

CSE 154

LECTURE 2: MORE HTML AND CSS



Links: `<a>`

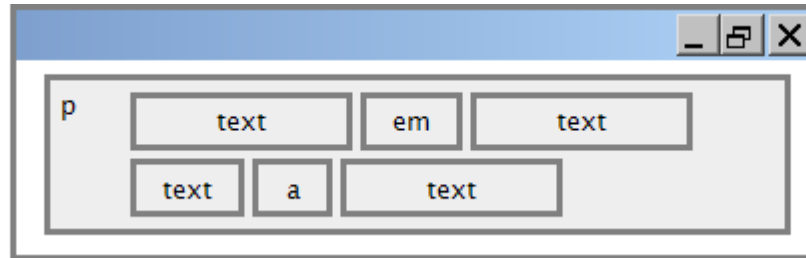
links, or "anchors", to other pages (inline)

```
<p>  
  Search  
  <a href="http://www.google.com/">Google</a> or our  
  <a href="lectures.html">Lecture Notes</a>.  
</p>
```

Search Google or our Lecture Notes.

- uses the href attribute to specify the destination URL
 - can be absolute (to another web site) or relative (to another page on this site)
- anchors are inline elements; must be placed in a block element such as p or h1

Block and inline elements



block elements contain an entire large region of content

- examples: paragraphs, lists, table cells
- the browser places a margin of whitespace between block elements for separation

inline elements affect a small amount of content

- examples: bold text, code fragments, images
- the browser allows many inline elements to appear on the same line
- must be nested inside a block element

Quotations <blockquote>

a lengthy quotation (block)

```
<p>As Lincoln said in his famous Gettysburg Address:</p>
<blockquote>
  <p>Fourscore and seven years ago, our fathers brought forth
  on this continent a new nation, conceived in liberty, and
  dedicated to the proposition that all men are created equal.</p>
</blockquote>
```

HTML

As Lincoln said in his famous Gettysburg Address:

Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

output

Inline quotations <q>

a short quotation (inline)

<code><p>Quoth the Raven, <q>Nevermore.</q></p></code>	HTML
Quoth the Raven, “Nevermore.”	output

- Why not just write the following?

`<p>Quoth the Raven, "Nevermore."</p>`

Computer code `<code>`

a short section of computer code (usually shown in a fixed-width font)

```
<p>  
  The ul and ol  
  tags make lists.  
</p>
```

HTML

The `ul` and `ol` tags make lists.

output

Preformatted text <pre>

a large section of pre-formatted text (block)

<pre><pre> Bill Gates speaks You will be assimilated Microsoft fans delirious </pre></pre>	HTML
<pre> Bill Gates speaks You will be assimilated Microsoft fans delirious</pre>	output

- Displayed with exactly the whitespace / line breaks given in the text
- Shown in a fixed-width font by default

Line break: `
`

forces a line break in the middle of a block element (inline)

```
<p>
```

```
The woods are lovely, dark and deep, <br /> But I have promises to  
keep, <br /> And miles to go before I sleep, <br /> And miles to go  
before I sleep.
```

```
</p>
```

The woods are lovely, dark and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.

- **Warning:** Don't over-use br (guideline: ≥ 2 in a row is bad)

HTML Character Entities

a way of representing any Unicode character within a web page

character(s)	entity
< >	< >
é è ñ	é è ñ
™ ©	™ ©
π δ Δ	π δ Δ
И	И
" &	" &

- [Complete list of HTML entities](#)

Deletions and insertions: , <ins>

content that should be considered deleted or added to the document (inline)

```
<p>  
  <del>Final Exam</del> <ins>Midterm</ins> is on  
  <del>Aug 29</del> <ins>Apr 17</ins>.  
</p>
```

HTML

~~Final Exam~~ Midterm is on ~~Aug 29~~ Apr 17.

output

Abbreviations: <abbr>

an abbreviation, acronym, or slang term (inline)

```
<p>
```

```
Safe divers always remember to check their  
<abbr title="Self-Contained Underwater Breathing  
Apparatus">SCUBA</abbr> gear.
```

```
</p>
```

HTML

Safe divers always remember to check their SCUBA gear.

output

- The title will appear when the abbreviated word is hovered over
- In some browsers the abbreviated word will have a dashed underline

Phrase elements : ``, ``

em: emphasized text (usually rendered in italic)

strong: strongly emphasized text (usually rendered in bold)

```
<p>  
  HTML is <em>really</em>,  
  <strong>REALLY</strong> fun!  
</p>
```

HTML is *really*, **REALLY** fun!

- as usual, the tags must be properly nested for a valid page

Nesting tags

```
<p>  
    HTML is <em>really,  
    <strong>REALLY</em> lots of</strong> fun!  
</p>
```

- tags must be correctly nested
 - (a closing tag must match the most recently opened tag)
- the browser may render it correctly anyway, but it is invalid HTML
 - (how would we get the above effect in a valid way?)

Images: ``

inserts a graphical image into the page (inline)

```

```



- the src attribute specifies the image URL
- HTML5 also requires an alt attribute describing the image

Web Standards

It is important to write proper HTML code and follow proper syntax.

Why use valid HTML and web standards?

- more rigid and structured language
- more interoperable across different web browsers
- more likely that our pages will display correctly in the future
- can be interchanged with other XML data: [SVG](#) (graphics), [MathML](#), [MusicML](#), [etc.](#)

W3C HTML Validator

```
<p>  
  <a href="http://validator.w3.org/check/referer">  
      
  </a>  
</p>
```



- validator.w3.org
- checks your HTML code to make sure it follows the official HTML syntax
- more picky than the browser, which may render bad HTML correctly

The bad way to produce styles

<p>

```
<font face="Arial">Welcome to Greasy Joe's.</font>
You will <b>never</b>, <i>ever</i>, <u>EVER</u> beat
<font size="+4" color="red">OUR</font> prices!
```

</p>

Welcome to Greasy Joe's. You will **never**, *ever*, EVER beat **OUR** prices!

Cascading Style Sheets (CSS): `<link>`

```
<head>
  ...
  <link href="filename" type="text/css" rel="stylesheet" />
  ...
</head>
```

- CSS describes the appearance and layout of information on a web page (as opposed to HTML, which describes the content of the page)
- can be embedded in HTML or placed into separate .css file (preferred)

Basic CSS rule syntax

```
selector {  
    property: value;  
    property: value;  
    ...  
    property: value;  
}  
  
p {  
    font-family: sans-serif;  
    color: red;  
}
```

- a CSS file consists of one or more rules
- a rule's selector specifies HTML element(s) and applies style properties
- a selector of * selects all elements

CSS properties for colors

```
p {  
    color: red;  
    background-color: yellow;  
}
```

This paragraph uses the style above.

Property	Description
<u>color</u>	color of an element's text
<u>background-color</u>	color that will appear behind the element

Specifying colors

```
p { color: red; }  
h2 { color: rgb(128, 0, 196); }  
h4 { color: #FF8800; }
```

This paragraph uses the first style above.

This h2 uses the second style above.

This h4 uses the third style above.

- color names: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, (white), yellow
- RGB codes: red, green, and blue values from 0 (none) to 255 (full)
- hex codes: RGB values in base-16 from 00 (0, none) to FF (255, full)

CSS properties for fonts

property	description
<u>font-family</u>	which font will be used
<u>font-size</u>	how large the letters will be drawn
<u>font-style</u>	used to enable/disable italic style
<u>font-weight</u>	used to enable/disable bold style
<u>Complete list of font properties</u>	

font-size

```
p {  
    font-size: 14pt;  
}
```

This paragraph uses the style above.

- units: pixels (px) vs. point (pt) vs. m-size (em)
16px, 16pt, **1.16em**
- vague font sizes: xx-small, x-small, small, medium, large, x-large, **xx-large**,
smaller, **larger**
- percentage font sizes, e.g.: 90%, 120%

font-family

```
p {  
  font-family: Georgia;  
}  
h2 {  
  font-family: "Courier New";  
}
```

This paragraph uses the first style above.

This h2 uses the second style above.

- enclose multi-word font names in quotes

More about font-family

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

This paragraph uses the above style.

- can specify multiple fonts from highest to lowest priority
- generic font names:

serif, sans-serif, *cursive*, **FANTASY**, monospace

font-weight, font-style

```
p {  
  font-weight: bold;  
  font-style: italic;  
}
```

This paragraph uses the style above.

- either of the above can be set to normal to turn them off (e.g. headings)