appropriate to the property
 - For width properties, use *ex* units, which corresponds to the width of the lower-case 'x' character
 - For height properties, use *em* units, which corresponds to the height of the capital 'M' character
- Relative units are preferred to absolute units, like *px*, *pt*, and *in*
- One reason for this is that display surfaces may have more or less than 96 pixels per inch, in which case your 14px font will look too small
- Other relative values like *smaller*, *bolder*, and percentages are also better choices than the absolute values

Pixels & Points

- A pixel (px) is a single square 'picture element' (hence pix-el), i.e. a single dot in your image; a 10 x10 image is made up of a set of pixels in a grid 10 wide by 10 high, totaling 100 pixels
- The 'point' (pt) on the other hand is a unit of length, commonly used to measure the height of a font, but technically capable of measuring any length
 - 1pt is equal to exactly 1/72th of an inch; in traditional print *technically* 72pt is 1 inch (0.996264 inches)
- How many pixels = 1pt depends on the resolution of your image; if your resolution is 72ppi (pixels per inch), then one point will equal exactly one pixel

Chart That Converts Points To Pixels (And Ems And %)

[It's an approximation, which will depend on font, browser and OS, but it's a good starting point]

Points	Pixels	Ems	Percent
6pt	8px	0.5em	50%
7pt	9px	0.55em	55%
7.5pt	10px	0.625em	62.5%
8pt	11px	0.7em	70%
9pt	12px	0.75em	75%
10pt	13px	0.8em	80%
10.5pt	14px	0.875em	87.5%
11pt	15px	0.95em	95%
12pt	16px	1em	100%
13pt	17px	1.05em	105%
13.5pt	18px	1.125em	112.5%
14pt	19px	1.2em	120%
14.5pt	20px	1.25em	125%
15pt	21px	1.3em	130%
16pt	22px	1.4em	140%
17pt	23px	1.45em	145%
18pt	24px	1.5em	150%
20pt	26px	1.6em	160%
22pt	29px	1.8em	180%
24pt	32px	2em	200%
26pt	35px	2.2em	220%
27pt	36px	2.25em	225%
28pt	37px	2.3em	230%
29pt	38px	2.35em	235%
30pt	40px	2.45em	245%
32pt	42px	2.55em	255%
34pt	45px	2.75em	275%
36pt	48px	3em	300%



Picking a Font

- Text characters are based on **fonts** that define the style and appearance of each character in the alphabet
- The general structure of defining font for any page element is

`font-family: fonts;`

where *fonts* is a comma-separated list, also known as a **font stack**

Picking a Font (con't)

- **Specific font** – Identified by name and based on a font definition file stored in a user's computer or accessible on the web
- **Generic font** – Describes the general appearance of the characters in the text but does not specify any particular font definition file
 - Supports the font groups **serif, sans-serif, monospace, cursive, and fantasy**

Font Specification

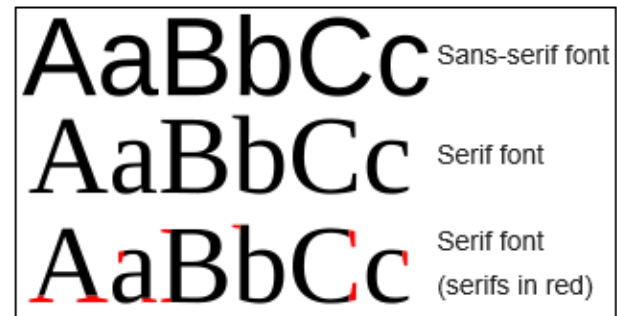
- Specific Fonts
 - Font-family: Arial
 - Values: Arial, Times New Roman, Courier, ...
- Font Family
 - font-family: sans-serif
 - Values: sans-serif, serif, monospaced, cursive, ...
- Recommended specify both (if font not available, will use one from specified family)
 - font-family: Arial, sans-serif

Font Family

- For font-family, you can specify an actual font name, like "Courier New"
- If the name has spaces, you should quote it, and it is case-insensitive
- You can specify a comma-separated list of font names, which will be used in the order listed when some are not found on the user's system
- Designers should use the most desired font first, the most compatible font second-to-last, and the generic font family last and always. For example a common font-family value is:
 - Verdana, Arial, sans-serif

Sans-serif Fonts

- A **sans-serif**, **sans serif**, **gothic**, or simply **sans** letterform is one that **does not have extending features** called "serifs" at the end of strokes
- Sans-serif fonts tend to have less line width variation than serif fonts
- In most print, they are often used for headings rather than for body text
- They are often used to convey simplicity and modernity or minimalism
- **Sans-serif fonts have become the most prevalent for display of text on computer screens**



Font Properties

- Can specify all font properties at once:
 - font:15px arial, sans-serif;
 - font: italic bold 12px/30px Georgia, serif;

Property/Value	Description
<i>font-style</i>	Specifies the font style. See font-style for possible values
<i>font-variant</i>	Specifies the font variant. See font-variant for possible values
<i>font-weight</i>	Specifies the font weight. See font-weight for possible values
<i>font-size/line-height</i>	Specifies the font size and the line-height. See font-size and line-height for possible values
<i>font-family</i>	Specifies the font family. See font-family for possible values

“Web Safe” Fonts

Arial

abcdefghijklmnopqrstuvwxyz/1234567890

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

Arial Black

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Arial Black', Gadget, sans-serif;

Century Gothic

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Century Gothic', sans-serif;

Comic Sans MS

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Comic Sans MS', cursive;

Courier New

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Courier New', Courier, monospace;

Georgia

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: Georgia, serif;

Impact

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: Impact, Charcoal, sans-serif;

Lucida Console

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Lucida Console', Monaco, monospace;

Lucida Sans Unicode

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Lucida Sans Unicode', 'Lucida Grande', sans-serif;

Palatino Linotype

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Palatino Linotype', 'Book Antiqua', Palatino, serif;

Tahoma

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: Tahoma, Geneva, sans-serif;

Times New Roman

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Times New Roman', Times, serif;

Trebuchet MS

abcdefghijklmnopqrstuvwxyz/1234567890

font-family: 'Trebuchet MS', Helvetica, sans-serif;

Verdana

abcdefghijklmnopqrstuvwxyz/1234567890

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

Web Fonts

- **Web font** – Definition font is supplied to the browser in an external file
- The format most universally accepted in almost all current browsers and on almost all devices is the Web Open Font Format (WOFF)

Format	Description	Browser
Embedded OpenType (EOT)	A compact form of OpenType fonts designed for use as embedded fonts in style sheets	Internet Explorer (IE)
TrueType (TTF)	Font standard used on the Mac OS and Microsoft Windows operating systems	IE, Firefox, Chrome, Safari, Opera
OpenType (OTF)	Font format built on the TrueType format developed by Microsoft	IE, Firefox, Chrome, Safari, Opera
Scalable Vector Graphics (SVG)	Font format based on an XML vocabulary designed to describe resizable graphics and vector images	Safari
Web Open Font Format (WOFF)	The W3C recommendation font format based on OpenType and TrueType with compression and additional metadata	IE, Firefox, Chrome, Safari, Opera

Web Fonts (con't)

- To access and load a web font, add the `@font-face` rule to the style sheet
- Once a web font is defined using the `@font-face` rule, it is included in the font stack
- General syntax to include `@font-face`

```
@font-face {  
    font-family: name;  
    src: url ('url1') format ('text1'),  
         url ('url2') format ('text2'),  
    ...;  
    descriptor1: value1;  
    descriptor2: value2;  
    ...  
}
```

Web Fonts (con't)

■ Where:

- `name` is the name of the font
- `url` is the location of the font definition file
- `text` is an optional text description of the font format
- and the `descriptor1: value1; pairs` are optional style properties of the font

Controlling Spacing and Indentation

- Kerning sets the space between characters, while **tracking** sets the space between words
- To control an element's kerning and tracking are:
`letter-spacing: value;`
`word-spacing: value;`
- **Leading** – sets the space between lines of text and is defined with the following `line-height` property:
`line-height: size;`
- Setting the indentation for the first line of text block by using the `text-indent` property
`text-indent: size;`
- A negative indentation value creates a **hanging indent**

Font Styles

- To specify the font style, use
`font-style: type;`
where *type* is normal, italic, or oblique
- To change the weight of the font, use
`font-weight: weight;`
where *weight* is the level of bold applied to the text
- To specify a text decoration, use
`text-decoration: type;`
where *type* is none, underline, overline, or line-through
- To transform text, use
`text-transform: type;`
where *type* is capitalize, uppercase, lowercase, or none
- To display a font variant of text, use
`font-variant: type;`
where *type* is normal or small-caps

Aligning Text Horizontally and Vertically

- To horizontally align the text , use `text-align: alignment;`
where *alignment* **is** left, right, center, **or** justify
- To vertically align the text within each line, use `vertical-align: alignment;`
where *alignment* **is** baseline, bottom, middle, sub, super, text-bottom, text-top, **or** top

Aligning Text Horizontally and Vertically (con't)

Value	Description
baseline	Aligns the baseline of the element with the baseline of the parent element
bottom	Aligns the bottom of the element with the bottom of the lowest element in the line
middle	Aligns the middle of the element with the middle of the surrounding content in the line
sub	Subscripts the element
super	Superscripts the element
text-bottom	Aligns the bottom of the element with the bottom of the text in the line
text-top	Aligns the top of the element with the top of the text in the line
top	Aligns the top of the element with the top of the tallest object in the line

CSS Font & Text Wizard

<http://www.somacore.com/p334.php>

Font	
font-family	serif sans-serif cursive fantasy monospace custom:
font-style	normal italic oblique
font-variant	normal small-caps
font-weight	normal bold bolder lighter 100 200 500 600 900
font-size	medium small large smaller larger custom:
Text	
line-height	100% 200% 80% 2em 1em 0.8em custom:
word-spacing	normal 1ex 1.5ex 2ex 5ex
letter-spacing	normal 0.1ex 0.3ex 0.75ex 1ex
text-decoration	none underline overline line-through blink
text-transform	none capitalize uppercase lowercase
text-align	left right center justify
text-indent	0ex 1ex 2ex 5ex 10ex 10% 20%

Working with CSS Color

- CSS supports 147 color names
- **Color values** – Values in which the color is given by an exact numeric representation
 - RGB values or HSL values
- **RGB triplet** – The intensity of primary colors expressed as a set of numbers

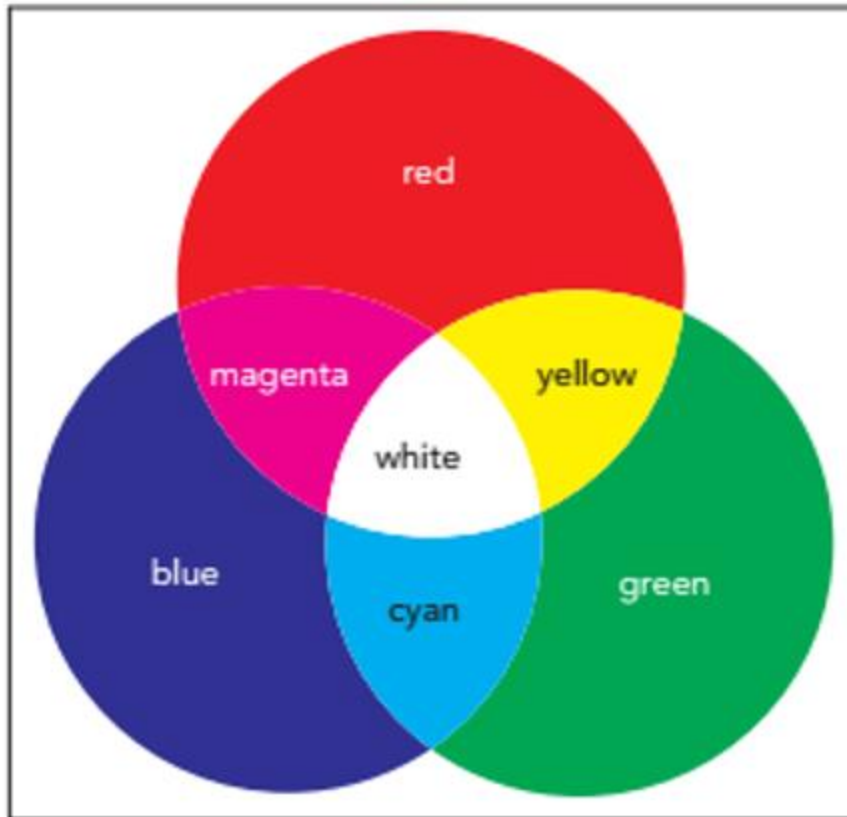
`rgb (red, green, blue)`

- **Hexadecimal numbers** – A number expressed in the base 16 numbering system

Color Names (partial)

white	champagne	palepink	paleblue
pear	apricot	cerise	azure
lime	orange	ruby	cerulean
yellowgreen	copper	crimson	steelblue
beige	salmon	rosepink	aqua
khaki	redorange	magenta	turquoise
olive	rust	redviolet	seagreen
umber	sienna	plum	aquamarine
ivory	seashell	violet	emerald
lemon	peach	fuchsia	jade
yellow	coral	purple	palegreen
gold	indianred	eggplant	applegreen
cream	pink	lavender	green
amber	red	amethyst	forestgreen
goldenrod	carmine	blueviolet	celadon
ocher	maroon	indigo	sage
bisque	mistyrose	babyblue	slate
tan	oldrose	skyblue	mauve
bronze	rosybrown	blue	taupe
sepia	rosewood	ultramarine	silver

RGB Color Values

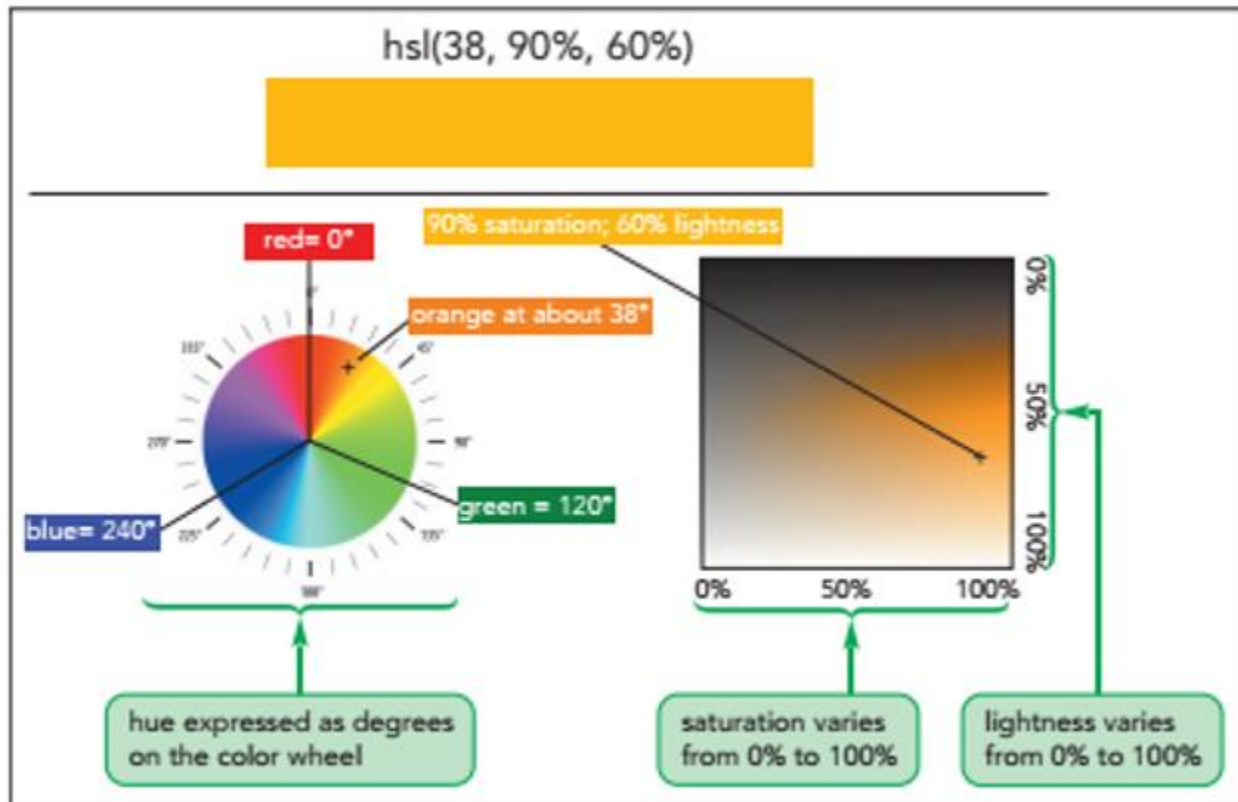


maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

HSL Color Values

- **Hue** – Tint of a color, represented by a direction on a color wheel
- **Saturation** – Measures the intensity of a color and ranges from 0% (no color) up to 100% (full color)
- **Lightness** – Measures the brightness of a color and ranges from 0% (black) up to 100% (white)

HSL Color Values (con't)



Defining Semi-Opaque Colors

- **Opacity** – Defines how solid a color appears
- A color's opacity is specified using the following properties:

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

`hsla(hue, saturation, lightness, opacity)`

where `opacity` sets the opacity of the color ranging from 0 (completely transparent) up to 1.0 (completely opaque)

Methods for Color Setting

In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato":

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

Same as color name "Tomato", but 50% transparent:

`rgba(255, 99, 71, 0.5)`

`hsla(9, 100%, 64%, 0.5)`

Setting Text and Background Colors

- CSS defines the text and background color for each element on a webpage

`color: color;`

`background-color: color;`

where *color* is a color name or a color value

`EM {background-color: pink; color: black}`

Internal CSS Style Tag (Element)

- <STYLE TYPE = “text/css”>
 - ...style rules...
- </STYLE>

MIME Types

Multipurpose Internet Mail Extensions
Specifies format type and sub type for different kinds of standard Internet data:

- text/html
- text/javascript
- text/css
- image/gif

- Normally goes in the heading section so that it applies to entire HTML document
- TYPE=“text/css” is required in XHTML **but not in HTML 5**

Style Settings

- Format:
 - *selector* {declarations}
- Selecting (override) existing tag values
 - `EM {background-color: pink; color: black}`
- Selecting classes (dot notation) – used when style is **not tied to a particular set of elements**
 - `.reverse {background-color: black; color: white}`
- Selecting ID's (# notation) – id's are unique in the document, and typically **tied to a set of elements**
 - `#reverse {background-color: black; color: white}`

Methods to Select Elements for Styling

- This style rule matches every h1 element in an HTML document, regardless of its location within the page:

```
h1 {  
  color: red;  
}
```

- To direct a style rule to an element at a specific location use **selector patterns** to match only those page elements that match a specified pattern

Attribute Selectors

- Selectors can be defined based on attributes and attribute values within elements
 - `id` – Identifies specific elements within the document
 - `Class` – Identifies a group of elements that share a similar characteristic or property

Attribute Selectors (con't)

Selector	Selects	Example	Selects
<code>elem#id</code>	Element <i>elem</i> with the ID value <i>id</i>	<code>h1#intro</code>	The <code>h1</code> heading with the id <i>intro</i>
<code>#id</code>	Any element with the ID value <i>id</i>	<code>#intro</code>	Any element with the id <i>intro</i>
<code>elem.class</code>	All <i>elem</i> elements with the <i>class</i> attribute value <i>class</i>	<code>p.main</code>	All paragraphs belonging to the <i>main</i> class
<code>.class</code>	All elements with the class value <i>class</i>	<code>.main</code>	All elements belonging to the <i>main</i> class
<code>elem[att]</code>	All <i>elem</i> elements containing the <i>att</i> attribute	<code>a[href]</code>	All hypertext elements containing the href attribute
<code>elem[att="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute equals <i>text</i>	<code>a[href="top.html"]</code>	All hypertext elements whose href attribute equals <i>top.html</i>

Attribute Selectors (con't)

Selector	Selects	Example	Selects
<code>elem[att~="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute contains the word <i>text</i>	<code>a[rel~="glossary"]</code>	All hypertext elements whose <i>rel</i> attribute contains the word <i>glossary</i>
<code>elem[att ="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute value is a hyphen-separated list of words beginning with <i>text</i>	<code>p[id ="first"]</code>	All paragraphs whose <i>id</i> attribute starts with the word <i>first</i> in a hyphen-separated list of words
<code>elem[att^="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute begins with <i>text</i>	<code>a[rel^="prev"]</code>	All hypertext elements whose <i>rel</i> attribute begins with <i>prev</i>
<code>elem[att\$="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute ends with <i>text</i>	<code>a[href\$="org"]</code>	All hypertext elements whose <i>href</i> attribute ends with <i>org</i>
<code>elem[att*="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute contains the value <i>text</i>	<code>a[href*="faq"]</code>	All hypertext elements whose <i>href</i> attribute contains the text string <i>faq</i>

Contextual Selectors

- **Contextual selector** – Specifies the context under which a particular page element is matched
- Context is based on the hierarchical structure of a document, which involves the relationships between a **parent element** containing one or more **child elements** and within those child elements several levels of **descendant elements**
- To match any element, a **wildcard selector** with the * character is used
- **Sibling selectors** are used to select elements based on elements that are adjacent to them in the document hierarchy

Contextual Selectors (con't)

Selector	Description
<code>*</code>	Matches any element
<code>elem</code>	Matches the element <i>elem</i> located anywhere in the document
<code>elem1, elem2, ...</code>	Matches any of the elements <i>elem1</i> , <i>elem2</i> , etc.
<code>parent descendant</code>	Matches the <i>descendant</i> element that is nested within the <i>parent</i> element at some level
<code>parent > child</code>	Matches the <i>child</i> element that is a child of the <i>parent</i> element
<code>elem1 + elem2</code>	Matches <i>elem2</i> that is immediately preceded by the sibling element <i>elem1</i>
<code>elem1 ~ elem2</code>	Matches <i>elem2</i> that follows the sibling element <i>elem1</i>

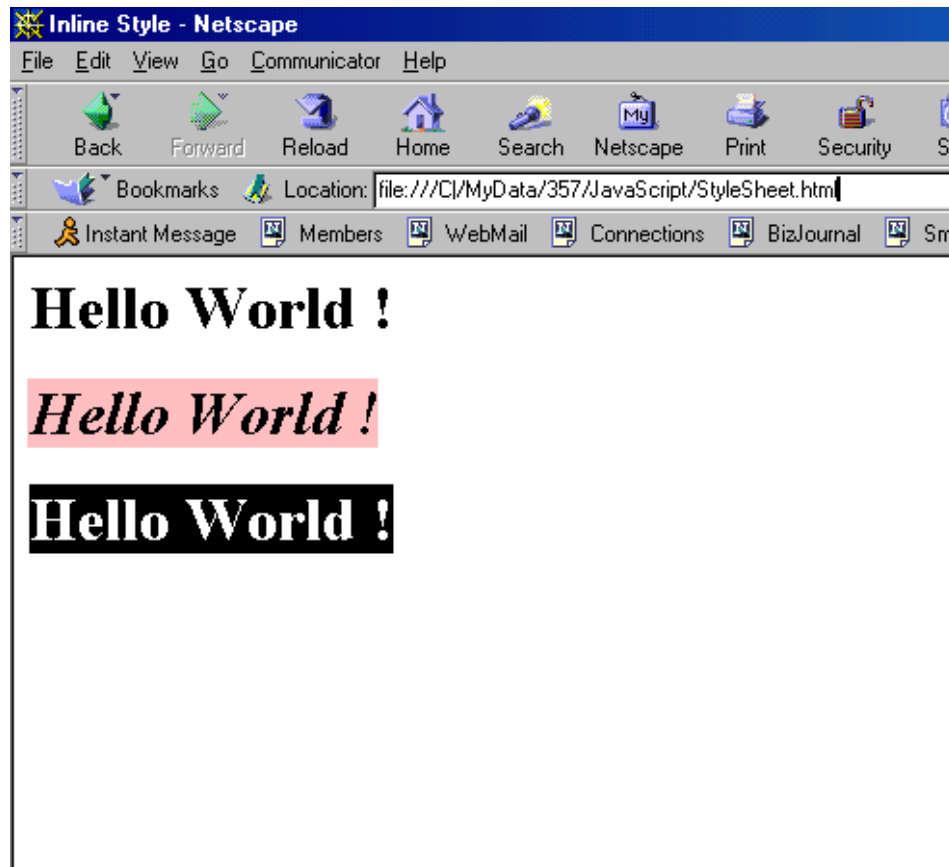
LAB

Try this in different browsers to see differences/problems !

```
■ <HTML>
■ <HEAD><TITLE>Styles</TITLE>
■ <SCRIPT LANGUAGE="JavaScript">
■ </SCRIPT>
■ <STYLE TYPE="text/css">
■     EM {background-color: pink; color: black}
■     P {font-size: 18pt}
■     .reverse {background-color: black; color: white}
■ </STYLE>
■ </HEAD>
■ <BODY>
■ <H1>Hello World !</H1>
■ <P><H1><EM>Hello World !</EM></H1></P>
■ <P><H1 CLASS = "reverse">Hello World !</H1></P>
■ </BODY></HTML>
```

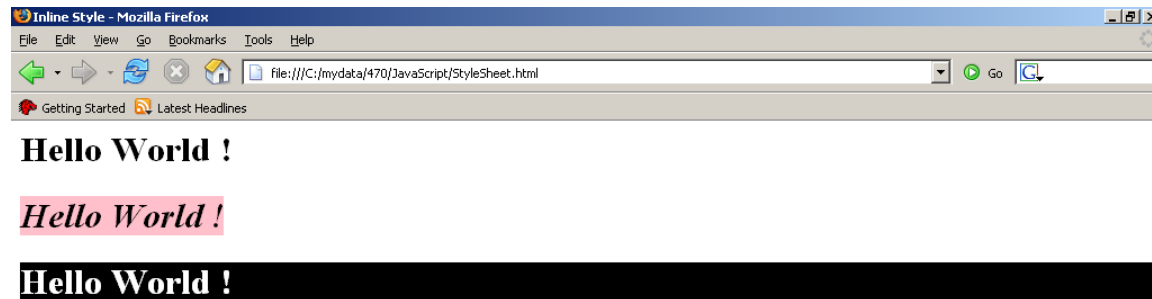
Style Settings

[StyleSheet.html]



This is in the old Netscape !

Newer Rendering in IE & FireFox



Using “ID” for Area Designation (id’s should be unique)

```
■ <HTML>
■ <HEAD><TITLE>Inline Style</TITLE>
■ <SCRIPT LANGUAGE="JavaScript">
■ <!--
■
■ // -->
■ </SCRIPT>
■ <STYLE TYPE="text/css">
■     EM {background-color: pink; color: black}
■     P {font-size: 18pt}
■     #reverse {background-color: black; color: white}
■ </STYLE></HEAD>
■ <BODY>
■ <H1>Hello World !</H1>
■ <P><H1><EM>Hello World !</EM></H1></P>
■ <P><H1 ID = "reverse">Hello World !</H1></P>
■ </BODY></HTML>
```

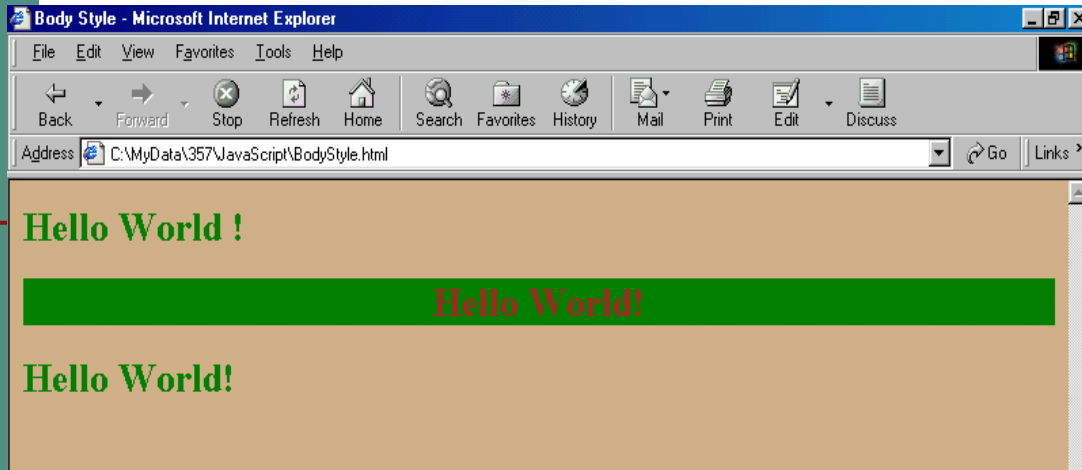
Styling the Entire Body

- You can also set the style for the entire body of your document including backgrounds:
 - `<STYLE TYPE = "text/css">`
 - `BODY {background-image: url(mybackground.gif);`
 - `background-position: center;`
 - `background-repeat: no-repeat}`
 - `P {font-size: 2em; color: red; text-indent: 1em}`
 - `</STYLE>`
- Many other background properties are available

Example: Body Styling

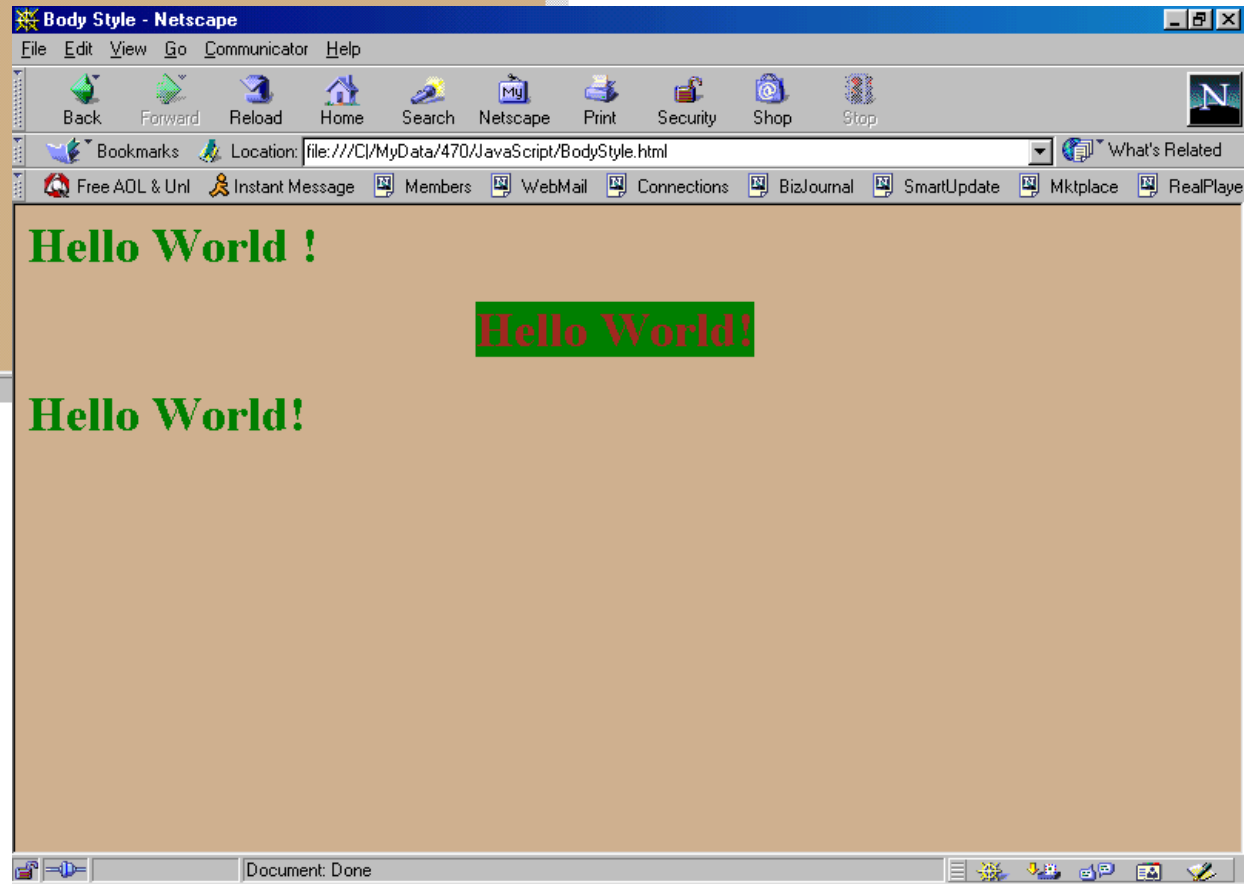
(SPAN & DIV are non formatting tags, DIV starts on it's own line)

```
<HTML>
<HEAD><TITLE>Body Style</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
// -->
</SCRIPT>
<STYLE TYPE="text/css">
    BODY {background-color: tan; color: green}
    DIV {text-align: center; background-color: green; color:
brown}
</STYLE></HEAD>
<BODY>
<H1>Hello World !</H1>
<DIV>
<H1>Hello World!</H1> ← Note this "div" overrides body setting
</DIV>
<H1>Hello World!</H1>
</BODY></HTML>
```



IE

Netscape



Styling the Scrollbar

```
■ <style type="text/css">
■ <!--
■ body {
  ■ scrollbar-arrow-color: #ffffff;
  ■ scrollbar-face-color: #000060;
  ■ scrollbar-3dlight-color: #ffffff;
  ■ scrollbar-highlight-color: #ffffff;
  ■ scrollbar-shadow-color: #000000;
  ■ scrollbar-darkshadow-color: #000000;
  ■ scrollbar-track-color: #BE1515; font-family:Arial; font-size:10pt
■ }
■ h1      { font-family: Arial; font-size: 12pt; font-weight: bold }
■ p      { font-family: Arial; font-size: 10pt }
■ h2      { font-family: Arial; font-size: 10pt; font-weight: bold }
■ -->
■ </style>
```

Table Backgrounds

[can also style table cells]

■ Style Tag

- `<STYLE TYPE="text/css">`
 - `.tealtable, .tealtable TD, .tealtable TH`
 - `{ background-color:teal; color:white; }`
- `</STYLE>`

■ Use inside table:

- `<TABLE CELLPADDING=5 CLASS="tealtable"> <TR>`
`<TH>Operator</TH> <TH>Ext</TH> <TH>City</TH> </TR>`
`<TR> <TD>Starflower</TD> <TD>8172</TD> <TD>San`
`Francisco</TD> </TR> <TR> <TD>Melody</TD>`
`<TD>5673</TD> <TD>San Pedro</TD> </TR> <TR>`
`<TD>Symphony</TD> <TD>3820</TD> <TD>Montreal</TD>`
`</TR> </TABLE>`

- NAV doesn't interpret styles code the same as IE when they are applied to tables
- To compensate for this, set rules for three different types of elements: any element associated with the **tealtable** class, **<TD ...>** elements inside **tealtable** elements, and **<TH ...>** elements inside **tealtable** elements

Using CSS for Table Borders (con't)

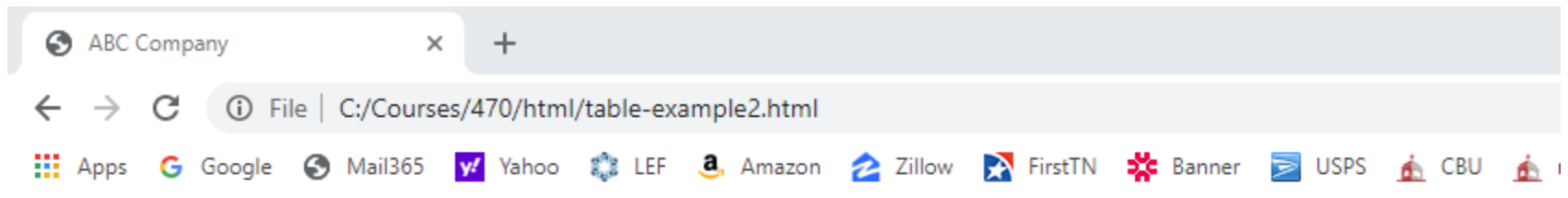
table-example2.html - Notepad

File Edit Format View Help

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>ABC Company</title>
    <style type="text/css">
      table.sample {
        border-width: 5px; border-spacing: 2px; border-style: outset; border-color: gray; border-collapse: separate; background-color: white;
      }
      table.sample th {
        border-width: 1px; padding: 1px; border-style: inset; border-color: gray; background-color: white; -moz-border-radius: ;
      }
      table.sample td {
        border-width: 1px; padding: 1px; border-style: inset; border-color: gray; background-color: white; -moz-border-radius: ;
      }
    </style>
  </head>
  <body>
    <h1 style="text-align:center">ABC Product Sales</h1>
    <table class="sample" width=400>
      <caption align=top><h2>Product Sales by Type and Color</h2></caption>
      <tr>
        <th>Product</th><th>Red</th><th>Green</th><th>Blue</th><th>Total</th>
      </tr>
      <tr>
        <th align="left">Pens</th><td align="center">11</td><td align="center">12</td><td align="center">6</td><td align="center">29</td>
      </tr>
      <tr>
        <th align="left">Pencils</th><td align="center">8</td><td align="center">5</td><td align="center">3</td><td align="center">16</td>
      </tr>
      <tr>
        <th align="left">Erasures</th><td align="center">21</td><td align="center">41</td><td align="center">9</td><td align="center">34</td>
      </tr>
      <tr>
        <th align="right">Total</th><td align="center">40</td><td align="center">21</td><td align="center">18</td><td align="center">79</td>
      </tr>
    </table>
  </body>
</html>
```



Using CSS for Table Borders (con't)



ABC Product Sales

Product Sales by Type and Color

Product	Red	Green	Blue	Total
Pens	11	12	6	29
Pencils	8	5	3	16
Erasures	21	41	9	34
Total	40	21	18	79

Table Styling Wizard

[<http://www.somacn.com/p141.php>]

TABLE										
border-width (px)	1px	2px	3px	4px	5px	thin	medium	thick	0px	
border-spacing	2px	1px	3px	4px	5px	0px				
border-style	outset	none	hidden	dotted	dashed	solid	double	ridge	groove	inset
border-color	gray	white	blue	green	black	red	custom:			
border-collapse	separate	collapse								
background-color	white	#FFFFFF0	#FAF0E6	#FFF5EE	#FFFAFA	custom:				

TD/TH										
border-width (px)	1px	2px	3px	4px	5px	thin	medium	thick	0px	
padding	1px	2px	3px	4px	5px	0px				
border-style	inset	none	hidden	dotted	dashed	solid	double	ridge	groove	outset
border-color	gray	white	blue	green	black	red	custom:			
background-color	white	#FFFFFF0	#FAF0E6	#FFF5EE	#FFFAFA	custom:				
-moz-border-radius	0px	3px	6px	9px	12px					

Class Exercise

- Try some style tags for the formatting your home page using an internal STYLE!
- Common CSS properties:
 - color
 - background-color, background image
 - text-decoration(none, underline, blink, ...)
 - font-color, font-size, font-weight (bold, ...), font-family,
 - word spacing, letter spacing, text align, ...

More Style Setting Capabilities

- Sub-classes of an existing tag (<TAG class = “nodec” ...)
 - TAG.nodec {text-decoration: none}
- “Psuedo” classes available for some tags (<A ...):
 - A:link{ color: #E4FFB0; text-decoration:none}
 - A:visited { color: #9876AA;
text-decoration:none}
 - A:active{ color: #FFCAB0;
text-decoration:none}
 - A:hover{ color: #663300;
text-decoration:none;font-weight: bold}
- Nesting (nested unordered lists will be underlined):
 - UL UL {text-decoration: underline}

Pseudo-Classes

- **Pseudo-class** – classifies an element based on its current status, position, or use in the document

`element: pseudo-class`

where `element` is an element from the document and `pseudo-class` is the name of a CSS pseudo-class

- **Structural pseudo-class** – classifies an element based on its location within the structure of the HTML document

Pseudo-Classes (con't)

Pseudo-Class	Matches
<code>:root</code>	The top element in the document hierarchy (the html element)
<code>:empty</code>	An element with no content
<code>:only-child</code>	An element with no siblings
<code>:first-child</code>	The first child of the parent element
<code>:last-child</code>	The last child of the parent element
<code>:first-of-type</code>	The first descendant of the parent that matches the specified type
<code>:last-of-type</code>	The last descendant of the parent that matches the specified type
<code>:nth-of-type(<i>n</i>)</code>	The <i>n</i> th element of the parent of the specified type
<code>:nth-last-of-type(<i>n</i>)</code>	The <i>n</i> th from the last element of the parent of the specified type
<code>:only-of-type</code>	An element that has no siblings of the same type
<code>:lang(<i>code</i>)</code>	The element that has the specified language indicated by <i>code</i>
<code>:not(<i>selector</i>)</code>	An element not matching the specified <i>selector</i>

Pseudo-Classes (con't)

■ Applying pseudo classes to list items:



Pseudo-Classes (con't)

■ Pseudo-classes for links (<A>):

Pseudo-Class	Description
<code>:link</code>	The link has not yet been visited by the user.
<code>:visited</code>	The link has been visited by the user.
<code>:active</code>	The element is in the process of being activated or clicked by the user.
<code>:hover</code>	The mouse pointer is hovering over the element.
<code>:focus</code>	The element is receiving the focus of the keyboard or mouse pointer.

Pseudo-Elements

- **Pseudo-element** – An object that exists only in the rendered page
- Pseudo-elements can be selected using the following CSS selector:

`element::pseudo-element`

where *element* is an element from the HTML file and *pseudo-element* is the name of a CSS pseudo-element

Pseudo-Element	Description
::first-letter	The first letter of the element text
::first-line	The first line of the element text
::before	Content inserted directly before the element
::after	Content inserted directly after the element

Style Inheritance

- **Style inheritance** – Process in which properties are passed from a parent element to its children
- For example, the following style rule sets the color of article text to blue and the rule is passed to any paragraph or other elements nested within that article

```
article {color: blue;}  
p {text-align: center;}
```

External Style Sheet

- External style sheets use the “relationship” link attribute (REL):
 - `<LINK REL="stylesheet" TYPE = "text/css" HREF= "...">`
- External style sheets are used to keep all the pages in one's site consistent !!!
- You can also use external style sheets made by organizations:
 - `<LINK REL="stylesheet" TYPE="text/css" href="http://www.w3.org/StyleSheets/Core/Traditional">`

Example of External Style

■ ExternalStyle.html

- `<HTML><HEAD><TITLE>External Style</TITLE>`
- `<SCRIPT LANGUAGE="JavaScript">`
- `<!--`
- `// -->`
- `</SCRIPT>`
- `<LINK REL="stylesheet" TYPE="text/css" HREF="ExternalStyle.css">`
- `</HEAD>`
- `<BODY>`
- `<H1>Hello World !</H1>`
- `<P><H1><EM>Hello World !</EM></H1></P>`
- `<P><H1 CLASS = "reverse">Hello World !</H1></P>`
- `</BODY></HTML>`

■ ExternalStyle.css

- `EM`
`{background-color: pink;`
`color: black}`
- `P {font-size: 18pt}`
- `.reverse`
`{background-color: black;`
`color: white}`

Another Example of an External Style Sheet

- BODY {background: url(background.jpg) #4169E1;
background-attachment: fixed
scrollbar-3d-light-color: Blue;
scrollbar-arrow-color: Yellow;
scrollbar-base-color: Yellow;
scrollbar-dark-shadow-color: Yellow;
scrollbar-face-color: #6666FF;
scrollbar-highlight-color: yellow;
scrollbar-shadow-color: Yellow; }
- A:link { color: blue }
- A:visited, A:active { color: yellow }
- A:link IMG { border: solid blue }
- A:visited IMG { border: solid yellow }
- A:active IMG { border: solid yellow }
- A:hover { text-decoration: underline;
color: blue; background-color: yellow }

Another Example of an External Style Sheet

```

■ <!--
■ A:link
■ { text-decoration: none; color:#202020 }
■ A:visited
■ { text-decoration: none; color:#202020 }
■ A:active
■ { text-decoration: line-through; color:#404040 }
■ A:hover
■ { text-decoration: line-through; color:#404040; }

■ { cursor: crosshair }

■ body
■ { background-color: none;
■ background-image: url(http://www.cbu.edu/~dhallida/470/chalk.jpg);
■ background-repeat: repeat;
■ background-attachment:fixed;
■ font-family: verdana;
■ color: #000000 ;
■ font-weight:normal;
■ font-size:xx-small;
■ scrollbar-face-color: FFFFFFFF;
■ scrollbar-highlight-color: FFFFFFFF;
■ scrollbar-3dlight-color: FFFFFFFF;
■ scrollbar-darkshadow-color: FFFFFFFF;
■ scrollbar-shadow-color: FFFFFFFF;
■ scrollbar-arrow-color: 000000;
■ scrollbar-track-color: FFFFFFFF; }

■ td
■ { font-family: verdana;
■ color: #000000;
■ font-size:xx-small; }

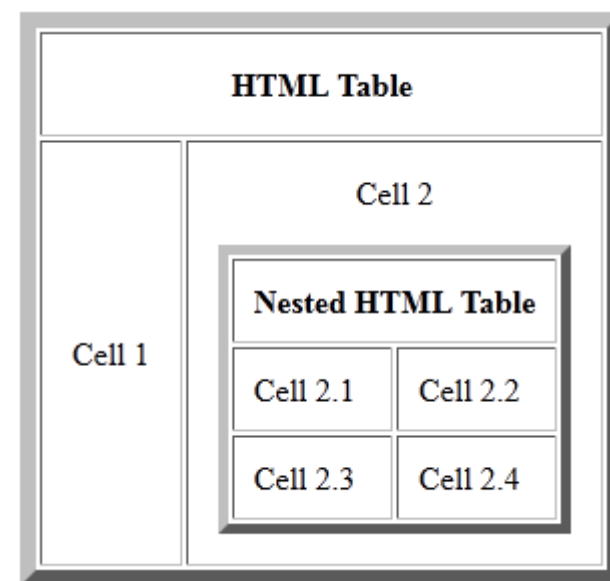
■ input, textarea
■ { background: #FFFFFFF;
■ font-family: verdana;
■ color: #000000;
■ font-size:xx-small;
■ border-style: dotted;
■ border-color: #000000;
■ border-width: 2 }
■ -->
```

Positioning Elements

- The STYLE attribute or tag can be used to position page elements in either an **absolute or relative** mode
- Positioning can be set up in the stylesheet as a class or ID (#)
- The position property is either “absolute” or “relative”
- Position is given relative to the top, bottom, right, and left margins (in pixels, inches, etc.; default is px)
- A “z-index” may also be specified for the dimension into the page; elements with higher z-index are positioned on top (default is zero); do not use negative numbers for browser compatibility reasons

CSS Positioning vs Tables

- CSS has some advantages over the use of **nested tables** for positioning:
 - Areas can overlap
 - Z positioning
- However, tables are compatible with older browsers
 - Nested tables are typically needed for fine positioning



Absolute Positioning

- <HTML>
- <HEAD><TITLE>Positioning</TITLE>
- <SCRIPT LANGUAGE="JavaScript">
- <!--
-
- // -->
- </SCRIPT>
- </HEAD>
- <BODY>
- <IMG SRC="sea.gif"
- STYLE="position: absolute; top: 0px; left 0px; z-index:1">
- <H1 STYLE="position: absolute; top: 20px; left: 50px; z-index:2">
- ■ Hello World !
- </H1>
- </BODY></HTML>



Relative Positioning

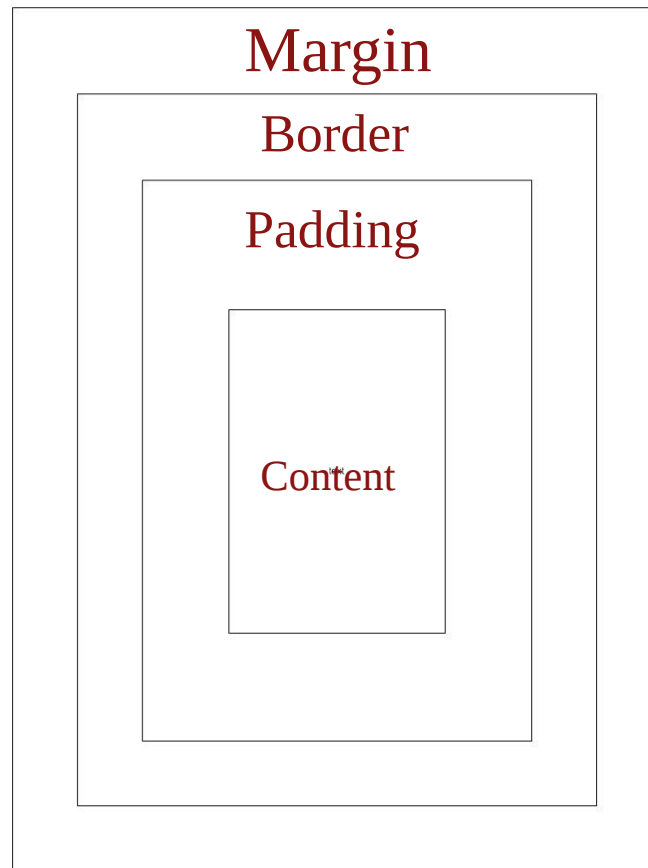
```
<HTML>
<HEAD><TITLE>Relative Positioning</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
// -->
</SCRIPT>
<STYLE TYPE="text/css">
    SPAN {color:red; font-size:.5em}
    .super {position:relative; top:-1ex}
    .sub {position:relative; bottom:-1ex}
</STYLE></HEAD>
<BODY>
Hello <SPAN Class="super">Hello</SPAN>
World <SPAN Class="sub">World</SPAN>!
</BODY></HTML>
```



The Box Model & “Layering”

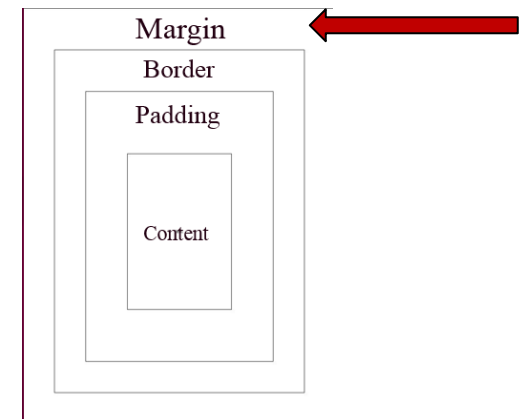
- Block level elements (such as DIV, SPAN, P, Hi) have a “box model” that can apply to their content
- The content of the block is surrounded by “padding”, a border, and a margin (see next figure)
- The padding size and margin size can be set for each side of this “box”
- The border-width can be set to a size or type (thick, thin, medium); and the border-style can be set (none, hidden, solid, double, groove, ridge, inset, and outset)
- In addition to the positioning discussed previously, the “box” can be set to “float”, and content before and after the box will wrap around it (until a <BR clear=all>)

Box Model

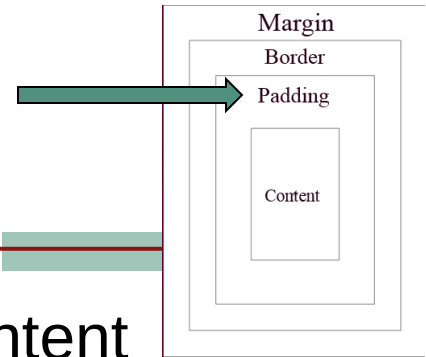


Margin

- The margin clears an area around an element (outside the border) – px or em
- The margin does not have a background color, and is completely transparent:
 - `margin:100px 50px;`
- The top, right, bottom, and left margin can be changed independently using separate properties:
 - `margin-top:100px;`
`margin-bottom:100px;`
`margin-right:50px;`
`margin-left:50px;`

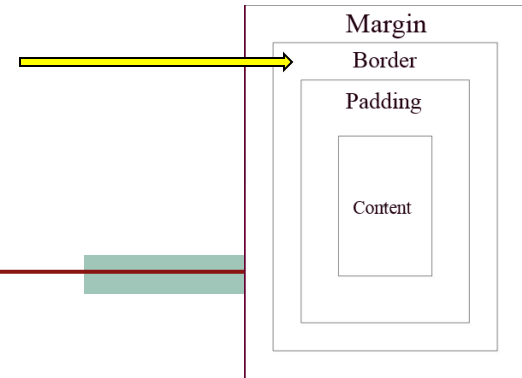


Padding



- The padding clears an area around the content (inside the border) of an element
- The padding is affected by the background color of the element
- The top, right, bottom, and left padding can be changed independently using separate properties:
 - `padding-top:25px;`
`padding-bottom:25px;`
`padding-right:50px;`
`padding-left:50px;`
- A shorthand padding property can also be used, to change all paddings at once: `padding:25px 50px;`

Border



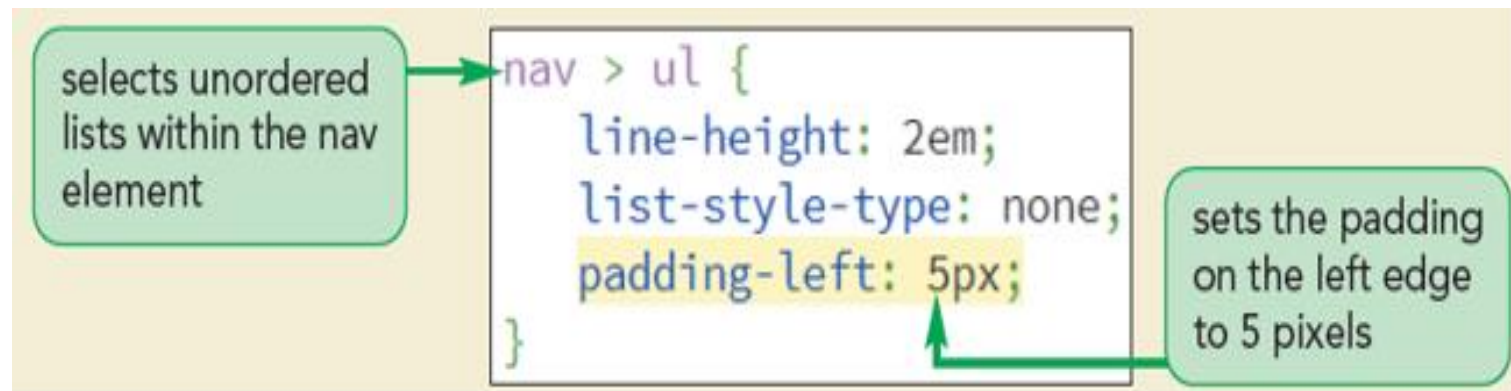
- The border-style property specifies what kind of border to display (none, solid, dashed, dotted, etc.)
- The border-color specifies the border color
- The border-width specified the border size in px or thin, medium, thick
- The border property is a shorthand for specifying width, style, and color

Setting the Padding Space

- To set the width of the padding space, use the following `padding` property

`padding: size;`

where *size* is expressed in one of the CSS units of length or the keyword `auto` to let the browser automatically choose the padding



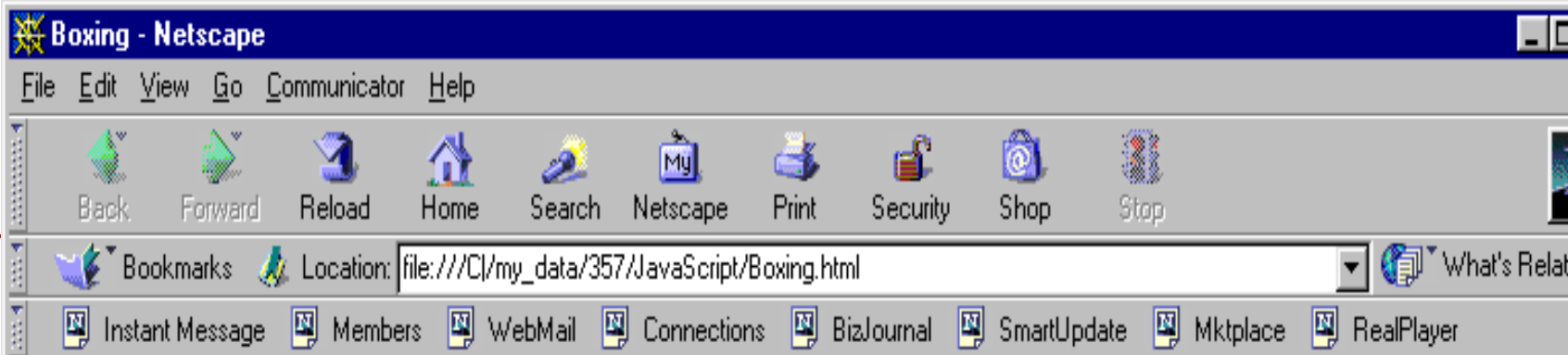
Setting the Margin and the Border Spaces

- To set the size of the margin around block elements, use
`margin: size;`
`margin: top right bottom left;`
- To set the size of the border space, use
`border-width: size;`
`border-width: top right bottom left;`



Without BREAK

[illegible]

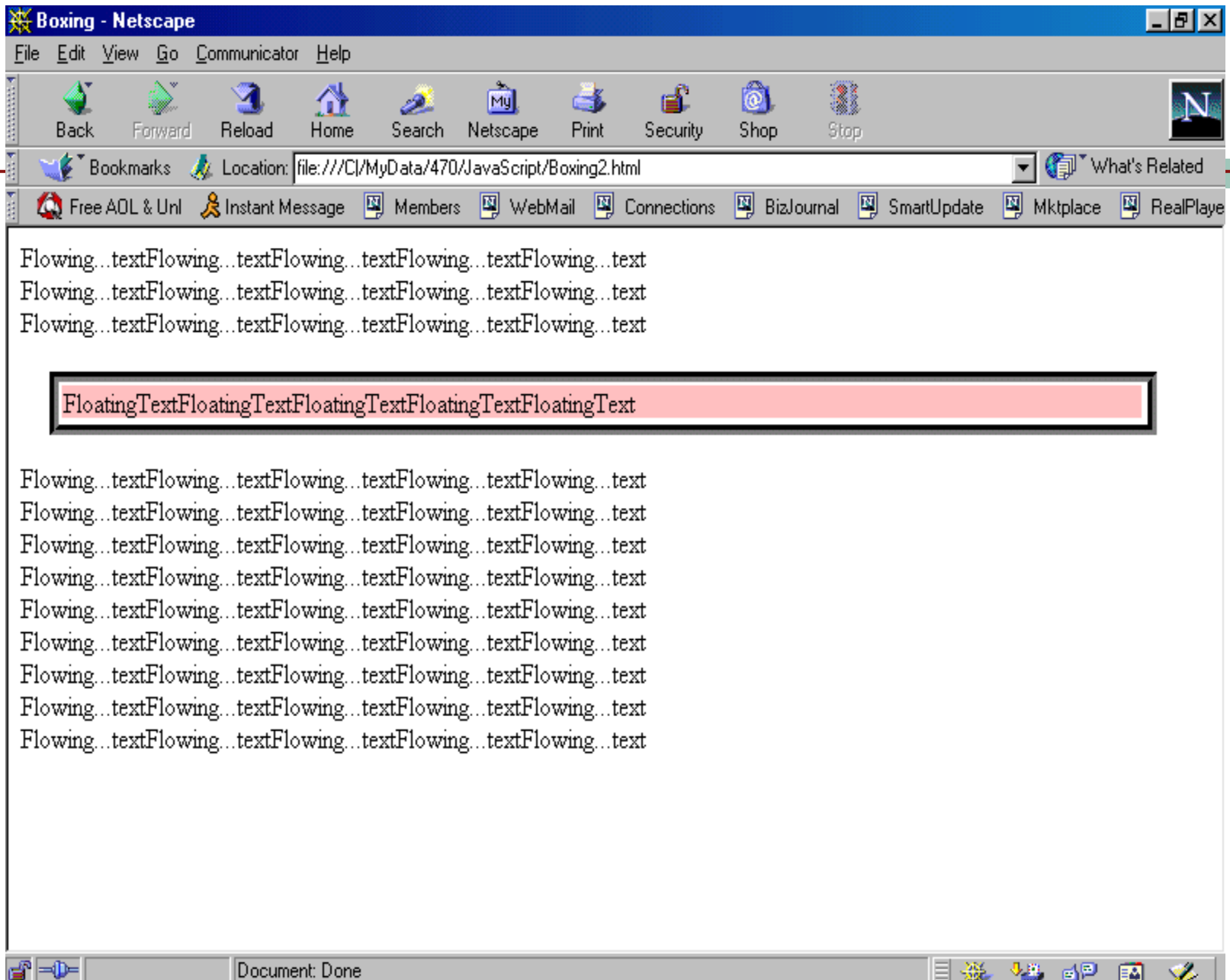


Flowing...textFlowing...textFlowing...textFlowing...textFlowing...text
Flowing...textFlowing...textFlowing...textFlowing...textFlowing...text
Flowing...textFlowing...textFlowing...textFlowing...textFlowing...text

FloatingTextFloatingTextFloatingTextFloatingTextFloatingText
FloatingTextFloatingTextFloatingTextFloatingTextFloatingText

[illegible]

[illegible]



“Cascading”

- Style sheets are called “cascading” since a local style attribute will override/extend an internal style sheet which will override/extend an external style sheet (which will “perhaps” override/extend a user defined style)
- The style defined “closest” to the element will prevail



Styling Lists with CSS

- To change the type of list marker or to prevent any display of a list marker, use

`list-style-type: type;`

where *type* is the various types of markers

list-style-type	Marker(s)
disc	●
circle	○
square	■
decimal	1, 2, 3, 4, ...
decimal-leading-zero	01, 02, 03, 04, ...
lower-roman	i, ii, iii, iv, ...
upper-roman	I, II, III, IV, ...
lower-alpha	a, b, c, d, ...
upper-alpha	A, B, C, D, ...
lower-greek	α, β, γ, δ, ...
upper-greek	A, B, Γ, Δ, ...
none	no marker displayed

Lists (con't)

- Nested lists can be displayed in an outline style using contextual selectors
- The following style rules create an outline style for a nested ordered list:

```
ol {list-style-type: upper-roman;}  
ol ol {list-style-type: upper-alpha;}  
ol ol ol {list-style-type: decimal;}
```

- A customized graphic image for the list marker can be supplied by the user

```
list-style-image: url (url);
```

where *url* is the URL of a graphic file containing the marker image

Lists (con't)

- CSS treats each list item as a block-level element, placed within a virtual box in which the list marker is placed outside of the list text
- To change the default behaviour, use
`list-style-position: position;`
where *position* is either outside or inside

CSS Display Properties

```
display: none  
  
display: inline  
display: block  
display: list-item  
display: inline-block  
display: inline-table  
display: table  
display: table-cell  
display: table-column  
display: table-column-group  
display: table-footer-group  
display: table-header-group  
display: table-row  
display: table-row-group  
display: flex  
display: inline-flex  
display: grid  
display: inline-grid  
display: ruby  
display: ruby-base  
display: ruby-text  
display: ruby-base-container  
display: ruby-text-container  
display: run-in  
  
display: inherit  
display: initial  
display: unset
```

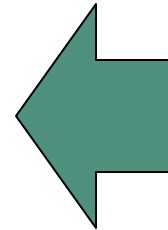


Table
Emulation

Using CSS for Layout of Tables

Table model values (CSS 2.1)	table	These elements behaves like <code><table></code> HTML elements. It defines a block-level box.
	table-caption	These elements behave like <code><caption></code> HTML elements.
	table-cell	These elements behave like <code><td></code> HTML elements.
	table-column	These elements behave like <code><col></code> HTML elements.
	table-column-group	These elements behave like <code><colgroup></code> HTML elements.
	table-footer-group	These elements behave like <code><tfoot></code> HTML elements.
	table-header-group	These elements behave like <code><thead></code> HTML elements.
	table-row	These elements behave like <code><tr></code> HTML elements.
	table-row-group	These elements behave like <code><tbody></code> HTML elements

Using CSS for Layout Tables (con't)

Cyber Incidents



Denial of Service



Malware



Unauthorized Access



Inappropriate Usage

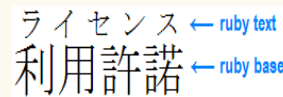
What is CSS ruby ?

Ruby is the commonly-used name for a run of text that appears alongside another run of text (referred to as the "base") and serves as an annotation or a pronunciation guide associated with that run of text.

The following figures show two examples of Ruby, a simple case and one with more complicated structure.

EXAMPLE 1

In this first example, a single annotation is used to annotate the base text.



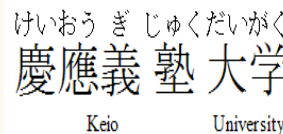
ライセンス ← ruby text
利用許諾 ← ruby base

Figure 1. Example of ruby used in Japanese (simple case)

In Japanese typography, this case is sometimes called taigo ruby or group-ruby (per-word ruby), because the annotation as a whole is associated with multi-character word (as a whole).

EXAMPLE 2

In this second example, two levels of annotations are attached to a base sequence: the hiragana characters on top refer to the pronunciation of each of the base kanji characters, while the words "Keio" and "University" on the bottom are annotations describing the English translation.



けいおう ぎ じゅくだいがく
慶應義塾 大学
Keio University

Figure 2. Complex ruby with annotation text over and under the base characters

Generating Content with CSS

- New content can be added either before or after an element using the following `before` and `after` pseudo-elements:

```
element::before {content: text;
```

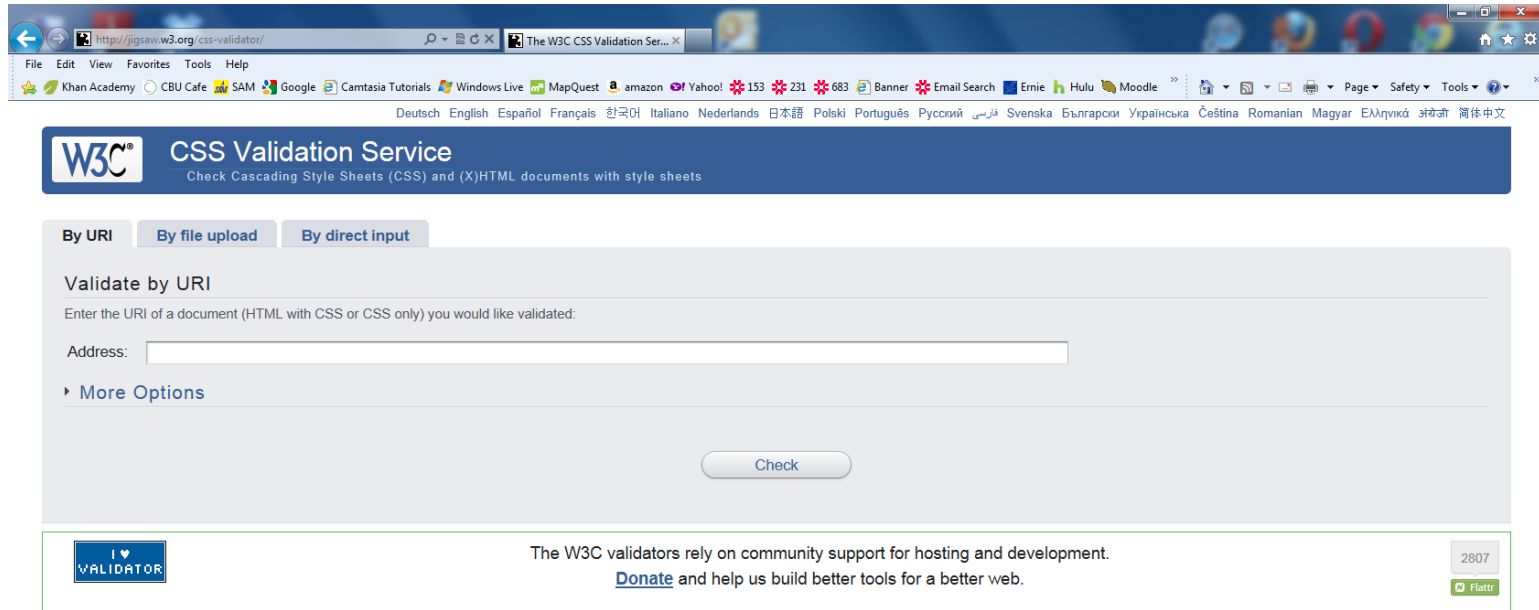
```
element::after {content: text;
```

where *text* is the content to be inserted into the rendered web page

Value	Description
<code>none</code>	Sets the content to an empty text string
<code>counter</code>	Displays a counter value
<code>attr(<i>attribute</i>)</code>	Displays the value of the selector's <i>attribute</i>
<code>text</code>	Displays the specified <i>text</i>
<code>open-quote</code>	Displays an opening quotation mark
<code>close-quote</code>	Displays a closing quotation mark
<code>no-open-quote</code>	Removes an opening quotation mark, if previously specified
<code>no-close-quote</code>	Removes a closing quotation mark, if previously specified
<code>url(<i>url</i>)</code>	Displays the content of the media (image, video, etc.) from the file located at <i>url</i>

CSS Validation

[<http://jigsaw.w3.org/css-validator/>]



The screenshot shows the W3C CSS Validation Service web interface. The browser's address bar displays the URL <http://jigsaw.w3.org/css-validator/>. The page has a blue header with the W3C logo and the text "CSS Validation Service" and "Check Cascading Style Sheets (CSS) and (X)HTML documents with style sheets". Below the header, there are three tabs: "By URI", "By file upload", and "By direct input". The "By URI" tab is selected, and the "Validate by URI" section is active. It contains a text input field labeled "Address:" and a "Check" button. Below the input field, there is a link for "More Options". At the bottom of the page, there is a footer with a "I ♥ VALIDATOR" logo, a message about community support, a "Donate" link, and a "Flattr" button.

W3C[®] CSS Validation Service
Check Cascading Style Sheets (CSS) and (X)HTML documents with style sheets

By URI By file upload By direct input

Validate by URI

Enter the URI of a document (HTML with CSS or CSS only) you would like validated:

Address:

▸ More Options

Check

I ♥ VALIDATOR

The W3C validators rely on community support for hosting and development.
[Donate](#) and help us build better tools for a better web.

2807
Flattr

Note: If you want to validate your CSS style sheet embedded in an (X)HTML document, you should first [check that the \(X\)HTML you use is valid](#).

[About](#) [Documentation](#) [Download](#) [Feedback](#) [Credits](#)



COPYRIGHT © 1994-2009 W3C[®] (MIT, ERCIM, KEIO). ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.



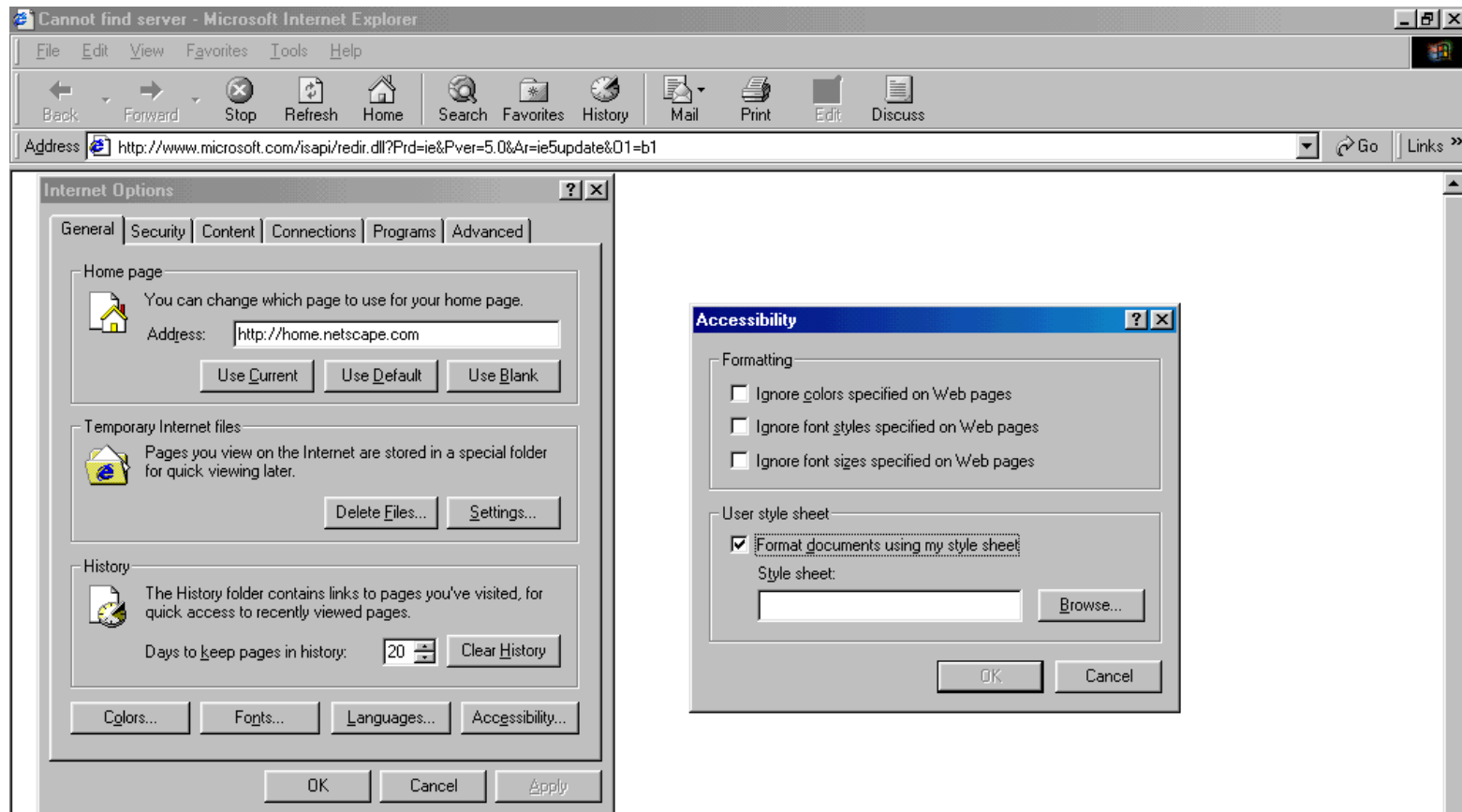
Types of Style Sheets

- **Browser styles** or **user agent styles** – Styles built into the browser
- **User-defined styles** – Styles defined by a user based on the configuration settings of the user's browser
- **External styles** – Styles created by a website author, placed within a CSS file, and linked to the website
- **Embedded styles** – Styles added to the head of an HTML document
- **Inline styles** – Styles added as element attributes within an HTML document and applied to only that particular element

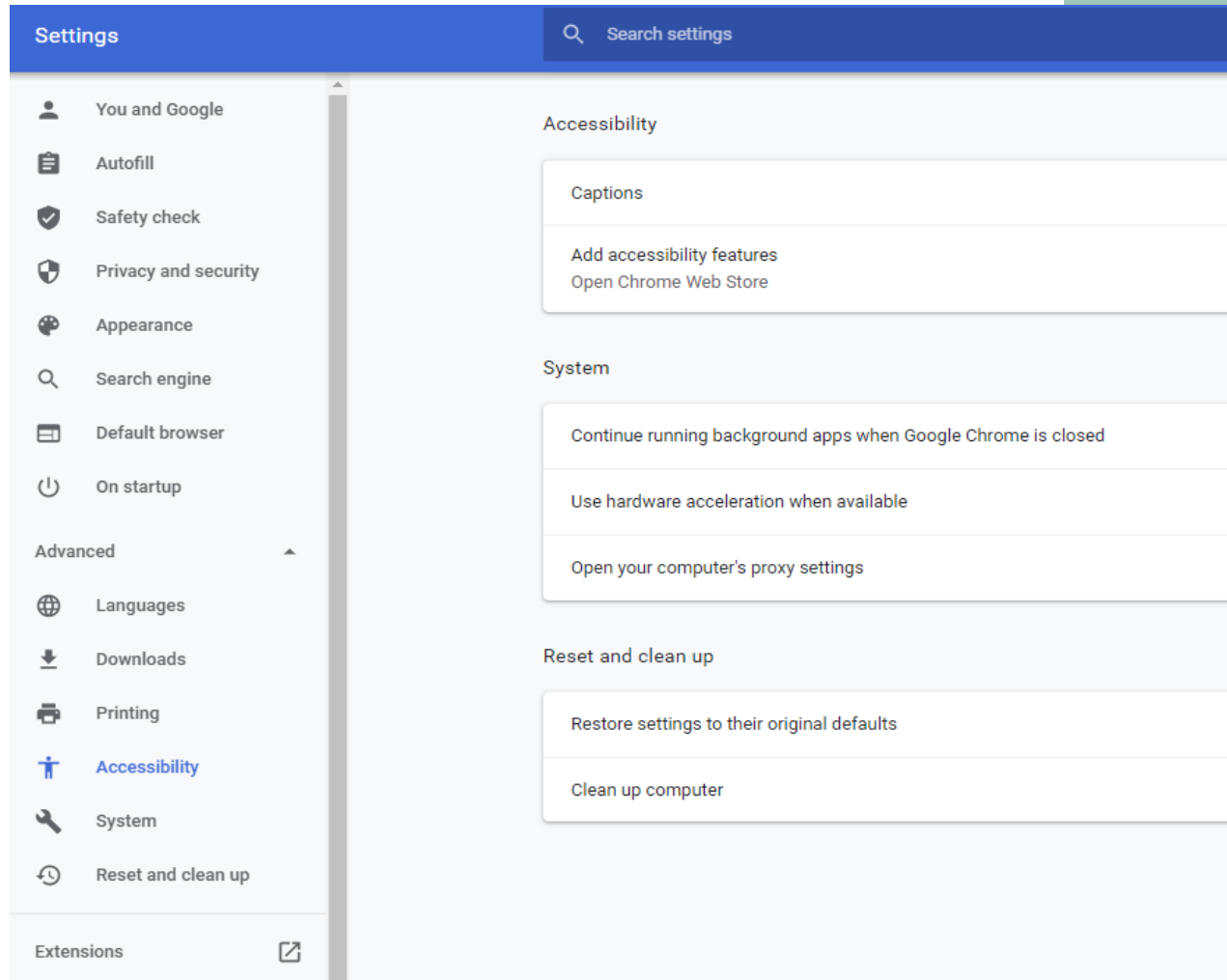
User Defined Style Sheets

- Browsers give users the capability to set their own fonts, colors, and other styles including turning off some dynamic settings
- When setting up your style specifications for your content:
 - It is best to use relative measures of size (em or ex), instead of absolute measures (such as points or pixels), so you do not override the users need for certain styles
 - The latest browsers that have moved from CSS1 to CSS2/3 do not override the user style settings, but earlier browsers based on CSS1 do

IE – Setting User Style Sheet (Tools->Internet Options->Advanced)



Chrome -> Settings -> Advanced -> Accessibility



Browser/Agent Extensions

- **Browser extensions** are an extended library of style properties in the browser
- **Vendor prefix** – Indicates the browser vendor that created and supports the style property
- One can directly link or import in these style

Vendor Prefix	Rendering Engine	Browsers
<u>-Khtml-</u>	<u>KHTML</u>	<u>Konqueror</u>
<u>-moz</u>	Mozilla	Firefox, Camino
<u>-ms-</u>	Trident	Internet Explorer
<u>-o-</u>	Presto	Opera, Nintendo Wii browser
<u>-webkit-</u>	<u>WebKit</u>	Android browser, Chrome, Safari

Importing Style Sheets

- Importing style sheets
- `@import` is a CSS at-rule used to import the content of a style sheet file

```
@import url(url);
```

where `url` is the URL of an external stylesheet file

- It is similar to adding link elements to an HTML file

CSS Reset

- Each browser/agent has its own defaults for specific aspects such as line width and height
- These default rules sometimes conflict with custom CSS rules
- To avoid this problem particularly with older browsers, web authors often have a “CSS reset” section in their CSS file
- More on this later in the course

Example CSS Reset Code

```
■ html, body, div, span, applet, object, iframe,  
■ h1, h2, h3, h4, h5, h6, p, blockquote, pre,  
■ a, abbr, acronym, address, big, cite, code,  
■ del, dfn, em, img, ins, kbd, q, s, samp,  
■ small, strike, strong, sub, sup, tt, var,  
■ b, u, i, center,  
■ dl, dt, dd, ol, ul, li,  
■ fieldset, form, label, legend,  
■ table, caption, tbody, tfoot, thead, tr, th, td,  
■ article, aside, canvas, details, embed,  
■ figure, figcaption, footer, header, hgroup,  
■ menu, nav, output, ruby, section, summary,  
■ time, mark, audio, video {  
■     margin: 0;  
■     padding: 0;  
■     border: 0;  
■     font-size: 100%;  
■     font: inherit;  
■     vertical-align: baseline;  
■ }  
■  
■ /* HTML5 display-role reset for older browsers */  
■ article, aside, details, figcaption, figure,  
■ footer, header, hgroup, menu, nav, section {display: block;}  
■ body {line-height: 1;}  
■ ol, ul {list-style: none;}  
■ blockquote, q {quotes: none;}  
■ blockquote:before, blockquote:after, q:before, q:after {content: ""; content: none;}  
■ table {border-collapse: collapse; border-spacing: 0;}
```

Common CSS Properties

Font Properties

- [Font Family](#)
- [Font Style](#)
- [Font Variant](#)
- [Font Weight](#)
- [Font Size](#)
- [Font](#)

Color and Background Properties

- [Color](#)
- [Background Color](#)
- [Background Image](#)
- [Background Repeat](#)
- [Background Attachment](#)
- [Background Position](#)
- [Background](#)

Text Properties

- [Word Spacing](#)
- [Letter Spacing](#)
- [Text Decoration](#)
- [Vertical Alignment](#)
- [Text Transformation](#)
- [Text Alignment](#)
- [Text Indentation](#)
- [Line Height](#)

CSS Properties (con't)

Box Properties

- [Top Margin](#)
- [Right Margin](#)
- [Bottom Margin](#)
- [Left Margin](#)
- [Margin](#)
- [Top Padding](#)
- [Right Padding](#)
- [Bottom Padding](#)
- [Left Padding](#)
- [Padding](#)
- [Top Border Width](#)
- [Right Border Width](#)
- [Bottom Border Width](#)
- [Left Border Width](#)
- [Border Width](#)
- [Border Color](#)
- [Border Style](#)
- [Top Border](#)
- [Right Border](#)
- [Bottom Border](#)
- [Left Border](#)
- [Border](#)
- [Width](#)
- [Height](#)
- [Float](#)
- [Clear](#)

Classification Properties

- [Display](#)
- [Whitespace](#)
- [List Style Type](#)
- [List Style Image](#)
- [List Style Position](#)
- [List Style](#)

Units

- [Length Units](#)
- [Percentage Units](#)
- [Color Units](#)
- [URLs](#)

CSS Property Index

[<http://www.bloobery.com/indexdot/css/propindex/all.htm>]

CSS Property Index

= Index DOT Css by Brian Wilson =

[Main Index](#) | [Property Index](#) | [CSS Support History](#) | [Browser History](#)

[All](#) | [Fonts](#) | [Text](#) | [Colors/Backgrounds](#) | [Scrollbars](#) | [Outlines](#) | [Lists](#) | [Tables](#)
[Classification](#) | [Dimensions](#) | [Positioning](#) | [Margins](#) | [Padding](#) | [Borders](#)
[Dynamic Content](#) | [Generated Content](#) | [International](#) | [Printing](#) | [Aural](#) | [Extensions](#)

A/B	accelerator azimuth background background-attachment background-color background-image background-position background-position-x background-position-y background-repeat behavior	border border-bottom border-bottom-color border-bottom-style border-bottom-width border-collapse border-color border-left border-left-color border-left-style border-left-width	border-right border-right-color border-right-style border-right-width border-spacing border-style border-top border-top-color border-top-style border-top-width border-width bottom
C	caption-side clear clip color	content counter-increment counter-reset	cue cue-after cue-before cursor
D/E	direction display	elevation empty-cells	
F	filter float	font font-family font-size font-size-adjust	font-stretch font-style font-variant font-weight

Internet 100%

W3schools – CSS

[https://www.w3schools.com/html/html_css.asp]

The screenshot displays the W3schools website interface. At the top, the logo 'w3schools.com' is on the left, and 'THE WORLD'S LARGEST' is on the right. A navigation bar contains links for Home, HTML (highlighted), CSS, JavaScript, SQL, Python, PHP, Bootstrap, How To, More, References, and Experiments. A left sidebar lists various HTML topics, with 'HTML CSS' highlighted in green. The main content area features a red banner for State Farm with the text 'Here to help life go right.' and a 'Get A Quote' button. Below this is the title 'HTML Styles - CSS' with 'Previous' and 'Next' navigation buttons. A graphic shows overlapping colored boxes (blue, green, orange) with the text 'CSS = Styles and Colors' and 'Manipulate Text' below it. The section 'Styling HTML with CSS' follows, with a definition: 'CSS stands for Cascading Style Sheets.' and a description: 'CSS describes how HTML elements are to be displayed on screen, paper, or in other media.'

w3schools.com THE WORLD'S LARGEST

Home HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP HOW TO MORE REFERENCES EXPER

HTML Tutorial

- HTML HOME
- HTML Introduction
- HTML Editors
- HTML Basic
- HTML Elements
- HTML Attributes
- HTML Headings
- HTML Paragraphs
- HTML Styles
- HTML Formatting
- HTML Quotations
- HTML Comments
- HTML Colors
- HTML CSS**
- HTML Links
- HTML Images
- HTML Tables
- HTML Lists
- HTML Blocks
- HTML Classes
- HTML Id
- HTML Iframes
- HTML JavaScript
- HTML File Paths
- HTML Head
- HTML Layout
- HTML Responsive

Here to help life go right. Get A Quote State Farm

HTML Styles - CSS

< Previous Next >

CSS = Styles and Colors

Manipulate Text

Colors, Boxes

Styling HTML with CSS

CSS stands for Cascading Style Sheets.

CSS describes **how HTML elements are to be displayed on screen, paper, or in other media.**

References

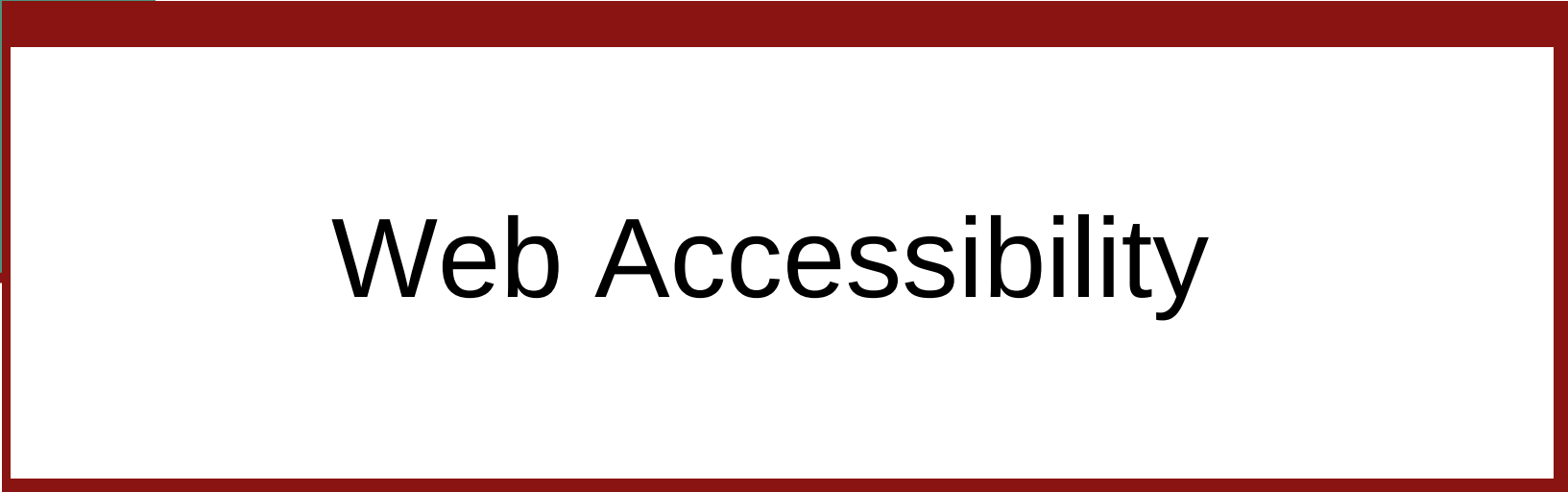
- CSS: The Definitive Guide: Visual Presentation for the Web by Eric A. Meyer and Estelle Weyl
- CSS: The Missing Manual by David Sawyer McFarland
- Cascading Style Sheets: Designing for the Web, Portable Documents by Hakon Wium Lie and Bert Bos
- Cascading Style Sheets (Quick Study Computer) by Inc. BarCharts

Homework

- Textbook Chapter 2 and Appendix A
- Use an external style sheet to define the characteristics of the background, tags, colors, and fonts in your personal web page (project 1)
 - email your .css file to the instructor (as an attachment) as well as the new version of your web page
- Hybrid assignment: appendix on accessibility & textbook appendix D



Appendix



Web Accessibility

Web Accessibility

- In order to reach out to all users, it is necessary to understand that certain users may have some accessibility issues
- About 20% of the US population have some form of disability
 - over 50 million Americans
 - over 1 trillion dollars in purchasing power
- 8 million visually impaired Americans
- www.census.gov/population/www/pop-profile/disabil.html

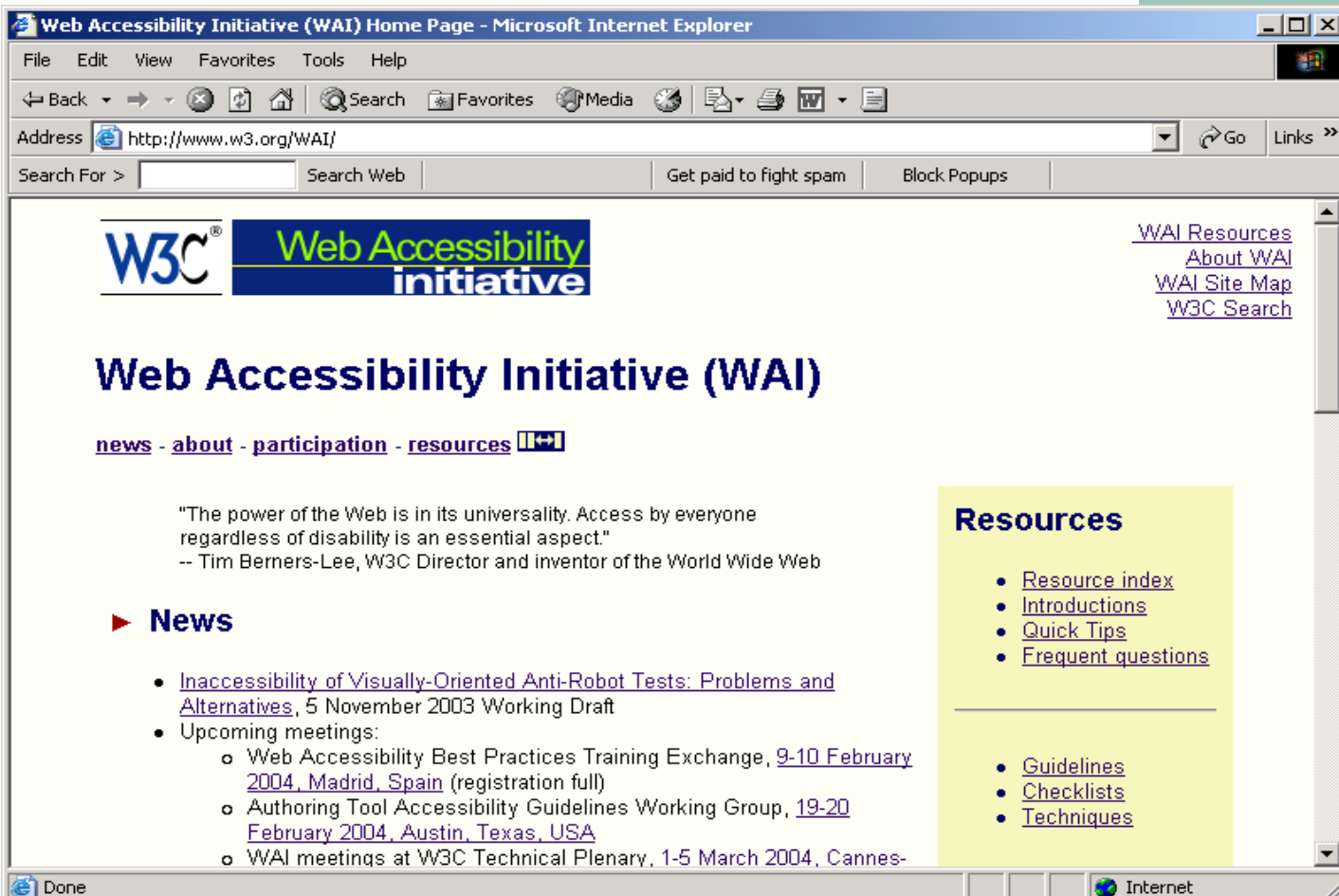
Web Accessibility (con't)

- Common issues here involve people that have:
 - Poor vision
 - Color blindness
 - Total blindness
 - Manual dexterity issues

In 1997 the W3C began its Web Accessibility Initiative (W3C/WAI)

- www.w3.org/wai

WAI



Accessibility Laws

- A U.S. Department of Justice ruling in issued 1996 said that the ADA (Americans with Disabilities Act) applies to Web pages
- Also in 1996, sections 255 and 251 were amended so that communication devices including wireless phones must be accessible to individuals with disabilities
- In 1997, the Individuals with Disabilities Education Act covered educational materials in schools
- In 1999, the National Federation for the Blind sued AOL on the ADA issue

Accessibility Laws (con't)

- In addition, Section 508 of the Rehabilitation Act (as amended) requires that when federal agencies [or federal contractors] develop, procure, maintain, or use IT, they must ensure that it is available to people with disabilities (the effective date of that measure is 6/25/2001)
- In addition Section 508 of the Rehabilitation Act stated that colleges with federal funding (most all colleges) cannot discriminate against individuals with disabilities
- The number of protests and lawsuits thus far in this area is small, but as governments and companies begin to publish work related data on their web sites, it is essential that disabled people also be able to utilize these sites

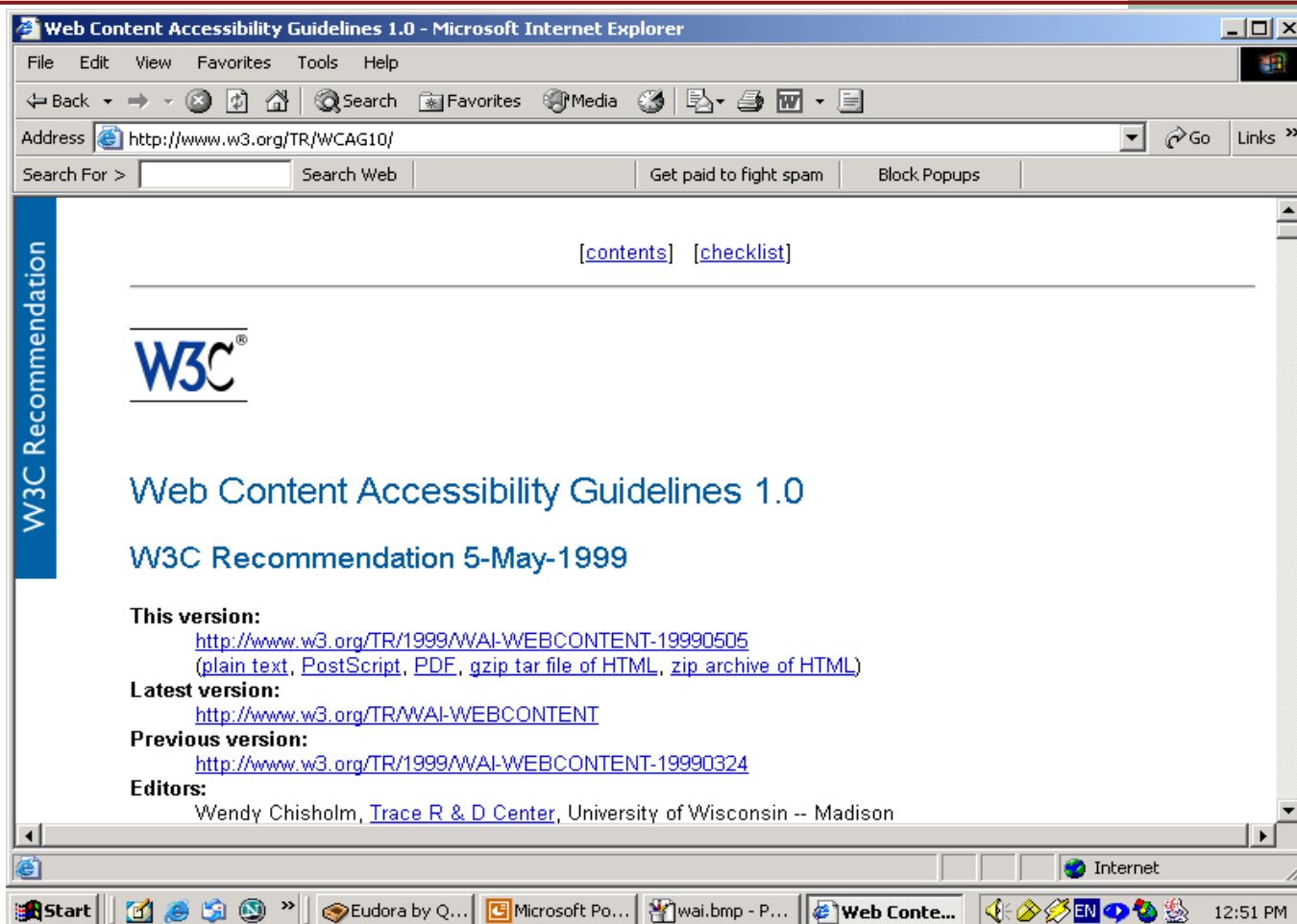
W3C/508 Checkpoints

[www.section508.gov]

- Priority 1 – Web Accessibility Guidelines
 - Web page must provide a text equivalent for every non-text element
 - All information conveyed with color must be available without color
- Priority 2
 - Use of stylesheets
 - Etc...
- Priority 3

See www.w3.org/TR/WCAG10

Accessibility Guidelines



Guideline Areas

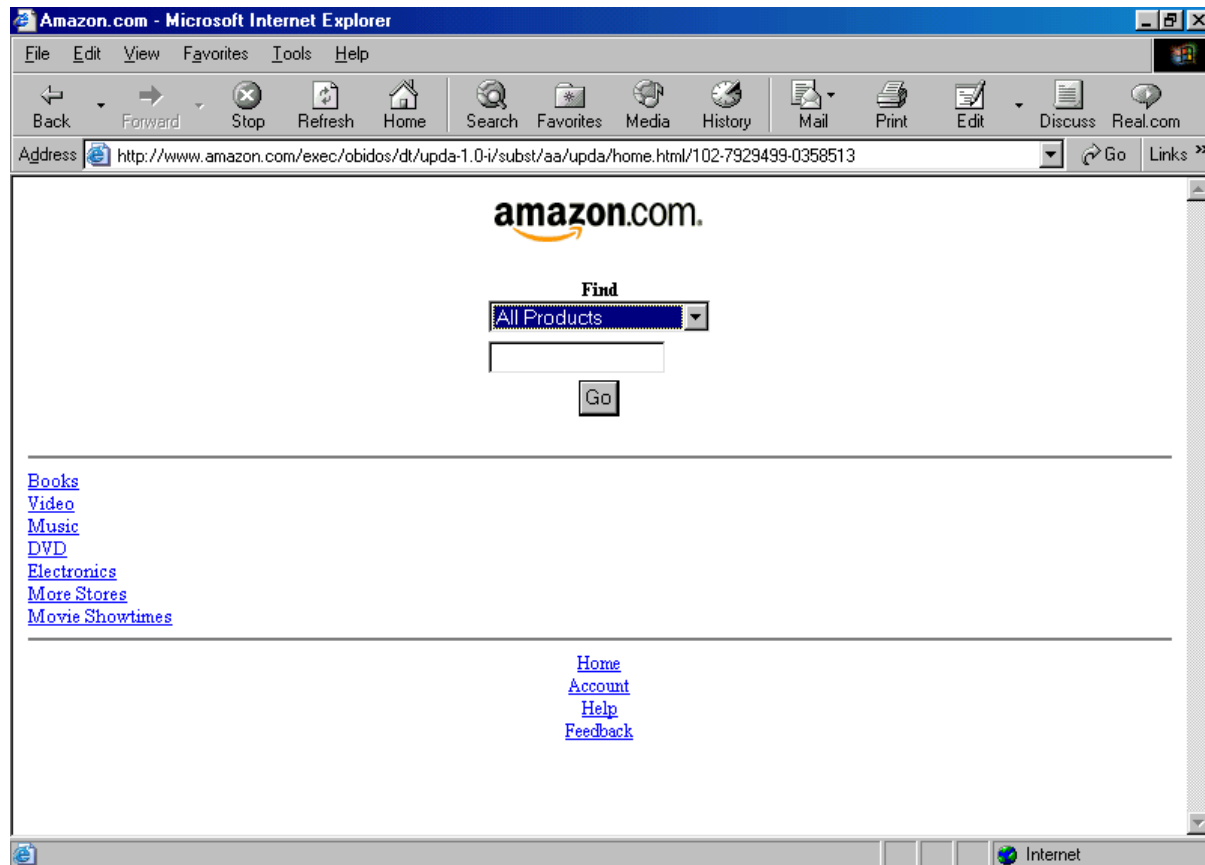
6. Web Content Accessibility Guidelines

1. Provide equivalent alternatives to auditory and visual content.
2. Don't rely on color alone.
3. Use markup and style sheets and do so properly.
4. Clarify natural language usage
5. Create tables that transform gracefully.
6. Ensure that pages featuring new technologies transform gracefully.
7. Ensure user control of time-sensitive content changes.
8. Ensure direct accessibility of embedded user interfaces.
9. Design for device-independence.
10. Use interim solutions.
11. Use W3C technologies and guidelines.
12. Provide context and orientation information.
13. Provide clear navigation mechanisms.
14. Ensure that documents are clear and simple.

Regular Amazon



www.amazon.com/access



Accessibility Tools

- There are a number of accessibility tools on the market:
 - JAWS – Job Access with Sound (www.hj.com/JAWS/JAWS35DemoOp.htm)
 - IBM's ViaVoice (www-4.ibm.com/software/speech)
 - VoiceXML (www.alphaworks.ibm.com)
- These provide Braille translators and printers, audio synthesis (“screen-reader software”), etc.
- These products work best when “translating” from a plain text HTML page
- Use of CSS, JS, DHTML, etc. may not let these products work correctly

Use of LINK tag

- The LINK tag can be used to direct visitors to another page that provides for handicaps (the media attribute lists handicap tools that will work on the alternative page):
 - `<HTML><HEAD>`
 - `<TITLE>...</TITLE>`
 - `<LINK title="Text-only version"`
 - `href="myPage_accessible.html"`
 - `media="aural, Braille, tty">`
 - `</HEAD>`
- Aural is speech synthesis, tty is a teletype terminal
- Newer browsers that support the media shown in the link tag can automatically load the other page

ALT & TITLE Attribute

- “Screen reader” software cannot synthesis images and links
- Use the ALT tag for your images to describe the image
- In the anchor tags, use the TITLE attribute:
 - `<UL>`
 - `<LI><A HREF=“...” TITLE=“This link will take you to a page that shows ...”>Item 1</A>`

Tables

- Tables also cause problems for handicap tools, unless the table is very simple
- Use the available table tags to logically subdivide a table: THEAD, TBODY, TFOOT, COLGROUP, COL
- Add a heading to your table with the CAPTION tag
- Add a summary to your table with the SUMMARY attribute of the TABLE tag

Tabbing

- Many HTML structures can have a “tab index” set for the items within the structure
- This will allow people who have trouble with the mouse to navigate easier
- Forms and list can easily be set up this way by using the tabindex=“i” attribute set up for the form elements or lists items
- In addition for forms, use the logical grouping tags: FIELDSET, LEGEND, OPTGROUP

WAI Guideline 1. Provide equivalent alternatives to auditory and visual content.

1.1 Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). *This includes:* images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ascii art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video. [Priority 1]

- For example, in HTML:
 - Use "alt" for the IMG, INPUT, and APPLET elements, or provide a text equivalent in the content of the OBJECT and APPLET elements.
 - For complex content (e.g., a chart) where the "alt" text does not provide a complete text equivalent, provide an additional description using, for example, "longdesc" with IMG or FRAME, a link inside an OBJECT element, or a [description link](#).
 - For image maps, either use the "alt" attribute with AREA, or use the MAP element with A elements (and other text) as content.
- Refer also to [checkpoint 9.1](#) and [checkpoint 13.10](#).
- [Techniques for checkpoint 1.1](#)

■ **1.2 Provide redundant text links for each active region of a server-side image map. [Priority 1]**

- Refer also to [checkpoint 1.5](#) and [checkpoint 9.1](#).
- [Techniques for checkpoint 1.2](#)

WAI Guideline 1 (con't)

- **1.3 [Until user agents](#) can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation. [Priority 1]**
 - Synchronize the [auditory description](#) with the audio track as per [checkpoint 1.4](#). Refer to [checkpoint 1.1](#) for information about textual equivalents for visual information.
 - [Techniques for checkpoint 1.3](#)
- **1.4 For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation. [Priority 1]**
 - [Techniques for checkpoint 1.4](#)
- **1.5 [Until user agents](#) render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map. [Priority 3]**
 - Refer also to [checkpoint 1.2](#) and [checkpoint 9.1](#).
 - [Techniques for checkpoint 1.5](#)

WAI Guideline 2. Don't rely on color alone.

- **2.1 Ensure that all information conveyed with color is also available without color, for example from context or markup.**
[Priority 1]
 - [Techniques for checkpoint 2.1](#)
- **2.2 Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.**
[Priority 2 for images, Priority 3 for text].