HTML

CS 4640 Programming Languages for Web Applications

[Robert W. Sebesta, “Programming the World Wide Web”]
[W3 Web Schools]

Anatomy of (Basic) Website

Your content + $\underbrace{\text{HTML}}_{\text{structure}}$ + $\underbrace{\text{CSS}}_{\text{presentation}}$ = Your website

A website is a way to present your content to the world, using HTML and CSS to present that content and make them look good

HTML: HyperText Markup Language

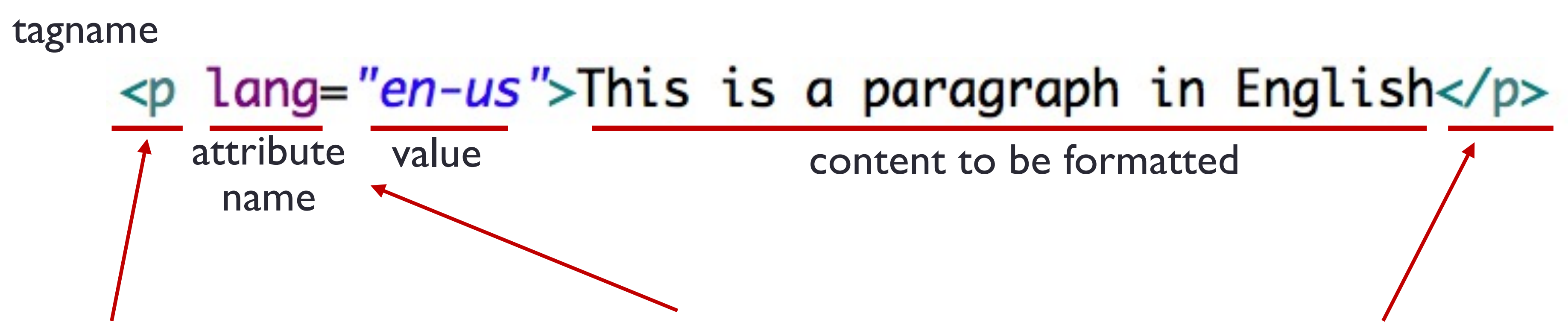
- Language for describing structure of a document
- An HTML file is a text file containing small markup tags (or elements)
- The markup tags tell the web browser how to display the page
- An HTML file denotes hierarchy of elements
- An HTML file can be created using a simple text editor, HTML editor, or IDE

HTML History

- Late 1980s: Tim Berners-Lee created first HTML version
- 1995: HTML 2.0 Published as standard with RFC 1866
- 1997: HTML 4.0 Standardized most modern HTML element with W3C recommendation
  - Encouraged use of CSS for styling elements over HTML attributes
- 2000: XHTML 1.0
  - Imposed stricter rules on HTML format
    - E.g., elements needed closing tag, attribute names in lowercase
- 2014: HTML5 published as W3C recommendation
  - New features for capturing more semantic information and declarative description of behavior
    - E.g., input constraints, new tags that explain purpose of content
  - Important changes to DOM

[https://en.wikipedia.org/wiki/HTML]

HTML Elements



```
<p lang="en-us">This is a paragraph in English</p>
```

Start a paragraph element

Opening tag begins an HTML element.

Opening tags must have a corresponding closing tag.

Set the language to English

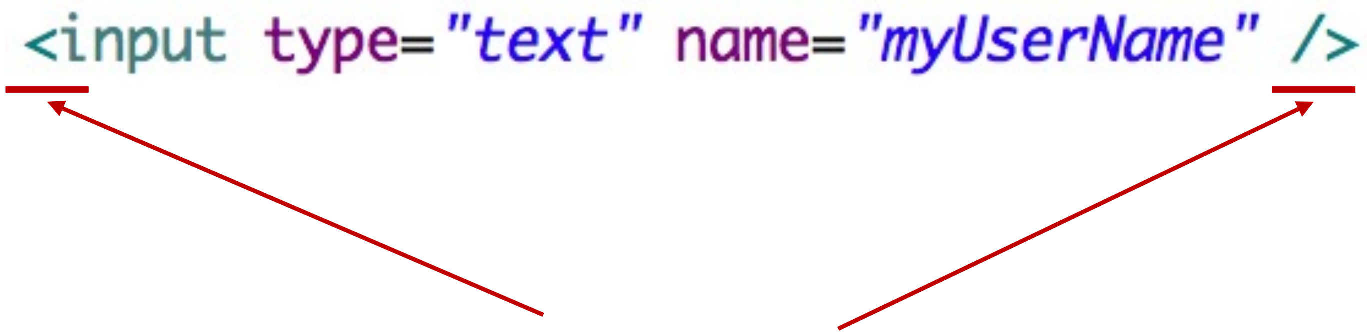
HTML attributes are name/value pairs that provide additional information about the contents of an element.

End a paragraph element

Closing tag ends and HTML element. All content between the tags and the tags themselves compromise an HTML element.

Each tag has a "start tag," "end tag," and some content in between, and optional attributes

HTML Elements



```
<input type="text" name="myUserName" />
```

Begin and end input element

Some HTML tags can be self closing, including a built-in closing tag

Content Types

Type	Description	Example
Metadata	Content hosted in the head of an HTML document. Doesn't appear in the web page but is used to describe a webpage and its relationships to other external resources	`<meta name="viewport" content="width=device-width, initial-scale=1">`
Flow	Text and all elements that can appear as content in the body of an HTML document	`<body>` `<h1>Heading</h1>` `<p>Some content…</p>` `</body>`
Sectioning	Used to structure the content of a web page and to help with layout	`<section class="highlight col">` `Some content …` `</section>`

Content Types (2)

Type	Description	Example

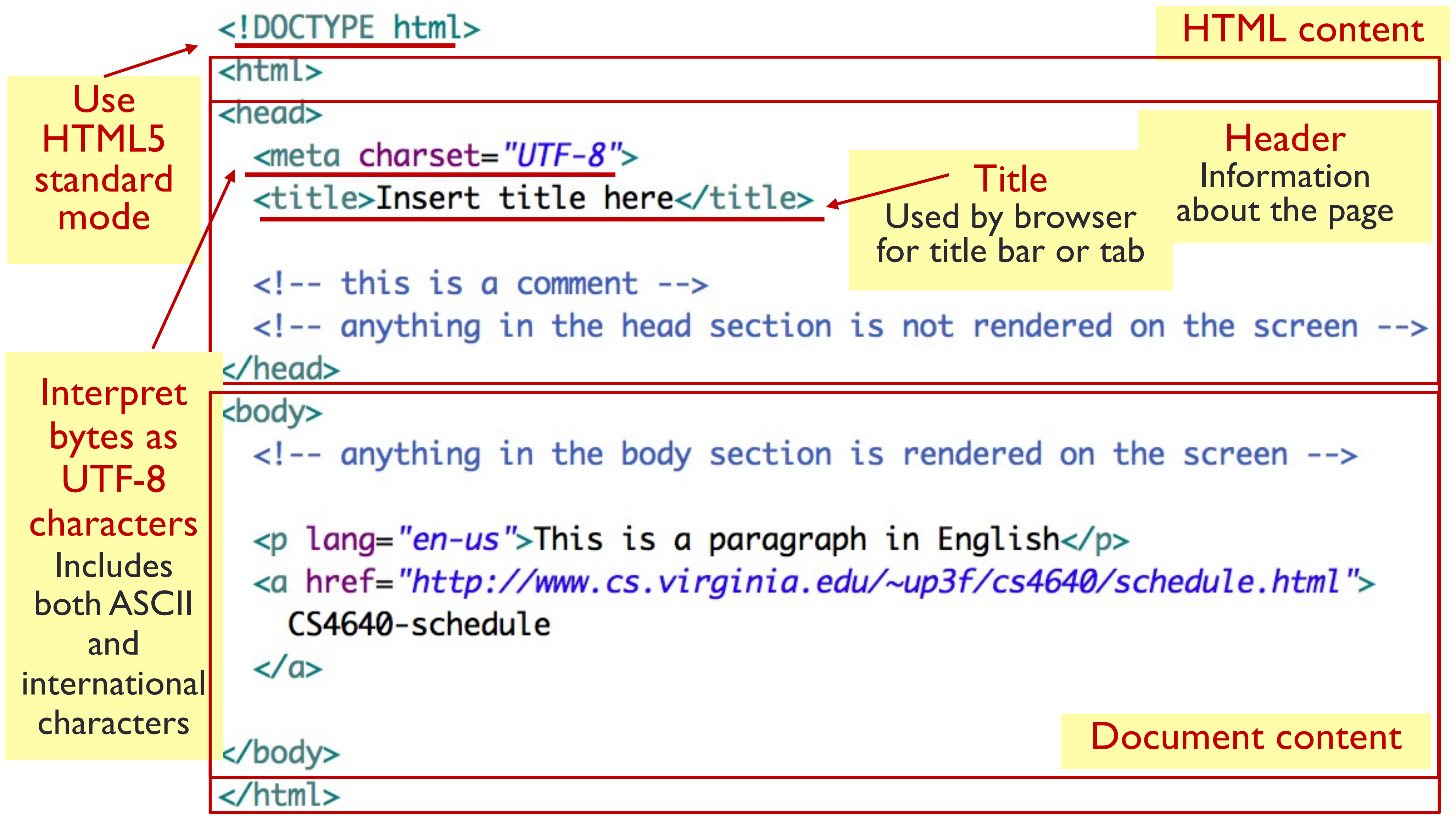
Phrasing	Elements for marking up content within a paragraph element such as text and typography	`<p>`
		   `<b>Emphasized text</b>`
		   `and some normal text`
		`</p>`

Heading	Elements used to define the headings of a section of an HTML document. The elements `h1-6` represent headings with `h1` having the highest ranking	`<h1>Main heading</h1>`
		`<h2>Sub-heading</h2>`

Embedded	Embedded content includes media, such as video, audio, and images	`<img src="media/monster.png” alt=“A cute monster image” width=“80%” />`

Interactive	Elements that a user can interact with such as media elements with controls, form inputs, buttons, and links	`<input type=“text” name=“username” required />`
		`<input type=“password” name=“pwd” required />`

A Starter HTML Document

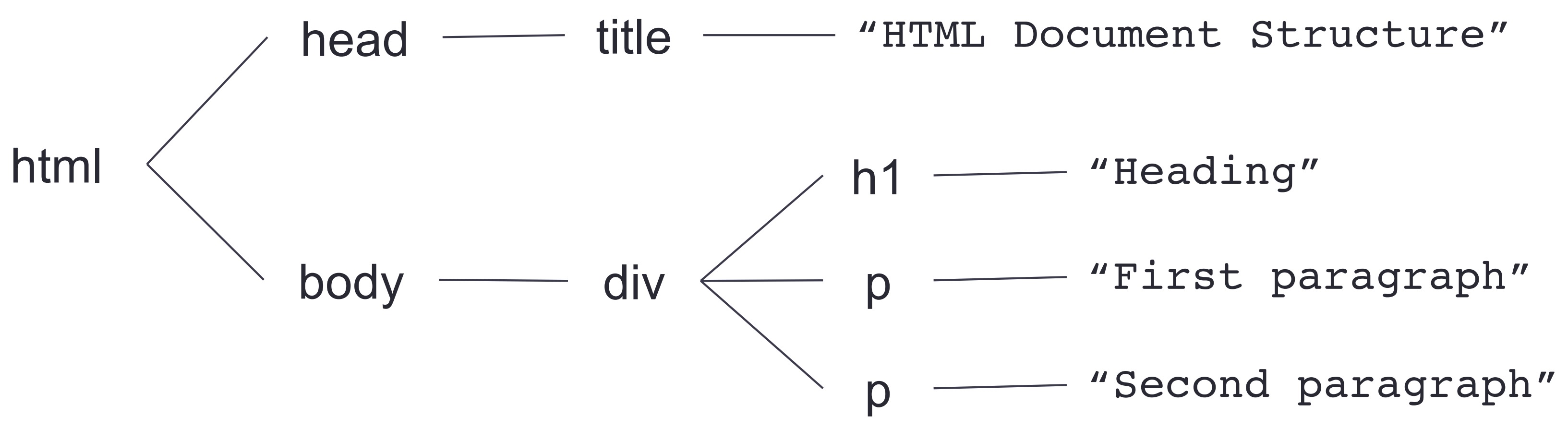


```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>

  <!-- this is a comment -->
  <!-- anything in the head section is not rendered on the screen -->
</head>
<body>
  <!-- anything in the body section is rendered on the screen -->

  <p lang="en-us">This is a paragraph in English</p>
  <a href="http://www.cs.virginia.edu/~up3f/cs4640/schedule.html">
    CS4640-schedule
  </a>

</body>
</html>
```

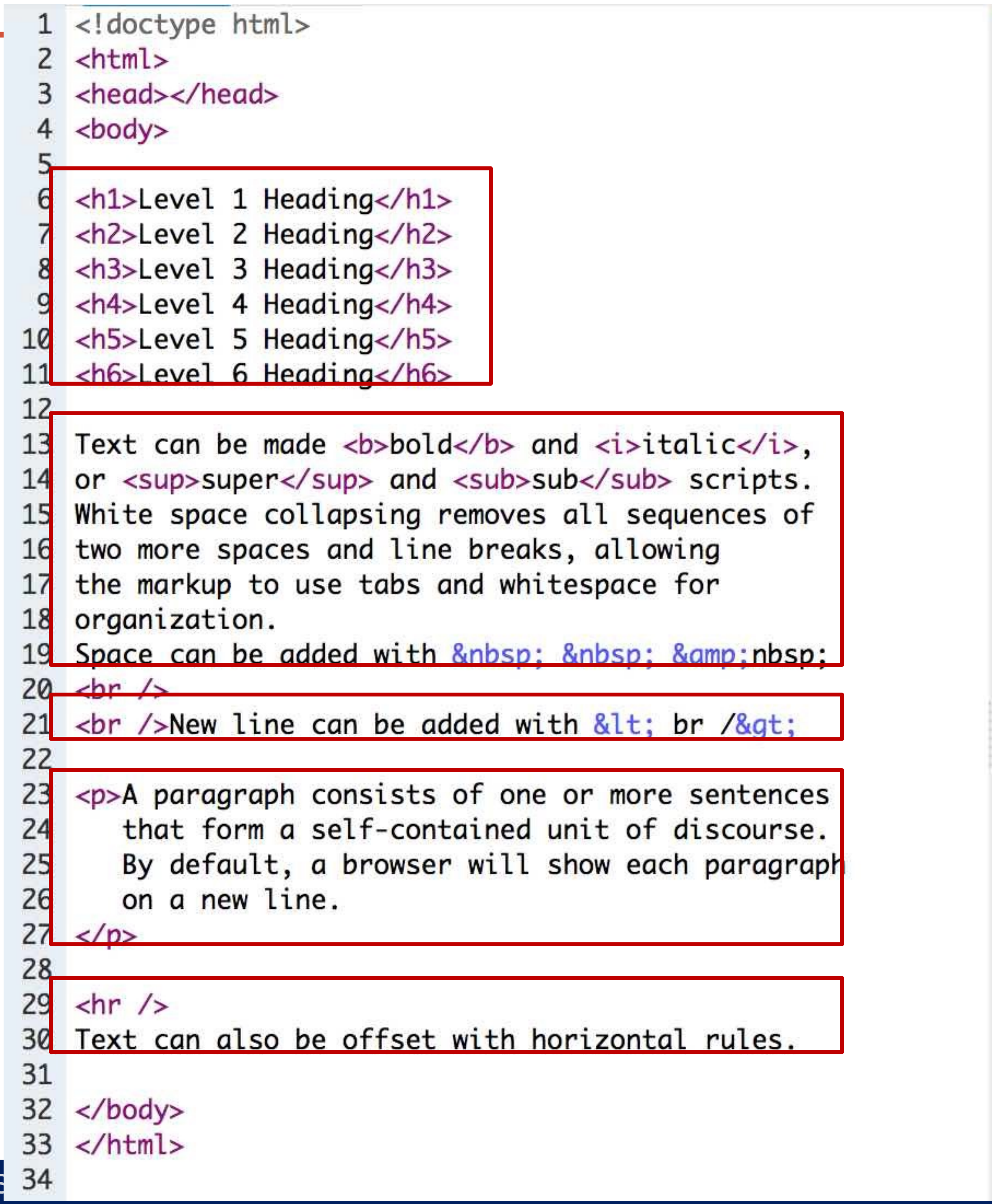
HTML content
Use HTML5 standard mode
Header Information about the page
Title Used by browser for title bar or tab
Interpret bytes as UTF-8 characters Includes both ASCII and international characters
Document content

HTML Document Structure

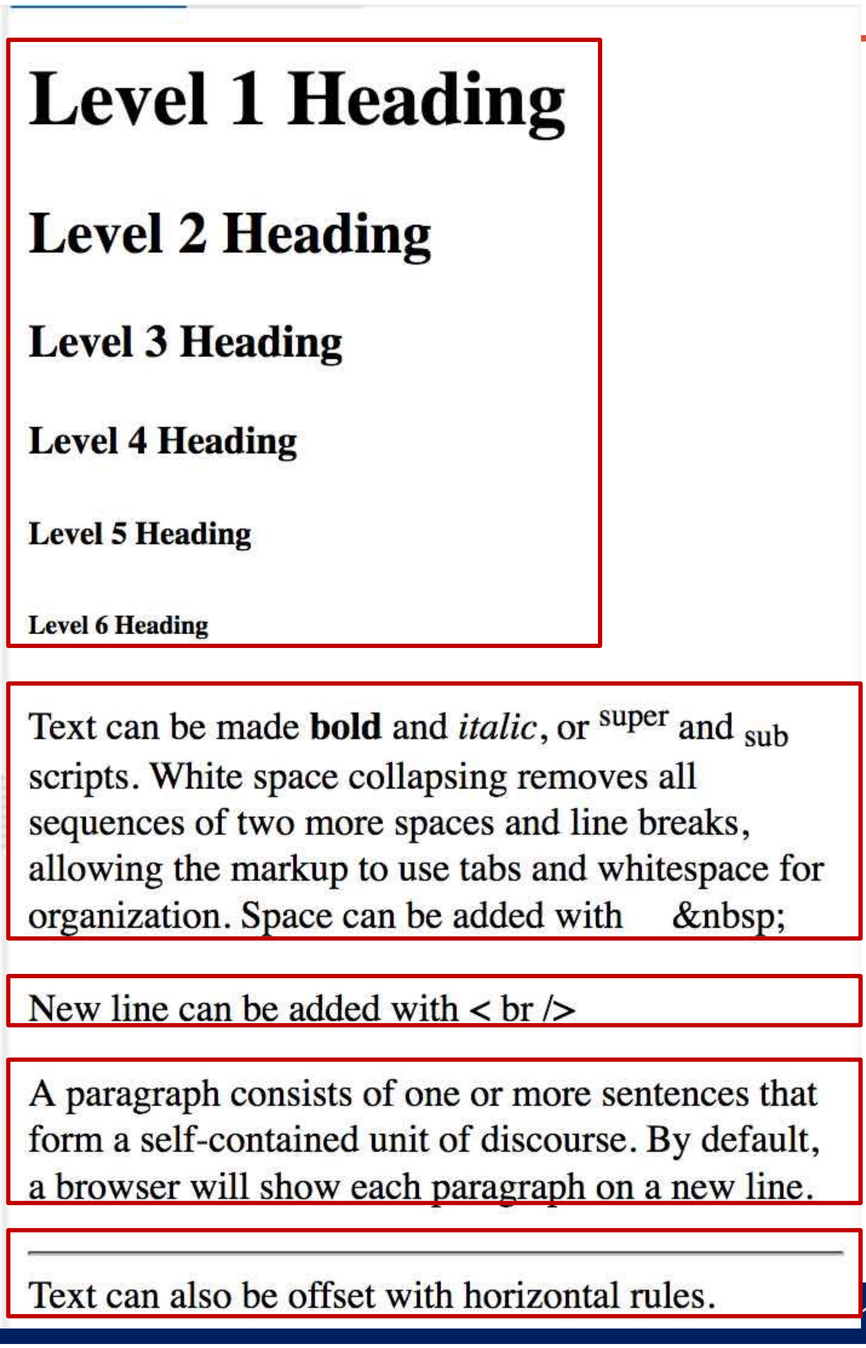
```
<html>
  <head>
    <title>HTML Document Structure</title>
  </head>
  <body>
    <div>
      <h1>Heading</h1>
      <p>First paragraph</p>
      <p>Second paragraph</p>
    </div>
  </body>
</html>
```



Text



```html
<!doctype html>
<html>
<head></head>
<body>

<h1>Level 1 Heading</h1>
<h2>Level 2 Heading</h2>
<h3>Level 3 Heading</h3>
<h4>Level 4 Heading</h4>
<h5>Level 5 Heading</h5>
<h6>Level 6 Heading</h6>

Text can be made <b>bold</b> and <i>italic</i>,
or <sup>super</sup> and <sub>sub</sub> scripts.
White space collapsing removes all sequences of
two more spaces and line breaks, allowing
the markup to use tabs and whitespace for
organization.
Space can be added with &nbsp; &nbsp; &amp;nbsp;
<br />
<br />New line can be added with &lt; br /&gt;

<p>A paragraph consists of one or more sentences
   that form a self-contained unit of discourse.
   By default, a browser will show each paragraph
   on a new line.
</p>

<hr />
Text can also be offset with horizontal rules.

</body>
</html>

```



Level 1 Heading

Level 2 Heading

Level 3 Heading

Level 4 Heading

Level 5 Heading

Level 6 Heading

Text can be made **bold** and *italic*, or $^{\text{super}}$ and $_{\text{sub}}$ scripts. White space collapsing removes all sequences of two more spaces and line breaks, allowing the markup to use tabs and whitespace for organization. Space can be added with

New line can be added with < br />

A paragraph consists of one or more sentences that form a self-contained unit of discourse. By default, a browser will show each paragraph on a new line.

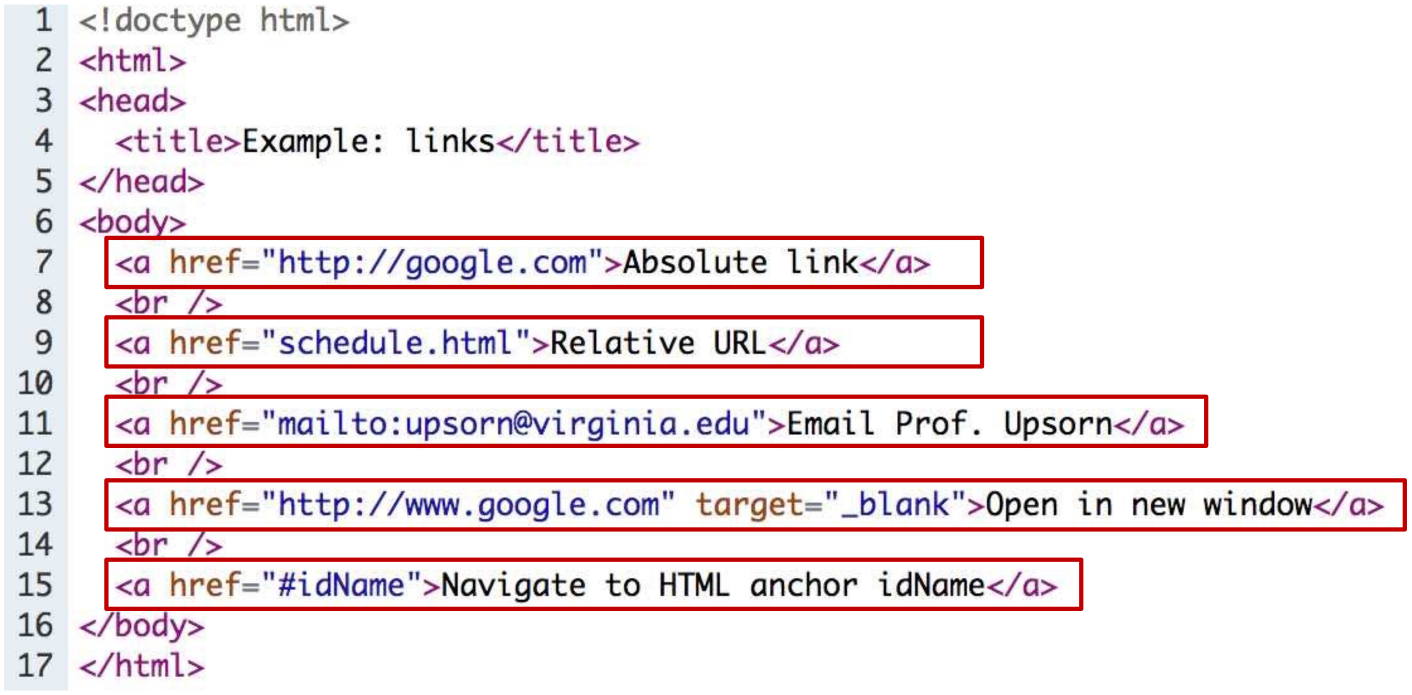
Text can also be offset with horizontal rules.

Semantic Markup

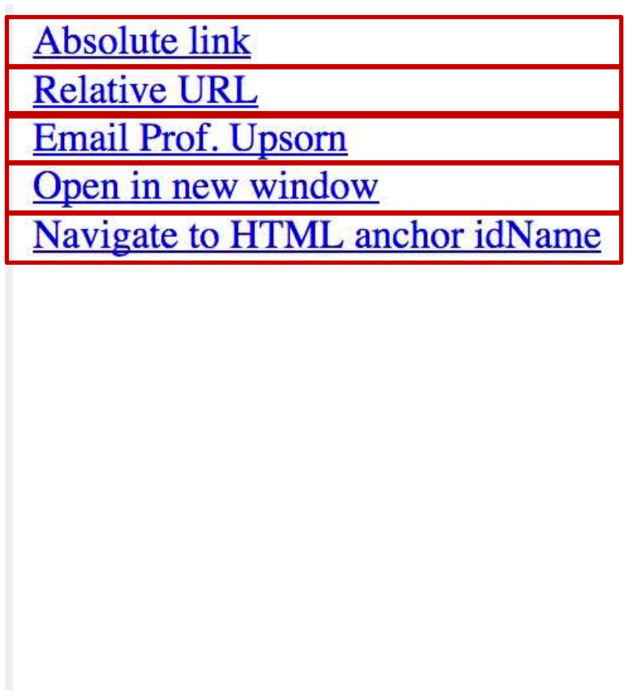
- Tags that can be used to denote the meaning of specific content
- Examples:

<strong>	An element that has importance
<blockquote>	An element that is a long quote
<q>	A short quote inline in paragraph
<abbr>	Abbreviation
<cite>	Reference to a work
<dfn>	The definition of a term
<address>	Contact information
<ins>	Content that is inserted
<del>	Content that is deleted

Links



```html
<!doctype html>
<html>
<head>
  <title>Example: links</title>
</head>
<body>
  <a href="http://google.com">Absolute link</a>
  <br />
  <a href="schedule.html">Relative URL</a>
  <br />
  <a href="mailto:upsorn@virginia.edu">Email Prof. Upsorn</a>
  <br />
  <a href="http://www.google.com" target="_blank">Open in new window</a>
  <br />
  <a href="#idName">Navigate to HTML anchor idName</a>
</body>
</html>
```



Absolute link
Relative URL
Email Prof. Upsorn
Open in new window
Navigate to HTML anchor idName

Images, Audio, and Video

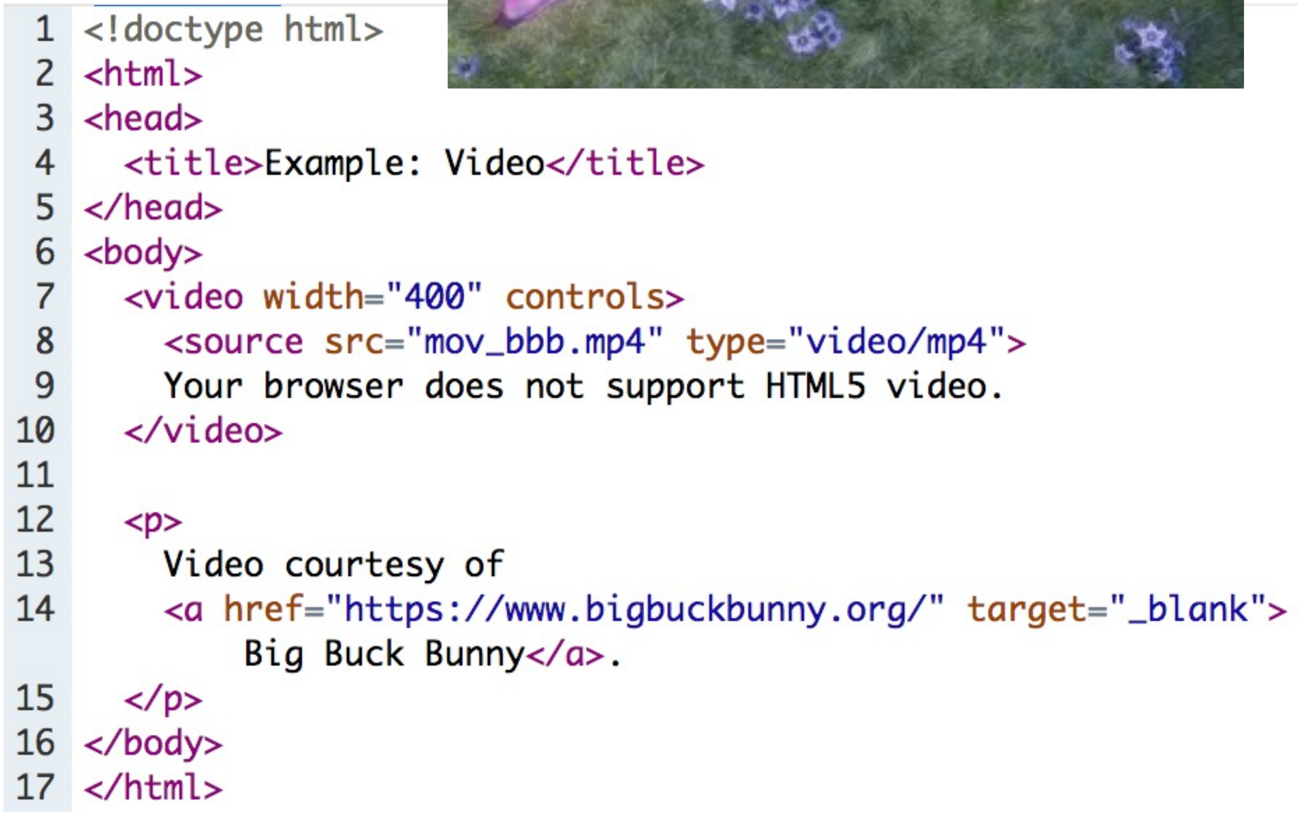
- HTML include standard support for <img>, <audio>, and <video>
- Use an alt attribute to make images accessible

```
<img src="http://www.cs.virginia.edu/~up3f/cs4640/images/thumb-up.jpg"
     alt="thumb-up"
     width="30"/>
```

- Common file formats
  - Images: .png, .gif, .jpg
  - Audio: .mp3
  - Video: .mp4

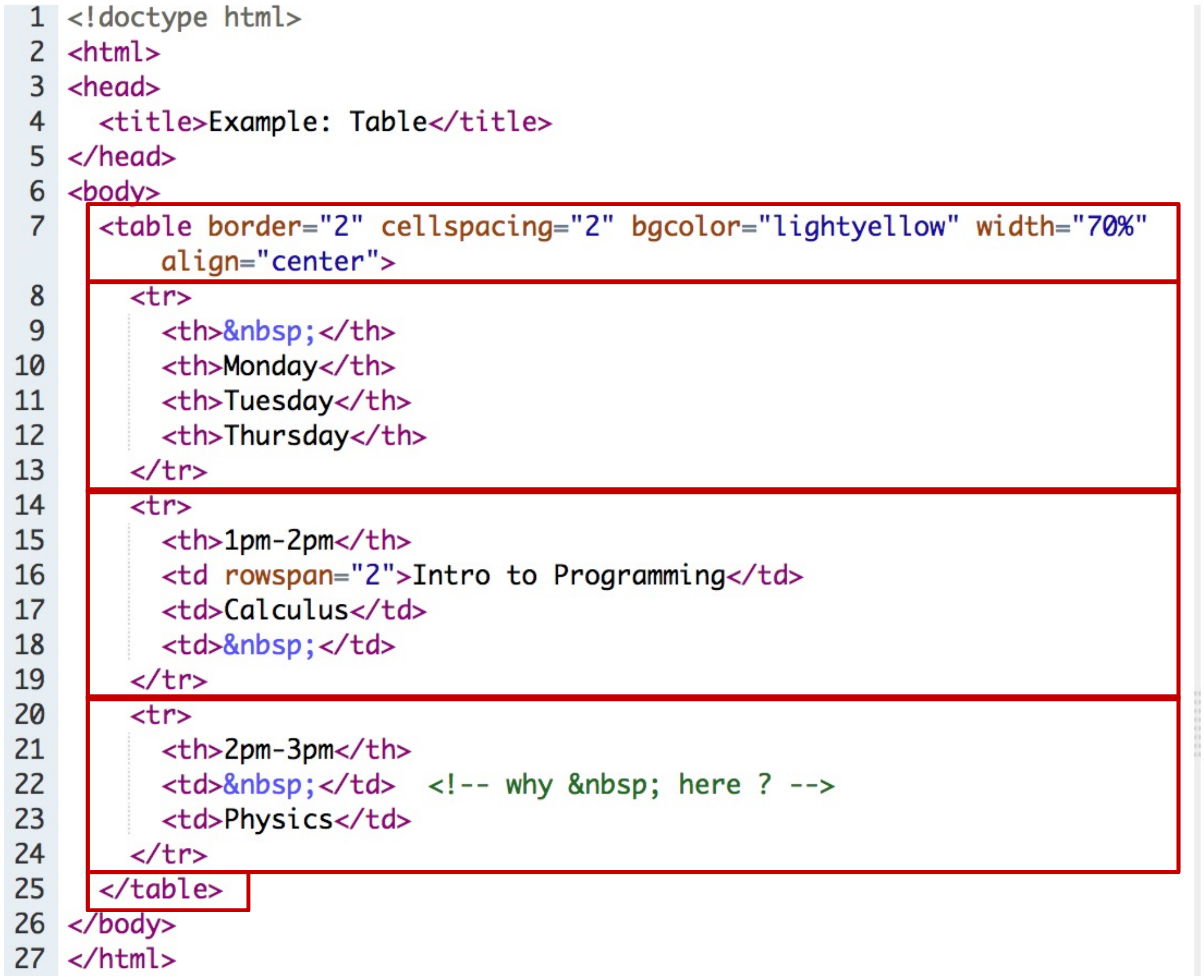
Video

- Important attributes for <video>
  - src – location of video
  - autoplay – tells browser to start play
  - controls – show the default controls
  - loop – loop the video
  - muted – mutes the audio from the video

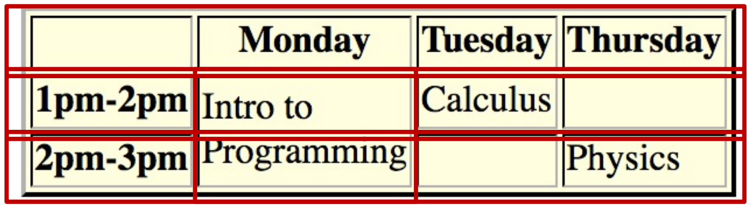


```
<!doctype html>
<html>
<head>
  <title>Example: Video</title>
</head>
<body>
  <video width="400" controls>
    <source src="mov_bbb.mp4" type="video/mp4">
    Your browser does not support HTML5 video.
  </video>

  <p>
    Video courtesy of
    <a href="https://www.bigbuckbunny.org/" target="_blank">
        Big Buck Bunny</a>.
  </p>
</body>
</html>
```

Tables



```
<!doctype html>
<html>
<head>
  <title>Example: Table</title>
</head>
<body>
  <table border="2" cellspacing="2" bgcolor="lightyellow" width="70%"
      align="center">
    <tr>
      <th>&nbsp;</th>
      <th>Monday</th>
      <th>Tuesday</th>
      <th>Thursday</th>
    </tr>
    <tr>
      <th>1pm-2pm</th>
      <td rowspan="2">Intro to Programming</td>
      <td>Calculus</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <th>2pm-3pm</th>
      <td>&nbsp;</td>  <!-- why &nbsp; here ? -->
      <td>Physics</td>
    </tr>
  </table>
</body>
</html>
```



	Monday	Tuesday	Thursday
1pm-2pm	Intro to Programming	Calculus	
2pm-3pm			Physics

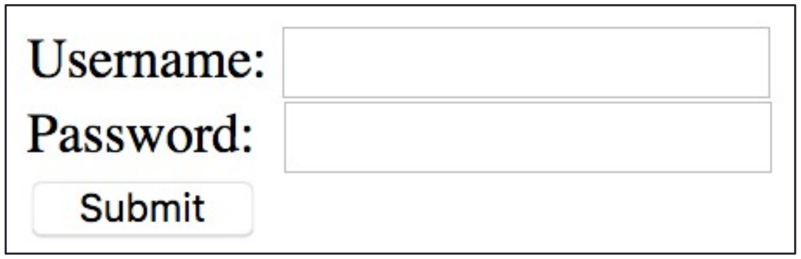
rowspan

Forms

Send form data to
plato.cs.virginia.edu/~up3f/formHandler.php

Action attribute should be omitted if not using form to submit data

```
<form action="http://plato.cs.virginia.edu/~up3f/formHandler.php" method="post">
  Username: <input type="text" name="username" value="" /> <br />
  Password: <input type="password" name="pwd" /> <br />
  <input type="submit" value="Submit" />
</form>
```

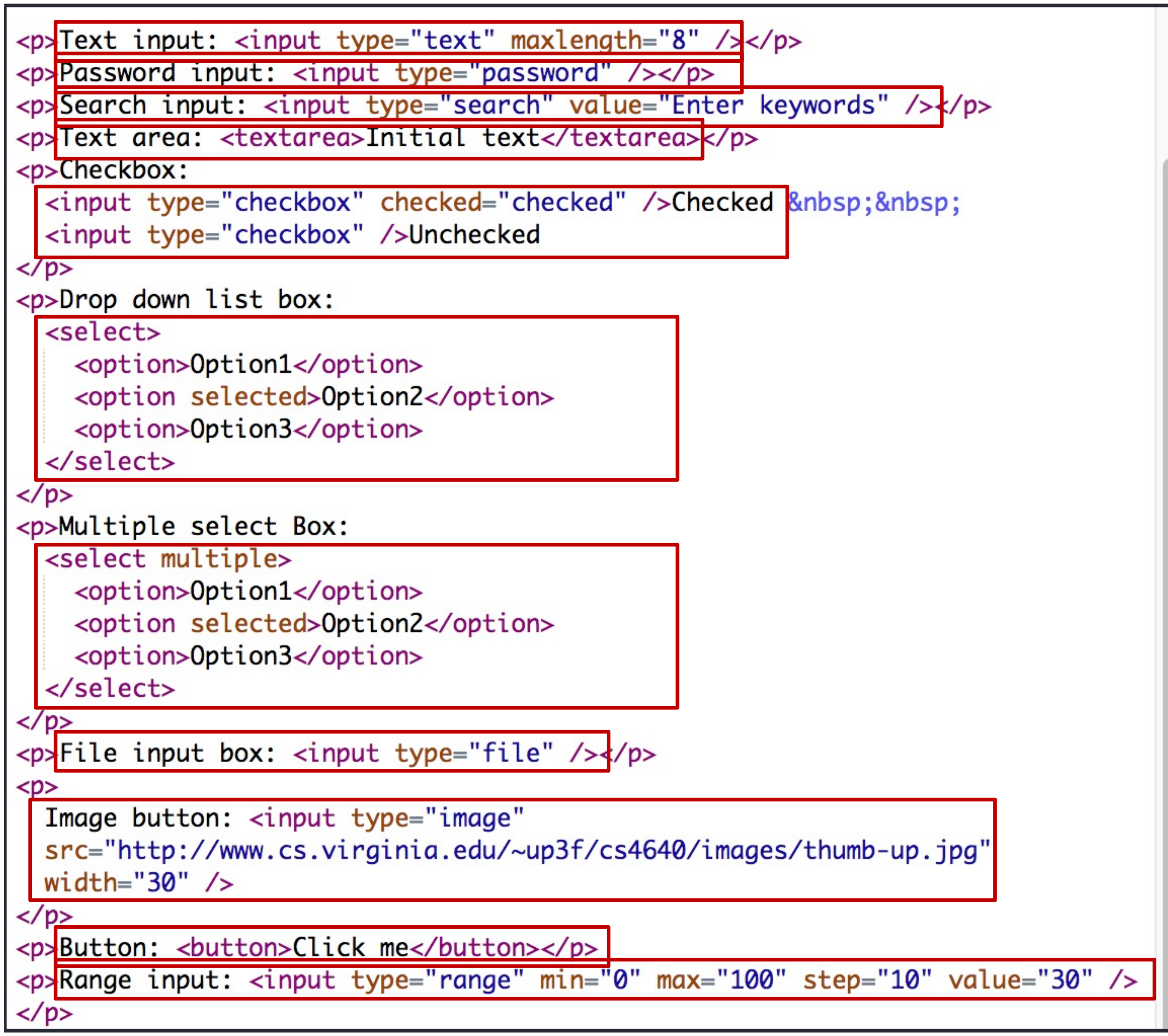


Transfer method

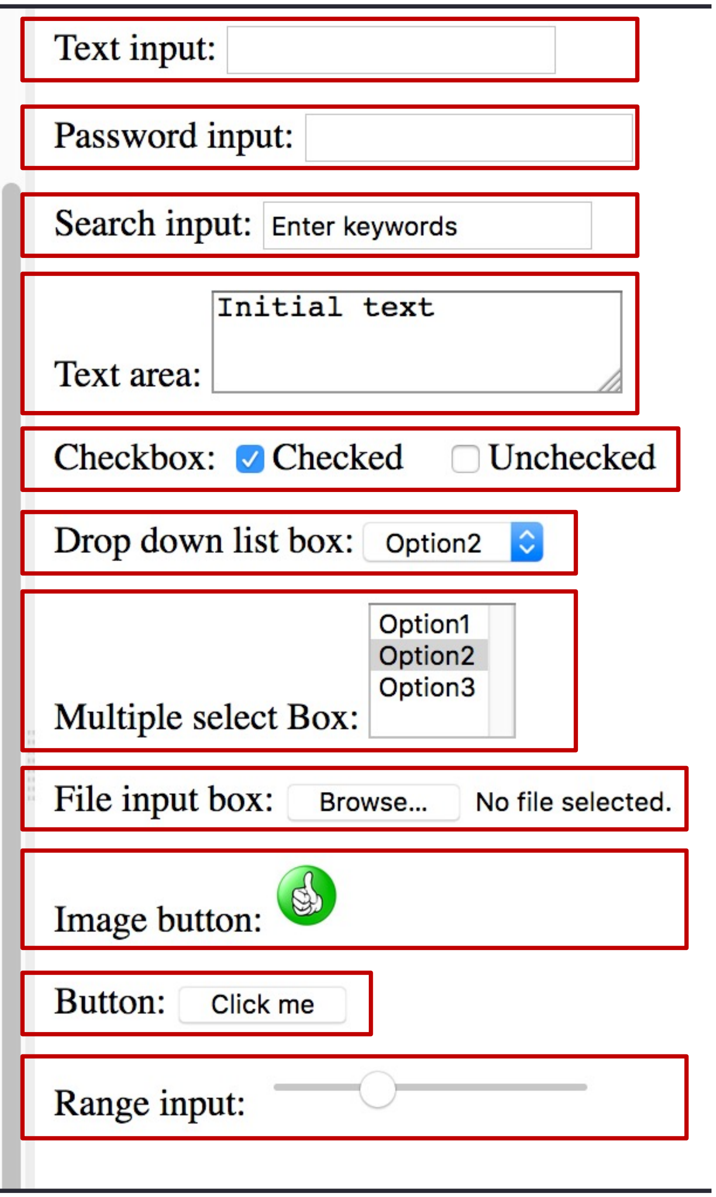
Method attribute specifies how data is transmitted to server.
Method="get" sends data appended to URL.
Method="post" sends data as an HTML document

- Elements in a form are submitted to the server.
- A form may (or may not) have controls.

Controls



```
<p>Text input: <input type="text" maxlength="8" /></p>
<p>Password input: <input type="password" /></p>
<p>Search input: <input type="search" value="Enter keywords" /></p>
<p>Text area: <textarea>Initial text</textarea></p>
<p>Checkbox:
  <input type="checkbox" checked="checked" />Checked &nbsp;&nbsp;
  <input type="checkbox" />Unchecked
</p>
<p>Drop down list box:
  <select>
    <option>Option1</option>
    <option selected>Option2</option>
    <option>Option3</option>
  </select>
</p>
<p>Multiple select Box:
  <select multiple>
    <option>Option1</option>
    <option selected>Option2</option>
    <option>Option3</option>
  </select>
</p>
<p>File input box: <input type="file" /></p>
<p>
  Image button: <input type="image"
  src="http://www.cs.virginia.edu/~up3f/cs4640/images/thumb-up.jpg"
  width="30" />
</p>
<p>Button: <button>Click me</button></p>
<p>Range input: <input type="range" min="0" max="100" step="10" value="30" />
</p>
```



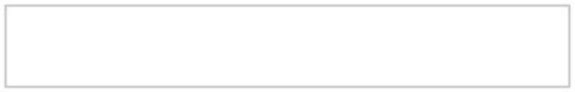
Text input:
Password input:
Search input: Enter keywords
Text area: Initial text
Checkbox: Checked Unchecked
Drop down list box: Option2
Multiple select Box: Option1 Option2 Option3
File input box: Browse... No file selected.
Image button:
Button: Click me
Range input:

Specialized Controls

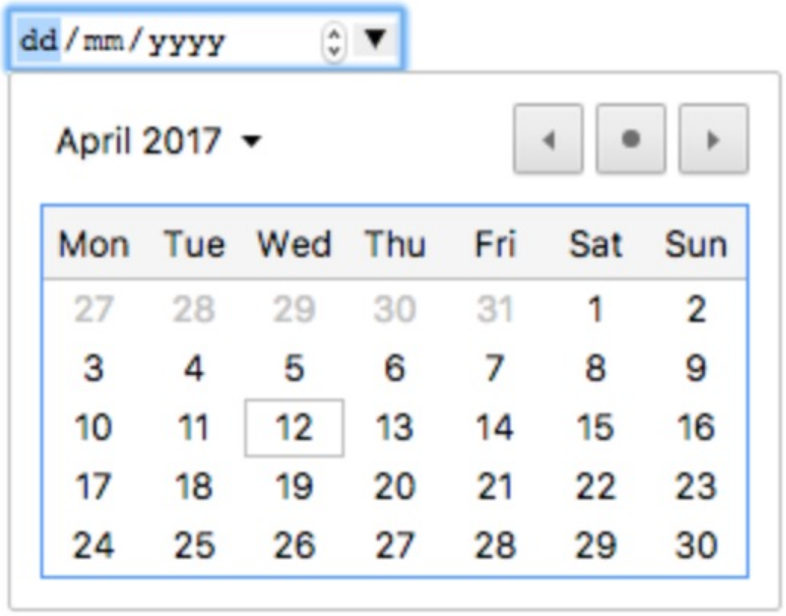
- A date input appears differently depending on browser support

```
<input type="date" />
```

Firefox



Chrome

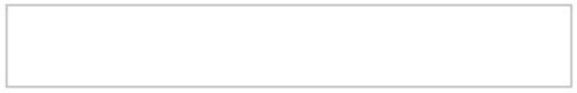


Specialized Controls

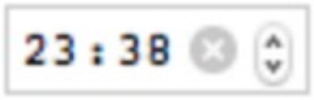
- A time input appears differently depending on browser support

```
<input type="time" />
```

Firefox



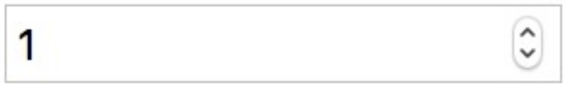
Chrome



23:38

- A number input can set restrictions on what numbers are accepted

```
<input type="number" min="1" max="12" />
```

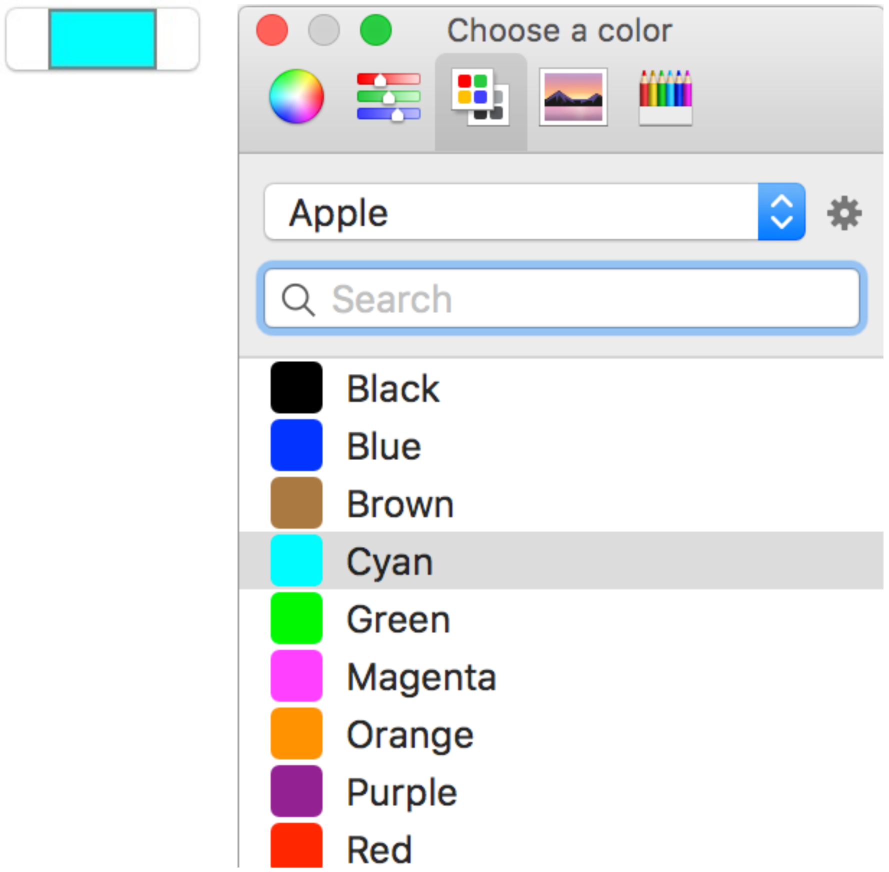


1

Specialized Controls

- A color input is used for input fields that should contain a color

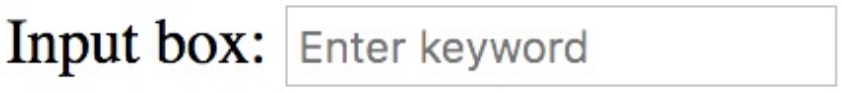
```
<input type="color" name="favcolor" value="#00ffff">
```



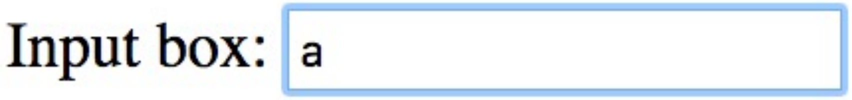
Labeling Inputs

- Suggestion can be placed inside input element

```
<p>Input box: <input type="text" placeholder="Enter keyword" /></p>
```

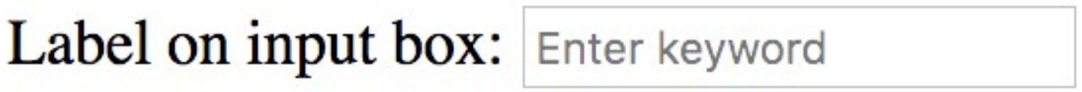

Input box: Enter keyword

- The suggestion disappears after user types


Input box: a

- Label can be attached to an input

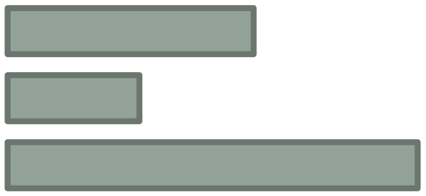
```
<p><label>
      Label on input box: <input type="text" placeholder="Enter keyword" />
   </label>
</p>
```


Label on input box: Enter keyword

Block vs. Inline Elements

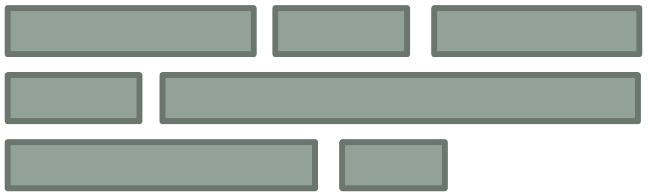
Block elements

- Appear on a new line
- Example:
  - <h1>
  - <p>
  - <li>
  - <table>
  - <form>
  - <div>



Inline elements

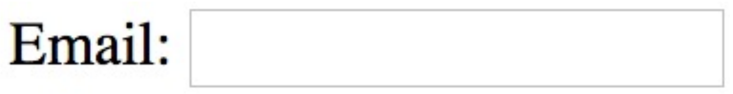
- Appear on the same line
- Example:
  - <a>
  - <b>
  - <em>
  - <input>
  - <img>
  - <span>



Validating Inputs

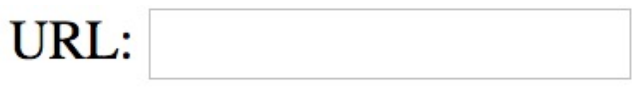
- Displays errors on invalid input immediately, making it easier to fix errors
- Check that input is a valid email

```
<label>Email: <input type="email" /></label>
```


Email:

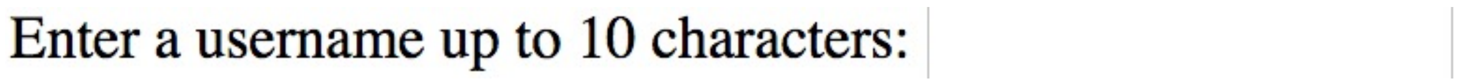
- Check that input is a valid URL

```
<label>URL: <input type="url" /></label>
```


URL:

- Constrain input to be at most max length

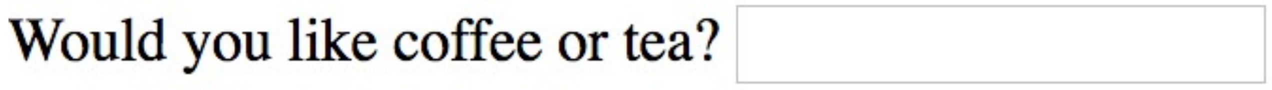
```
<label>Enter a username up to 10 characters:
  <input type="text" maxlength="10" />
</label>
```


Enter a username up to 10 characters:

Validating Inputs

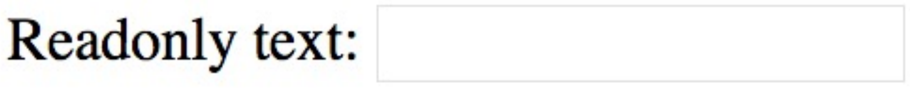
- Check that input matches regex pattern

```
<label>Would you like coffee or tea?
  <input type="text" pattern="coffee|tea" />
</label>
```


Would you like coffee or tea?

- Prevent all edits

```
<label>Readonly text:
  <input type="text" readonly />
</label>
```


Readonly text:

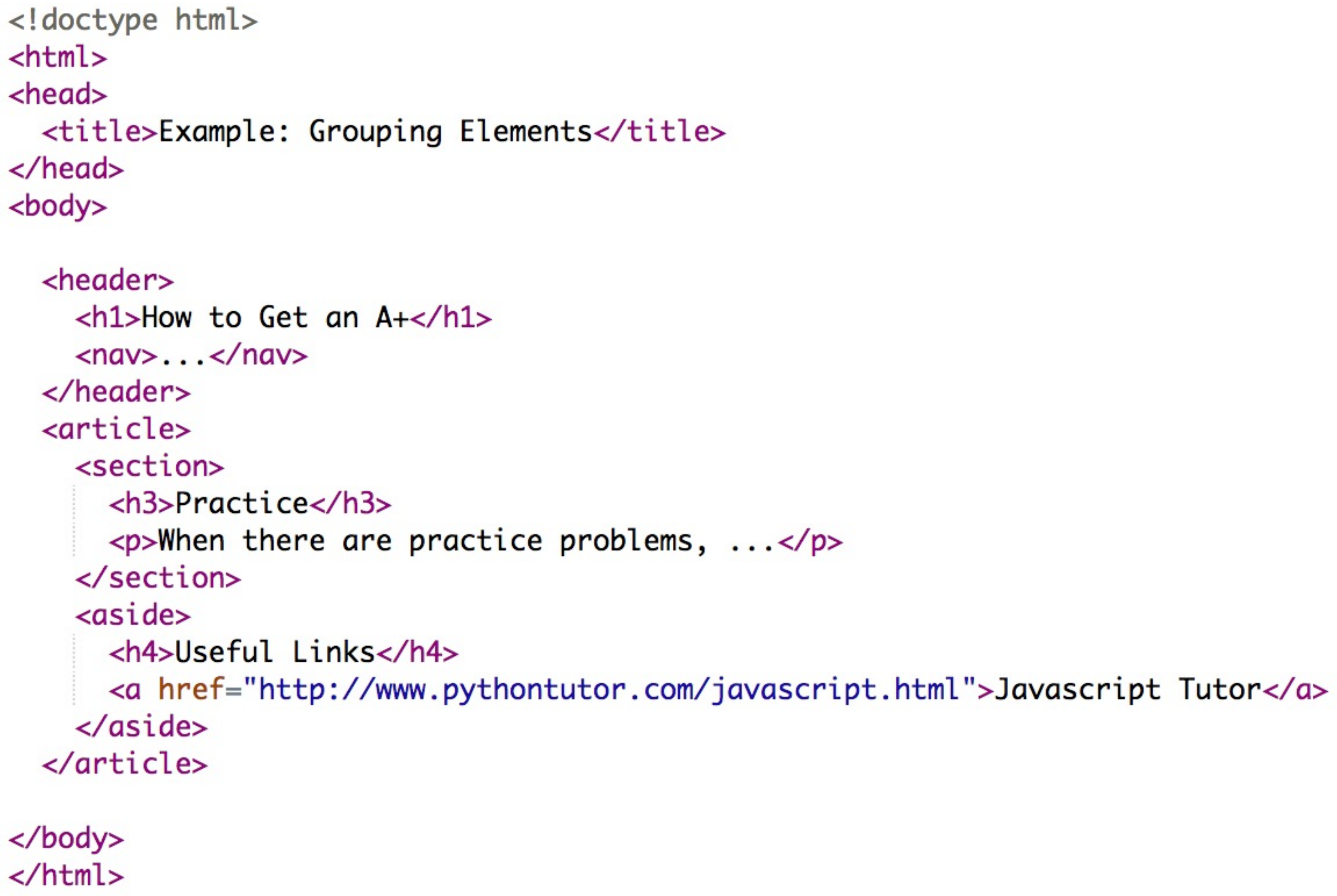
Grouping Elements

- Creates a parent or container element and a set of child elements
- Enables group to be styled together
- Can use any block or inline element or generic element
  - <div> : generic block element
  - <span> : generic inline element

Grouping Elements

- Semantic layout elements are block elements that associated meaning with group (useful for CSS selectors)

Some popular semantic layout elements are <header>, <footer>, <nav>, <article>, <aside>, <section>, <figcaption>



```html
<!doctype html>
<html>
<head>
  <title>Example: Grouping Elements</title>
</head>
<body>

  <header>
    <h1>How to Get an A+</h1>
    <nav>...</nav>
  </header>
  <article>
    <section>
      <h3>Practice</h3>
      <p>When there are practice problems, ...</p>
    </section>
    <aside>
      <h4>Useful Links</h4>
      <a href="http://www.pythontutor.com/javascript.html">Javascript Tutor</a>
    </aside>
  </article>

</body>
</html>
```

HTML Style

- Tags
  - Use lowercase for names
  - Use indentation to reflect hierarchy
  - Always close tags
- Use attribute name="value" format for attributes
- Use blank lines to break up documents into closely connected regions
- Use comments to describe purpose of regions

HTML Best Practices

- Use specialized controls or input validation where applicable
- Always include elements of HTML starter document
- Use label or placeholder for labeling controls
- Use alt to make images accessible