HTML5 and CSS3 Introductory

Formatting Text with CSS

Unit E Part 1

Objectives

- Tag Qualifying
- Assess web fonts
- Declare a font family
- Use a custom font
- Declare font size and line height
- Implement bold and italics
- Style pseudo-elements
- Specify hex and RGB colors
- Add shadows
- Create a media query

How Browsers Read CSS

- It's important as we continue to study CSS to understand how browsers read CSS.
- In English, we read from left to right. However, browsers read the selectors used in CSS from right to left.
- So the selector `header p` is read by the browser as `p` then `header`.
- The selector being targeted by the CSS, in the case above it is the header HTML type selector, is known as the "key selector".

Tag Qualifying

- What is the difference between:
  - `Header p .achievement { … }` and
  - `Header p.achievement { … }`?
- In the first example, we are wanting to select any element with a class name of "achievement" that is a child of any p element that is a child of a header element.
- While in the second example, we are wanting to select any p element that has a class name of "achievement" that is a child of any header element.

Tag Qualifying

- In general, whitespace in HTML and CSS doesn't matter, except in certain circumstances:
  - In the values for ID and class attributes in HTML where spaces between words means multiple ID or class names
  - Inside the selectors in CSS where spaces between selectors creates a descendant selector
- Normally, it is rare to use tag qualifying, but if you are wanting to reuse a class on multiple elements but make minor tweaks based on the element, then tag qualifying is beneficial in that case.

Assess Web Fonts

- Web page font display limited by fonts available on users' machines
- How to implement fonts consistently:
  - Specify multiple font families
    - Font stack is a list of font families in order of preference, separated by commas
  - Specify a generic font family
    - Generic font families is a grouping of font families according to shared characteristics
    - Add to end of font stack

Use Downloadable fonts

- User agent downloads and applies fonts not installed on user's computer
- Upload file containing elements of the font family to web publishing location or reference licensed downloadable font
- Add `@font-face` rule to style sheet
  - Indicates font name and location of necessary files

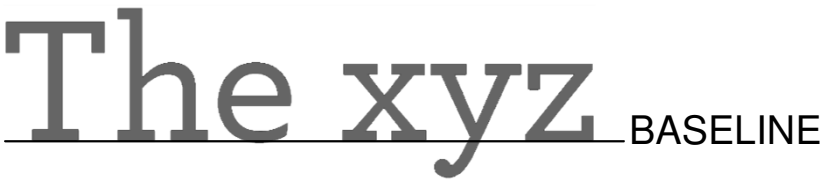
Configure Typeface with CSS

Font Family Category	Font Family Description	Font Typeface Examples
serif	Serif fonts have small embellishments on the end of letter strokes; often used for headings.	Times New Roman, Georgia, Palatino
sans-serif	Sans-serif fonts do not have serifs; often used for web page text.	Arial, Tahoma, Helvetica, Verdana
monospace	Fixed-width font; often used for code samples.	Courier New, Lucida Console
cursive	Hand-written style; use with caution; may be difficult to read on a web page.	Lucida Handwriting, Brush Script, Comic Sans MS
fantasy	Exaggerated style; use with caution; sometimes used for headings; may be difficult to read on a web page.	Jokerman, Impact, Papyrus

Typeface Terminology

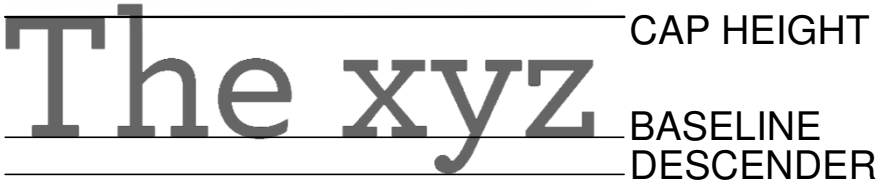
SERIF	SANS-SERIF	MONOSPACE
im	im	im

Typeface Terminology

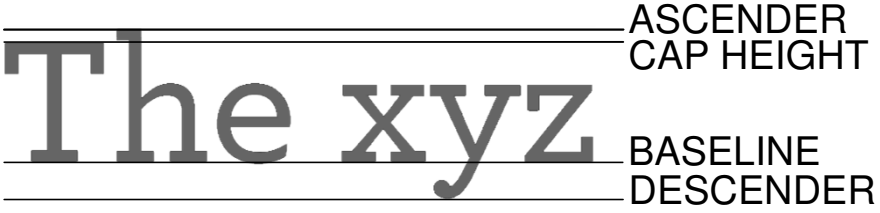


Typeface Terminology

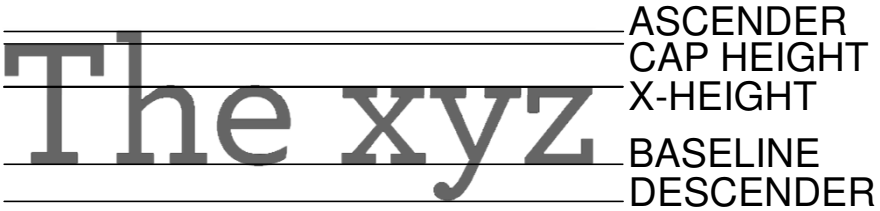
Typeface Terminology



Typeface Terminology



Typeface Terminology



Typeface Terminology

WEIGHT	STYLE	STRETCH
Light	Normal	Condensed
Medium	*Italic*	Regular
Bold	*Oblique*	Extended
Black		

font-family Property

- In CSS there are two types of font family names:
  - Generic – a group of font families with a similar look (like "serif" or "monospace"
  - Font family – a specific font family (like "Times New Roman" or "Arial")
- The font family of a text is set with the font-family property
- You should list several font names as a fallback system incase the computer doesn't have the first font.
- Start with the font you want and end with a generic family
- If the name of a font family is more than one word, it must be in quote marks
- More than one family is specified in a comma-separated list

font-family Property

- Examples:

```
p {
    font-family: "Times New Roman", Times, Georgia, serif;
}

h1 {
    font-family: Arial, Helvetica, Verdana, sans-serif;
}
```

Commonly Used Font Stacks

font type	font stack
wide sans-serif	Verdana, Geneva, sans-serif
narrow sans-serif	Tahoma, Arial, Helvetica, sans-serif
wide serif	Georgia, Palatino, "Palatino Linotype", serif
narrow serif	"Times New Roman", Times, serif
monospace	"Courier New", Courier, monospace

Declare a Font Family

- `font-family` property: Used to specify fonts in CSS
  - Value is font stack that includes font family name(s) and ends with generic font family name
  - Can be included in any style rule to apply to elements selected in that rule
- Test all fonts in your font stack
  - Use browser developer tools to test and not permanently changing the CSS code

Code with font-family Declarations and Display

```
22     border-right: 1px solid black;
23     background-color: aliceblue;
24     font-family: Tahoma, Arial, Helvetica, sans-serif;
25  }
26
27  /* headings */
28  h1, h2 {
29     padding: 0.4em;
30     text-align: center;
31     font-family: "Times New Roman", Times, serif;
32  }
```

Use a Custom Font

- Custom fonts help to
  - Create uniformity
  - Add wider font possibilities
- Available online, e.g.
  - Google.com/fonts
  - Defaults to only the normal version of font
  - Make selections to make other versions, such as bold or italic, available

Use a Custom Font

- Code with link element to custom font

```
15        <script src="modernizr.custom.62074.js"></script>
16        <link href='http://fonts.googleapis.com/css?family=Bitter' rel='stylesheet'
17        type='text/css'>
18        <link rel="stylesheet" href="styles.css">
```

`link` element copied from Google Fonts website

`href` value references an address on a web server

`link` element for font placed before link element for website style sheet

- Custom font added to font stack

```
28  h1, h2 {
29      padding: 0.4em;
30      text-align: center;
31      font-family: Bitter, "Times New Roman", Times, serif;
32  }
```

Declare Font Size

- `font-size` property: used to specify font size of an element
  - Can specify size in many different units
- Useful to specify font-sizes in style sheet to standardize visual display of font size in different browsers
  - Specify `html` element in `px`
  - Specify other elements in `em` or `rem`

font-size Property

- The font-size property sets the size of a font.
- The value can be a text value, in em, px, or pt units, or as a percentage.
- It's strongly recommended that you use em or percentages instead of pixels (px) or points (pt) because it allows users to resize the text within the browser.
- 1em = 16px = 12pt = 100% = medium

font-size Property

Text Values	Em Units	Px Units	Pt Units	Percentage
xx-small	.5 em	8 px	6 pt	50%
x-small	.60 em	11 px	8 pt	60%
small	.75 em	13 px	10 pt	75%
medium	1 em	16 px	12 pt	100%
large	1.15 em	18 px	13.5 pt	110%
x-large	1.5 em	24 px	18 pt	150%
xx-large	2 em	30 px	24 pt	200%

font-size Property

```
body {
 font-family: Georgia, Times, serif;
 font-size: 12px;}

h1 {
 font-size: 200%;}

.credits {
 font-size: 1.3em;}

#important {
 font-size: 16pt;}

footer p {
 font-size: xx-small;}
```

Declare Line Height

- By default, each font family includes blank space above and below every line of text
- Use `line-height` property to set minimum amount of vertical space each line occupies
- Adjusting line-height helps create white space, which is used for visual effect

line-height Property

- The line-height property specifies the height of the line.
- The default line-height in browsers is 1.
- You can use px, pt, numbers, or percentages to adjust the line-height.
- ```
  p.small {
    line-height: 90%;}

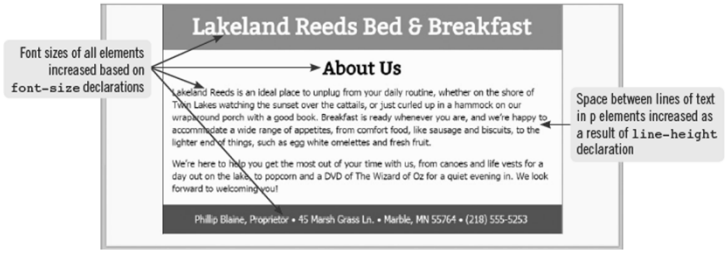
  p.big {
    line-height: 30px;}

  p.big {
    line-height: 2;}
  ```
  ```

Font Size and Line Height Example

```
/* reset styles */
html {
   font-size: 12px;
}
article, body, div, footer, header, h1, h2, p {
p {
   font-size: 1.2em;
   line-height: 1.4em;
}
```

```
h1 {
   color: ivory;
   background-color: darkgoldenrod;
   font-size: 3em;
}
h2 {
   font-size: 2.4em;
}
```



Implement Bold and Italics

- Bold and italics often applied to words or phrases
- Use `span` element to isolate specific sections for formatting:
  - `span` element creates an inline element
    - Does not fill space of parent element
    - Is not rendered with line breaks before or after
    - Assign `class` value to `span` element

font-weight Property

- The font-weight property sets how thick or think characters should be displayed. In other words, how bold should the text be.
- Can take the following values
  - Normal (default), bold, bolder, lighter
  - 100, 200, 300, 400 (same as normal), 500, 600, 700 (same as bold), 800, 900
- p.normal {
  font-weight: normal;}

  p.thick {
  font-weight: bold;}

  p.thicker {
  font-weight: 900;}

font-style Property

- The font-style property specifies the font style for a text.
- The values for font-style are:
  - Normal
  - Italic
  - Oblique
- p.normal {
  font-style: normal;}

  p.italic {
  font-style: italic;}

  p.oblique {
  font-style: oblique;}

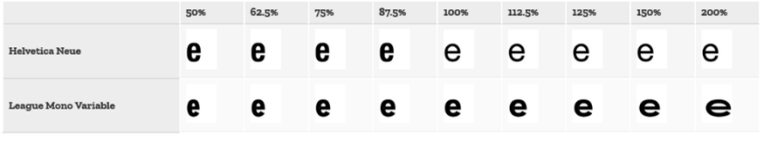
Italic vs. Oblique

- Oblique type is text that slants slightly to the right
- Italic type is text that slants slightly to the right
- However, oblique uses the same glyphs (characters) as the normal font just distorted
- Whereas, italic uses a completely different set of glyphs than the normal font.

The five boxing wizards jump quickly. Normal
The five boxing wizards jump quickly. Italic
The five boxing wizards jump quickly. Oblique

font-stretch Property

- The font-stretch property is used to select whether the normal (default), condensed, or expanded face of a font is used.
- Some font families offer additional faces where the characters are narrower than normal (condensed faces) and others where the characters are wider than normal (expanded faces).
  - font-stretch: ultra-condensed;
  - font-stretch: extra-condensed;
  - font-stretch: condensed;
  - font-stretch: semi-condensed;
  - font-stretch: normal;
  - font-stretch: semi-expanded;
  - font-stretch: expanded;
  - font-stretch: extra-expanded;
  - font-stretch: ultra-expanded;



font-stretch Property

```
<div class="container">
   <p class="condensed">an elephantine lizard</p>
   <p class="normal">an elephantine lizard</p>
   <p class="expanded">an elephantine lizard</p>
</div>
```

```
@font-face {
 src: url('https://mdn.mozillademos.org/files/16014/LeagueMonoVariable.ttf');
 font-family:'LeagueMonoVariable';
 font-style: normal;
}

.container { font: 1.5rem 'LeagueMonoVariable', sans-serif; }
.condensed {  font-stretch: 50%; }
.normal {  font-stretch: 100%; }
.expanded {   font-stretch: 200%; }
```

an elephantine lizard

an elephantine lizard

an elephantine lizard

text-decoration Property

- The text-decoration property is used to specify the decoration added to text.
- The values are none, underline, overline, or line-through.
- a { text-decoration: none; }
- h1 { text-decoration: underline; }
- h3 { text-decoration: overline; }
- .deleted { text-decoration: line-through; }

This is a heading

McLennan Community College

Another heading

~~This is text that was originally included in my post, but I deleted because it was inaccurate.~~

text-indent Property

- The text-indent property configures the indentation of the first line of text in an element (we covered how to indent an entire paragraph when we covered padding and margins.)
- Can accept a value in px, pt, em, and percentages. The value can be a negative number.
- p {
  text-indent: 50px;}

text-transform Property

- The text-transform property configures the capitalization of the text.
- The default is none, which means the text should be rendered as it appears in the code. If you capitalize a letter, then the browser will render that letter as a capital letter.
- Other values you can use are: capitalize, lowercase, and uppercase.
- p.uppercase {
  text-transform: uppercase;}

  p.lowercase {
  text-transform: lowercase;}

  p.capitalize {
  text-transform: capitalize;}

letter-spacing Property

- The letter-spacing property configures the space between text characters.
- The default is normal, however you can specify a numeric pixel or em unit.
- h3 { letter-spacing: normal ;}
- h2 { letter-spacing: 3px; }

This is normal spacing

This has extra letter spacing

CSS Font Properties to Format Text

- `Font-weight` for bold
- `Font-style` for italic

property	description	example
`font-style`	Provides access to a font family's italic or oblique style	`font-style: italic;`
`font-variant`	Provides access to a font family's small caps style	`font-variant: small-caps;`
`font-weight`	Sets the weight of text using keywords or values including `normal`, `bold`, or a multiple of `100` up to `900`, where `100` is lightest and `900` is heaviest	`font-weight: bold;`
`font-size`	Changes an element's font size to a value specified in an applicable unit, including px, %, rem, or em	`font-size: 2em;`
`line-height`	Specifies the height of each line containing text in an applicable unit, including px, %, rem, or em	`line-height: 1.4em;`
`font-family`	Specifies one or more font families and/or a generic font, separated by commas; font family names composed of multiple words must be surrounded by quotes	`font-family: "Times New Roman", Times, serif;`
`font`	Shorthand property that can set all 6 preceding properties in a single declaration, using the syntax [`font-style`] [`font-variant`] [`font-weight`] `font-size` [/`line-height`] `font-family`; at minimum, both `font-size` and `font-family` values must be specified, and all values within square brackets are optional	`font: italic small-caps bold 2em/1.4em "Times New Roman", Times, serif`

Element

- The element is an inline-level element and is used to group inline-elements in a document.
- The element provides a way to add a hook to part of a text or part of a document.
- There is no line break before or after the

Example

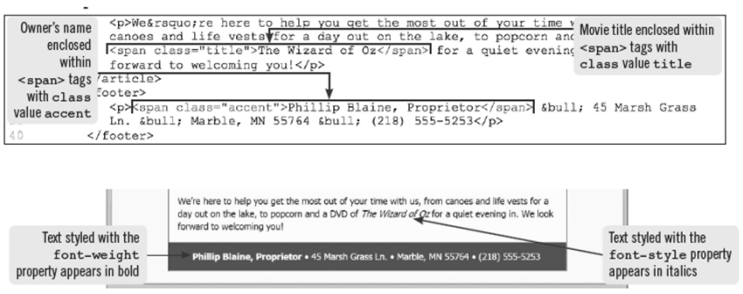
- Embedded CSS:

```
<style>
    .companyname { font-weight: bold;
        font-family: Georgia, "Times New Roman," serif;
        font-size: 1.25em;
    }
</style>
```

- HTML

```
<p>Your needs are important to us at <span class="companyname">Acme Web Design</span>.  We will work with you to build your website.</p>
```

Span Element Example



Style Pseudo-Elements

- Pseudo-element is a selector that enables you to isolate a portion of a larger element for styling
  - Selector format example
    - `article p:first-line`
- Create a drop cap
  - Use `:first-letter` pseudo-element
  - Use `float` property
  - Use `line-height` property

CSS Pseudo-Elements

pseudo-element	effect	properties commonly used with
:first-line	styles the first line of text in the current element	font-family, font-size, font-weight
:first-letter	styles the first letter of text in the current element	font-family, font-size, line-height
:before	inserts specified content before the content of the current element	content
:after	inserts specified content after the content of the current element	content
:selection	styles web page content selected by user	background-color, color

Code for :first-line and :first-letter Pseudo-Elements

```
/* main content */
article {
   background-color: ivory;
}
article p {
   padding: 0 1em 1em;
}
article p:first-line {
   font-weight: bold;
}
article p:first-letter {
   font-size: 3em;
   float: left;
   line-height: 0.9em;
}
.title {
   font-style: italic;
}
```

Pseudo-Elements Rendered in Browser

<blockquote> Element

- The <blockquote> element is used to specify a section is quoted from another source. It should be used for longer quotes.
- It is important to note that you should use the <p> element within the <blockquote> element.
- While not required, it is highly recommended that a <blockquote> element use the *cite* attribute to indicate where the quote is from online.
- Optionally, you can have a citation inside the blockquote. It must be contained inside a footer or cite element.
- Content inside a blockquote other than citations and in-line changes must be quoted from another source.

<blockquote> Element

- Usage example
  - <blockquote> It has been said, 'time heals all wounds.' I do not agree. The wounds remain. In time, the mind, protecting its sanity, covers them with scar tissue and the pain lessens. But it is never gone.
    <footer>Rose Kennedy</footer>
    </blockquote>

cite Attribute

- The *cite* attribute specifies the URL that designates the source of the quotation.
- The syntax for the *cite* attribute is:
  - <blockquote cite="*URL*">
- The accepted values for the *cite* attribute are either an absolute URL or a relative URL.
  - An absolute URL points to another web site
  - A relative URL points to a file within a web site.
- Usage example
  - <blockquote cite="http://www.goodreads.com/quotes/140515-it-has-been-said-time-heals-all-wounds-i-do">…</blockquote>
  - <blockquote cite="index.htm">…</blockquote>

<cite> Element

- The <cite> element is used to define the title of a work (e.g. a book, a song, a movie, a TV show, a painting, a sculpture, etc.)
- It can also be used to list the name of the author (person, people, or organization) or an URL reference for a piece of work.
- Usage example:
  - <cite>Chaos</cite> is the latest book in the Kay Scarpetta series by award-winning novelist, Patricia Cornwell.
  - Battle not with monsters, lest ye become a monster, and if you gaze into the abyss, the abyss gazes also into you.
    <cite>- Friedrich Nietzche</cite>

Complete Blockquote Example

```
<blockquote cite="http://www.quotationspage.com/quote/40436.html">
     <p>Battle not with monsters, lest ye become a monster, and if you gaze into
     the abyss, the abyss gazes also into you.</p>
     <footer>- <cite>Friedrich Nietzche</cite></footer>
</blockquote>
```

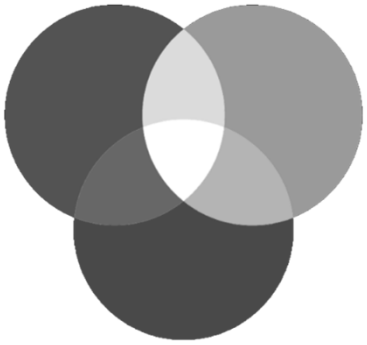
Specify HEX and RGB Colors

- color and background-color properties are used to specify the color of text and other HTML elements
- Four ways to specify colors:
  - color name
  - hexadecimal format
  - rgb format, includes rgba
  - hsl format, includes hsla

CSS Color Systems

system	format	support
name	*name* where *name* is a name from the CSS3 list of 147 colors	all browsers
hexadecimal	`#rrggbb` or `#rgb` where `rr` (or `r`) is the red value, `gg` (or `g`) is the green value, and `bb` (or `b`) is the blue value in hexadecimal absolute value (`00`–`FF` or `0`–`F`)	all browsers
RGB	`rgb(rrr,ggg,bbb)` where `rrr` is the red value, `ggg` is the green value, and `bbb` is the blue value in absolute value (0–255) or percent (0%–100%)	all browsers
RGBa	`rgba(rrr,ggg,bbb,a)` same as RGB but where `a` is a decimal value representing the alpha (transparency); 0 is fully transparent and 1 is fully opaque	modern browsers (not IE8)
HSL	`hsl(hhh,sss%,lll%)` where `hhh` is the hue value in degrees (0–360), `sss` is the saturation value in percent (0%–100%), and `lll` is the light value in percent (0%–100%)	modern browsers (not IE8)
HSLa	`hsla(hhh,sss,lll,a)` same as HSL but where `a` is a decimal value representing the alpha (transparency); 0 is fully transparent and 1 is fully opaque	modern browsers (not IE8)

Understanding Color



Understanding Color

- The colors on your monitor are made by mixing red, green, and blue.
- We've been using hexadecimal values to represent the red, green, and blue values to create color.
- A select group of colors also have HTML color names
- We can also specify color using RGB(a) and HSL(a)

Using Color on Web Pages

- Computer monitors display color as intensities of red, green, and blue light
- RGB Color
- The values of red, green, and blue vary from 0 to 255.
- Hexadecimal numbers (base 16) represent these color values.

Hexadecimal Color Values

- # indicates a hexadecimal value
- Hex value pairs range from 00 to FF
- Three hex value pairs describe an RGB color

Red: #FF0000

Green: #00FF00

Blue: #0000FF

Black: #000000

White: #FFFFFF

Grey: #CCCCCC

RGB Color

- RGB color requires three values:
  - Red color
  - Green color
  - Blue color
- The values for red, green, and blue must be decimal values from 0 to 255.
- Example:
  - h1 { color: rgb(102,205,170); }

RGB VALUES

`rgb(102,205,170)`

RGBA Color

- RGBA color is an extension of the RGB color values by adding in an alpha channel.
- The alpha channel specifies the opacity of the color.
- RGBA color requires four values: Red color, Green color, Blue color, and Alpha (transparency)
- The values for red, green and blue must be decimal values from 0 to 255.
- The alpha value must be a number between 0 (transparent) and 1 (opaque)
- Example:

```
h1 { color: #ffffff;
  color: rgba(255, 255, 255, 0.7); }
```

HSL Color

HUE **SATURATION** **LIGHTNESS**

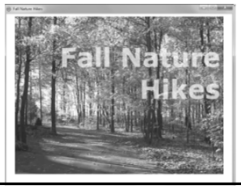
- Hue is a degree on the color wheel (from 0 to 360)
  - 0 (or 360) is red, 120 is green, 240 is blue
- Saturation is a percentage value
  - 0% means a shade of gray and 100% is the full color
- Lightness is also a percentage value
  - 0% is black and 100% is white.

HSL Color

- Using the HSL color, you specify the degree of the hue, the percentage of saturation, and the percentage of lightness.
- Example:
  - h1 {background-color: hsl(120, 100%, 50%);}
- **THIS IS MY HEADING**

HSLA Color

- HSLA color is an extension of the HSL color values by adding in an alpha channel.
- The alpha channel specifies the opacity of the color.
- HSLA color requires four values: degree of the hue, percentage of saturation, percentage of lightness, and alpha (transparency)
- The value for hue must be a value between 0 and 360 while the values for saturation and lightness must be a percent between 0% and 100%.
- The alpha value must be a number between 0 (transparent) and 1 (opaque)
- Example:
  h1 {background-color: hsla(120, 100%, 100%, 0.7);}



Code Using Hex and RGB Colors

```
/* body */
body {
   max-width: 640px;
   margin: 0 auto;
   border-left: 1px solid black;
   border-right: 1px solid black;
   background-color: #dee9f9;
   font-family: Tahoma, Arial, Helvetica, sans-serif;
}

/* headings */
h1, h2 {
   padding: 0.4em;
   text-align: center;
   font-family: Bitter, "Times New Roman", Times, serif;
}
h1 {
   color: ivory;
   background-color: rgb(184,140,77);
   font-size: 3em;
}

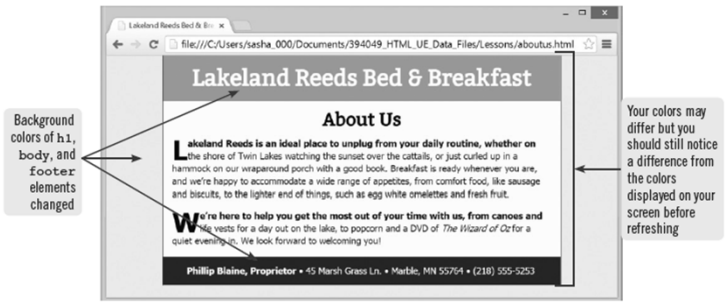
/* footer section */
footer {
   padding: 0.6em;
   color: ivory;
   background-color: rgb(52,24,15);
   text-align: center;
}
```

Hex value where first two characters specify the red value, next two the green value, and last two the blue value

RGB value where the first number specifies the red value, the next one the green value, and the last one the blue value

OK to use different color schemes in the same style sheet

Hex and RGB Color Codes Rendered in Browser

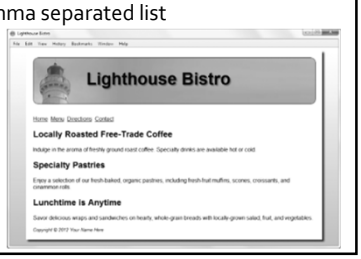


Add Shadows

- `text-shadow`: creates the appearance of a shadow behind text
- `box-shadow`: creates the appearance of a shadow behind an element
- `text-shadow` and `box-shadow` both
  - Support four values: horizontal offset, vertical offset, blur, and shadow color
- `box-shadow` also supports
  - Spread distance and inset

box-shadow Property

- The box-shadow property attaches one or more drop-shadows to the box.
- Required values are the position of the horizontal shadow and the position of the vertical shadow.
- Optional values include the blur distance, the size of the shadow (spread), the color of the shadow, and whether the shadow is outset (default) or inset.
- You can set multiple shadows on a box with a comma separated list
- Example:
  - div {
    box-shadow: 10px 10px 5px 10px #888888;
    }



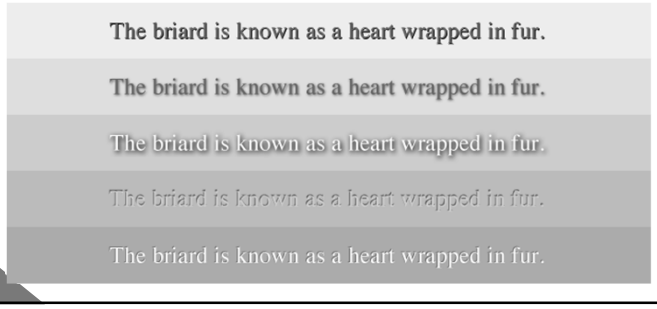
text-shadow Property

- The text-shadow property adds shadow to your text.
- You can have multiple shadows applied to a single piece of text using a comma separated list.
- Required values are the position of the horizontal shadow and the position of the vertical shadow.
- Optional values include the blur radius and the color of the shadow.
- Example:
  - div { text-shadow: 0px 0px 4px #FFF, 0px -5px 4px #FF3, 2px -10px 6px #FD3, -2px -15px 11px #F80, 2px -25px 18px #F20; }

Multiple shadows are Hot

text-shadow Property

- Examples of what can be accomplished with the text-shadow property



Syntax of text-shadow and box-shadow Properties

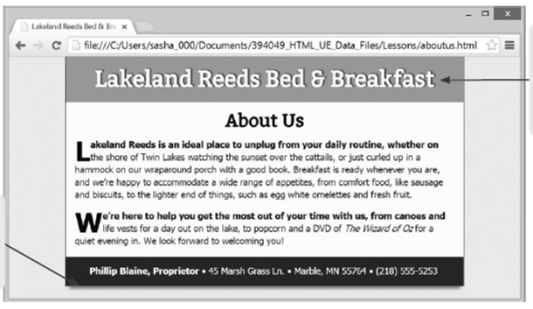
property	value	affects	notes
`text-shadow` and `box-shadow`	horizontal offset	location of shadow horizontally behind text or box	Required value; must be the first number in the list; positive value offsets shadow to the right, and negative value offsets shadow to the left
	vertical offset	location of shadow vertically behind text or box	Required value; must be the second number in the list; positive value offsets the shadow down, and negative value offsets the shadow up
	blur radius	blurriness of shadow	Optional value; must be positive; must be third number in list; value of 0 creates a shadow with a sharp edge
	color	color of shadow behind text or box	Optional value; may appear before or after numerical settings
`box-shadow` only	spread distance	expansion or contraction of shadow	Optional value; must be the fourth number in the list; positive value expands shadow by the specified value, and negative value contracts the shadow
	inset	whether shadow is displayed outside or inside of border	Optional `inset` keyword makes the shadow display inside the element border; may appear before or after other settings

Examples of text-shadow and box-shadow

Browser Example of text-shadow and box-shadow

```
26    background-color: #dee9f9;
27    font-family: Tahoma, Arial, Helvetica, sans-serif;
28    box-shadow: 0 10px 6px -6px #777;
29  }
30  p {
```

```
41  h1 {
42    color: ivory;
43    background-color: rgb(184,148,77);
44    font-size: 3em;
45    text-shadow: 2px -1px 2px rgb(52,24,15);
46  }
```



Heading text formatted with dark brown shadow that extends above it and to the right

Box shadow added to bottom of body element to create raised appearance

Create a Media Query

- Media queries: used to create a group of rules for a specific device
  - Starts with `@media`
  - Lists one or more values for media type
- Media type values for media queries

media type	intended use
all	All devices
print	Printed output and print preview on a screen
screen	Computer screen
speech	Screen reader

@media At-Rule

- The @media CSS at-rule associates a set of nested statements with a condition defined by a media query in a CSS block that is delimited by curly braces
- A media query is composed of a optional media type and/or a number of media features
- For now we are going to focus on just media types
  - all
  - print
  - screen
  - speech
- Since we haven't specified a media type for all of the CSS we have used in class to date, they apply to all media

Print Styling Best Practices

- Hide non-essential content
  - nav { display: none; }
- Configure font size and color for printing
  - Use pt font sizes and use dark text color
- Control page breaks
  - .newpage { page-break-before: always; }

page-break-before Property

- The page-break-before property is used to specify whether a page break should occur BEFORE a specified element.
- The property cannot be used on an empty div or on absolutely positioned elements.
- Use values of auto (default), always, or avoid.
  - auto lets the browser determine page breaks
  - always means to always insert a page break before the element
  - avoid means to avoid a page break before the element, if possible

page-break-after Property

- The page-break-after property is used to specify whether a page break should occur AFTER a specified element.
- The property cannot be used on an empty div or on absolutely positioned elements.
- Use values of auto (default), always, or avoid.
  - auto lets the browser determine page breaks
  - always means to always insert a page break after the element
  - avoid means to avoid a page break after the element, if possible

page-break-inside Property

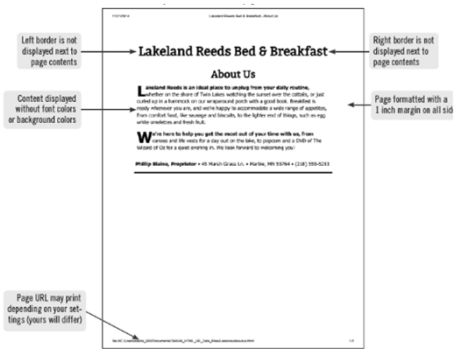
- The page-break-inside property is used to specify whether a page break is allowed inside a specified element.
- The property cannot be used on absolutely positioned elements.
- Use values of auto (default), always, or avoid.
  - auto lets the browser determine page breaks
  - avoid means to avoid page break inside the element, if possible

Media Query for Printed Output

```
/* print styles */
@media print {
   body, h1, article, footer {
      color: #000;
      background-color: #fff;
   }
   body {
      max-width: 100%;
      border: 0;
   }
}
@page {
   margin: 1in;
}
```

Style rules indented within curly braces for media query

Printed Output Based on Media Query



Summary

- It is important to recognize how spaces impact the meaning in CSS selectors and when to use tag qualifying
- It is important to implement fonts so that they will be displayed consistently on different user agents
- Font stacks are declared using the `font-family` property
- Use custom fonts to widen font possibilities and to create uniformity
- Different aspects of font can be set using CSS properties, such as:
  - `font-size` property: set the font size
  - `font-weight` property: make font bold
  - `font-style` property: make font italic

Summary

- Use `line-height` property to create space above and below every line of text
- It is important to implement fonts so that they will be displayed consistently on different user agents
- Font stacks are declared using the `font-family` property
- `span` element: generic element used to isolate a portion of another element
  - Used to create an inline element

Summary

- Use custom fonts to widen font possibilities and to create uniformity
- Different aspects of font can be set using CSS properties, such as:
  - `font-size` property: set the font size
  - `font-weight` property: make font bold
  - `font-style` property: make font italic
  - `Font-stretch` property: use a condensed or expanded face instead of the normal face
- Use `line-height` property to create space above and below every line of text

Summary

- `span` element: generic element used to isolate a portion of another element
  - Used to create an inline element
  - Used to apply bold/italic to text specified by `span` element
- Pseudo-elements to style a portion of an element
- `:before` and `:after` allow you to generate repeated content from style rules, without changing HTML code

Summary

- In CSS, colors specified by name, `rgb/rgba`, `hex`, or `hsl/hsla` value
  - color property: sets font color
  - background property: sets background color
- `text-shadow` and `box-shadow` properties used to create shadows
  - both support: horizontal/vertical offset, blur radius, color
  - `box-shadow` only supports: spread distance, inset
- Media query specifies style rules for a specific device
  - Starts with `@media`
  - Supports media type values: `all`, `print`, `screen`, and `speech`

Questions?


Lab Time