

Hypertext Markup Language (HTML)

- Describes the *content* and structure of information on a web page
- Not the same as the presentation (appearance on screen)
- Surrounds text content with opening and closing tags
- Each tag's name is called an element
 - ▣ syntax: `<element> content </element>`
 - ▣ example: `<p>This is a paragraph</p>`

XHTML

3

- Uses a markup format called XML
- XML + HTML = XHTML
- Standardized in 2000
- A strict XHTML page uses some different syntax and tags than HTML

CSC443: Web Programming

Structure of XHTML page

4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    information about the page
  </head>

  <body>
    page contents
  </body>
</html>
```

HTML

- HTML is saved with extension .html
- Basic structure: tags that enclose content, i.e., elements
- **Header** describes the page
- **Body** contains the page's contents

CSC443: Web Programming

Page Title <title>

5

```
...
    <head>
        <title> HARRY POTTER AND THE DEATHLY HALLOWS
- PART 2 </title>
    </head>
...
```

HTML

- ❑ Placed within the head of the page
- ❑ Displayed in web browser's title mark and when bookmarking the page

CSC443: Web Programming

Paragraph <p>

6

```
...
    <body>
        <p> Harry Potter and the Deathly Hallows,
the last book in the series, begins directly after the
events of the sixth book.
Voldemort      has completed his ascension to power and
gains          control of the Ministry of Magic</p>
    </body>
...
```

HTML

Harry Potter and the Deathly Hallows, the last book in the series, begins directly after the events of the sixth book. Voldemort has completed his ascension to power and gains control of the Ministry of Magic

output

- ❑ Placed within the body of the page

CSC443: Web Programming

Headings `<h1>`, `<h2>`, ... `<h6>`

7

```
<h1> Harry Potter </h1>  
<h2> Books </h2>  
<h3> Harry Potter and the Philosopher's Stone </h3>
```

HTML

Harry Potter

Books

Harry Potter and the Philosopher's Stone

output

CSC443: Web Programming

Horizontal rule `<hr />`

8

```
<p> First paragraph </p>  
<hr />  
<p> Second Paragraph </p>
```

HTML

First Paragraph

Second Paragraph

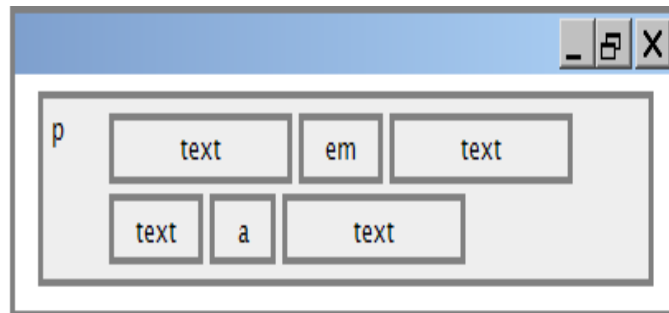
output

- ❑ Should be immediately closed with `/>`

CSC443: Web Programming

Block and Inline Statements

9

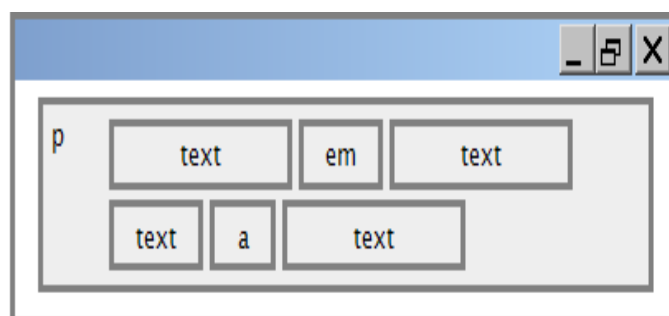


- Block elements contain an entire large region of content
 - ▣ examples: paragraphs, lists, table cells
 - ▣ the browser places a margin of whitespace between block elements for separation

CSC443: Web Programming

Block and Inline Statements (cont.)

10



- 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

CSC443: Web Programming

More HTML tags

11

- Some tags can contain additional information called attributes
 - ▣ syntax:
**<element attribute="value" attribute="value">
content </element>**
 - ▣ example: **Next page**

CSC443: Web Programming

More HTML tags

12

- Some tags don't contain content; can be opened and closed in one tag
 - ▣ syntax:
<element attribute="value" attribute="value" />
 - ▣ example: **<hr />**
 - ▣ example:

CSC443: Web Programming

Links <a>

13

```
<p>
Search
<a href="http://www.google.com/">Google</a>
now!
</p>
```

HTML

Search [Google](http://www.google.com/) now!

output

- The **href** attribute specifies the destination URL
- Links or *anchors* are inline elements, so they must be placed inside a block element such as a `p` or `h1`

CSC443: Web Programming

More about anchors

14

```
<p><a href="deathlyHallows-book.html">Harry Potter and the
Deathly Hallows Book</a></p>
```

```
<p><a href="http://en.wikipedia.org"
title="Search">Wikipedia</a></p>
```

HTML

[Harry Potter and the Deathly Hallows](#)

[Wikipedia](#)

output

- Types of URLs that can appear in anchors:
 - ▣ Absolute: to another web site
 - ▣ Relative: to another page on this web site

CSC443: Web Programming

Nesting tags

15

Bad

```
<p>  
<a href="deathlyHallows-book.html"> Harry Potter and the  
Deathly Hallows Book </p>  
<p>  
This text also links to Harry Potter Book</a>  
</p>
```

HTML

- ❑ 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 XHTML

CSC443: Web Programming

Images

16

```

```

HTML



- ❑ The src attribute specifies source of the image URL
- ❑ XHTML also requires an alt attribute describing the image

CSC443: Web Programming

More about images

17

```
<a href="http://harrypotter.net/">

</a>
```

HTML



- If placed inside an a anchor, the image will become a link
- The title attribute specifies an optional tooltip

Line Break

18

```
<p>One Ring to rule them all, One Ring to find them, <br
/> One Ring to bring them all and in the darkness bind
them.</p>
<p> In the Land of Mordor where the Shadows lie. </p>
```

HTML

One Ring to rule them all, One Ring to find them,
One Ring to bring them all and in the darkness bind them

In the Land of Mordor where the Shadows lie.

output

- br should be immediately closed with />
- br should not be used to separate paragraphs or used multiple times in a row to create spacing

Comments `<!-- ... -->`

19

```
<!-- My web page, by Bob Student  
CSE 380, Fall 2048 -->  
<p>CS courses are <!-- NOT --> a lot of fun!</p>
```

HTML

CS courses are a lot of fun!

output

- ❑ Comments are useful for disabling sections of a page
- ❑ Comments cannot be nested and cannot contain a --

CSC443: Web Programming

Phrase elements ``, ``

20

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

HTML

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

output

- ❑ **em**: emphasized text (usually in italic)
- ❑ **strong**: strongly emphasized text (usually in bold)
- ❑ The tags must be properly nested for a valid page

CSC443: Web Programming

Unordered list: ,

21

```
<ul>
<li>No shoes</li>
<li>No shirt</li>
<li>No problem!</li>
</ul>
```

HTML

- No shoes
- No shirt
- No problem!

output

- **ul** represents a bulleted list of items (block)
- **li** represents a single item within the list (block)

CSC443: Web Programming

More about unordered lists

22

```
<ul>
<li>Harry Potter characters:
<ul>
<li>Harry Potter</li>
<li>Hermione</li>
<li>Ron</li>
</ul>
</li>
<li>LOTR characters:
<ul>
<li>Frodo</li>
<li>Bilbo</li>
<li>Sam</li>
</ul>
</li>
</ul>
```

HTML

CSC443: Web Programming

More about unordered lists (cont.)

23

- Harry Potter characters:
 - Harry Potter
 - Hermione
 - Ron
- LOTR characters:
 - Frodo
 - Bilbo
 - Sam

output

CSC443: Web Programming

Ordered list

24

```
<p>Apple business model:</p>
<ol>
<li>Beat Microsoft</li>
<li>Beat Google</li>
<li>Conquer the world!</li>
</ol>
```

HTML

Apple business model:
1. Beat Microsoft
2. Beat Google
3. Conquer the world

output

- **ol** represents a numbered list of items
- we can make lists with letters or Roman numerals using CSS (later)

CSC443: Web Programming

Common error: Not closing a list

25

```
<ul>
<li>No shoes</li>
<li>No shirt</li>
<li>No problem!</li>
<p>Paragraph after list...</p>
```

HTML

- No shoes
- No shirt
- No problem!

Paragraph after list...

output

- ❑ If you leave a list open, subsequent contents will be indented

CSC443: Web Programming

Common Error: Improper nested list placement

26

```
<ul>
<li>Harry Potter characters:</li>
<ul>
<li>Harry Potter</li>
<li>Hermione</li>
<li>Ron</li>
</ul>
</li>
<li>LOTR characters:
<ul>
<li>Frodo</li>
<li>Bilbo</li>
<li>Sam</li>
</ul>
</li>
```

HTML

- ❑ closing the outer li too early (or not at all) will render correctly in most browsers, but it is incorrect XHTML

CSC443: Web Programming

Definition list <dl>, <dt>, <dd>

27

```
<dl>
<dt>newbie</dt> <dd>one who does not have mad skills</dd>
<dt>jaded</dt> <dd>tired, bored, or lacking enthusiasm
</dd>
<dt>frag</dt> <dd>a kill in a shooting game</dd>
</dl>
```

HTML

newbie	one who does not have mad skills
jaded	Tired, bored, or lacking enthusiasm
frag	a kill in a shooting game

output

- ❑ **dl** represents a list of definitions of terms
- ❑ **dt** represents each term, and **dd** its definition

CSC443: Web Programming

Tables <table>, <tr>, <td>

28

```
<table>
  <tr><td>1,1</td><td>1,2 okay</td></tr>
  <tr><td>2,1 real wide</td><td>2,2</td></tr>
</table>
```

HTML

1,1	1,2 okay
2,1 real wide	2,2

output

- ❑ **table** defines the overall table, **tr** each row, and **td** each cell's data
- ❑ Useful for displaying large row/column data sets
- ❑ NOTE: tables are sometimes used by novices for web page layout, but this is not proper semantic HTML and should be avoided

CSC443: Web Programming

Table headers, captions: <th>, <caption>

29

```
<table>
  <caption>My important data</caption>
  <tr><th>Column 1</th><th>Column 2</th></tr>
  <tr><td>1,1</td><td>1,2 okay</td></tr>
  <tr><td>2,1 real wide</td><td>2,2</td></tr>
</table>
```

HTML

My important data

Column 1

Column 2

1,1

1,2 okay

2,1 real wide

2,2

output

- `th` cells in a row are considered headers
- a caption at the start of the table labels its meaning

CSC443: Web Programming

Quotations <blockquote>

30

```
<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

- a lengthy quotation

CSC443: Web Programming

Inline quotations <q>

31

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

HTML

Quoth the Raven, “Nevermore.”

output

- a short quotation
- Why not just write the following?
- `<p>Quoth the Raven, "Nevermore."</p>`
- We don't use " marks for two reasons:
 - XHTML shouldn't contain literal quotation mark characters; they should be written as `"`;
 - using `<q>` allows us to apply CSS styles to quotations

HTML Character Entities

32

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

Inline quotations <q>

33

```
<lt;p>>
<lt;a
href=&quot;http://google.com/search?q=xenia&amp;ie=utf-
8&amp;aq=t&quot;&gt;
Search Google for Xenia
<lt;/a>>
<lt;/p>>
```

HTML

```
<p> <a href="http://google.com/search?q=xenia&ie=utf-8&aq=t"> Search
Google for Xenia </a> </p>
```

output

CSC443: Web Programming

Computer code <code>

34

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

HTML

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

output

□ code: a short section of computer code

CSC443: Web Programming

Preformatted text <pre>

35

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

HTML

```
Bill Gates speaks
    You will be assimilated
    Microsoft fans delirious
```

output

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

CSC443: Web Programming

Preformatted text <pre>

36

```
<pre><code>
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
</code></pre>
```

HTML

```
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

output

- When showing a large section of computer code, enclose it in a `pre` to preserve whitespace and a `code` to describe the semantics of the content

CSC443: Web Programming

Web Standards

37

- Why use XHTML 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.

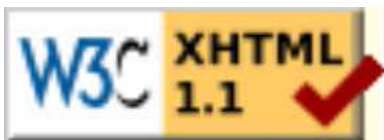
CSC443: Web Programming

W3C XHTML Validator

38

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

HTML



- checks your HTML code to make sure it meets the official strict XHTML specifications

CSC443: Web Programming

Web page metadata <meta>

39

```
<meta name="description"
content="Harry Potter Official Website." />
<meta name="keywords" content="harry potter, harry potter
and the deathly hallows, deathly hallows, ministry of
magic, resurrection stone, clock of invisibility" />
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1" />
```

HTML

- information about your page (for a browser, search engine, etc.)
- placed in the head of your XHTML page
- meta tags often have both the name and content attributes

- some meta tags use the http-equiv attribute instead of name

CSC443: Web Programming

meta element to aid browser/web server

40

```
<meta http-equiv="Content-Type"
content="type of document (character encoding)" />
<meta http-equiv="refresh"
content="how often to refresh the page (seconds)" />
</head>
```

HTML

- using the Content-Type gets rid of the W3C "tentatively valid" warning

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

- the meta refresh tag can also redirect from one page to another:

```
<meta http-equiv="refresh" content="5;url=http://www.bjp.com" />
```

- why would we want to do this? (example)

meta element to describe the page

41

```
<head>
<meta name="author"
content="web page's author" />
<meta name="revised"
content="web page version and/or last modification date"
/>
<meta name="generator"
content="the software used to create the page" />
</head>
```

HTML

- many WYSIWYG HTML editors place their names in the meta generator tag (why?)

CSC443: Web Programming

meta element to aid search engines

42

```
<head>
<meta name="description"
content="how you want search engines to display your page"
/>
<meta name="keywords"
content="words to associate with your page (comma
separated)" />
</head>
```

HTML

- these are suggestions to search engines about how to index your page
- the search engine may choose to ignore them (why?)

CSC443: Web Programming