



Programming Languages for Web Applications

# HyperText Markup Language

# The Basic Idea

# The Tardis Tales

A collection of Doctor Who themed fan fiction, centered around the 10<sup>th</sup> and 11<sup>th</sup> doctors and their companions.

## Story 1 – The two doctors

• • •

# Your Content

```
+ <html>
    <body>
        <p> <nav>
            <h1>
        <h2> <a>
            <table>
                <b> <u> <i>
                    <blink>
                    <img>
=
```

## The Tardis Tales

A collection of Doctor Who themed fan fiction, centered around the **10<sup>th</sup>** and **11<sup>th</sup>** doctors and their companions

- Story 1 - The two doctors

# Your Website

# What is HTML?

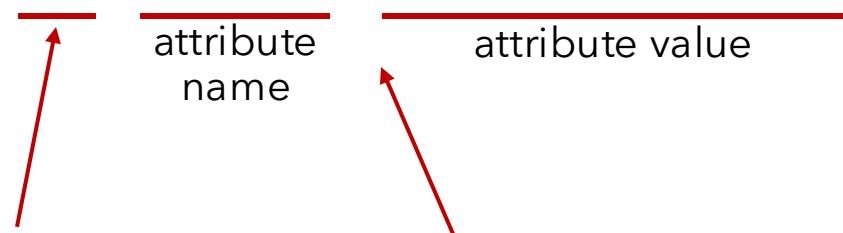
HyperText Markup Language

- Language for describing **structure of a document**
- A **text file** containing **markup tags** (or elements)
- Markup tags tell the web browser *how* to display the page
- HTML provides hierarchy of elements
- Can be created using a simple text editor, HTML editor (WYSIWYG), or IDE

May also describe  
structure of the  
page (semantics)

# HTML Elements / Tags

```
<a href="page.html">This is a link.</a>
```



Start an anchor element

Opening tag begins an HTML element.

Opening tags **must** have a corresponding closing tag.

Set the link target location (`href`) to `page.html`

HTML **attributes** are name-value pairs that provide additional information or context about the contents of an element.



End the anchor element

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

# HTML Elements / Tags

```

```

## Void elements

Some HTML elements  
do not have any  
contents.

Closing / **used to be required** in XHTML and some variants.

Closing / **optional and discouraged** in HTML5

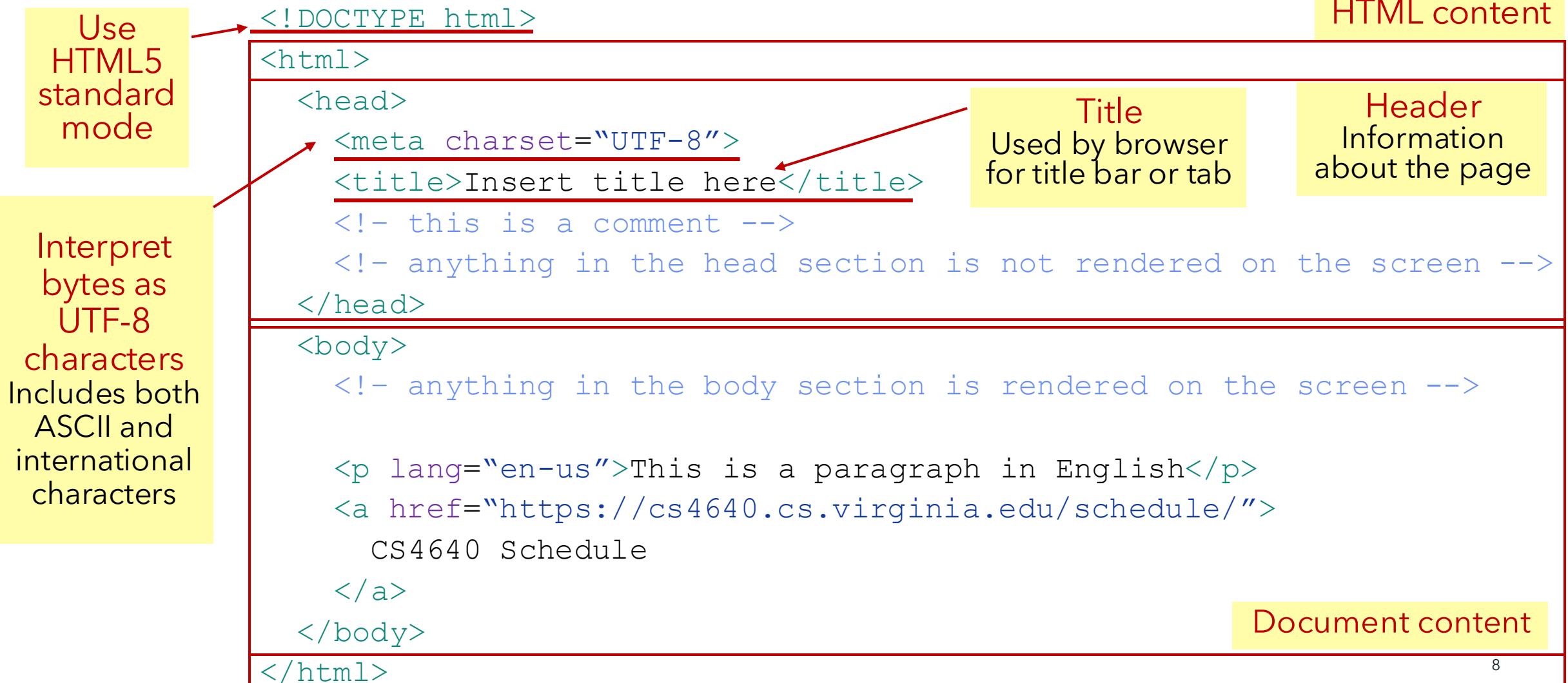
# HTML Contents

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	<pre>&lt;meta name="viewport" content="width=device-width, initial-scale=1"&gt;</pre>
Flow	Text and all elements that can appear as content in the body of an HTML document	<pre>&lt;body&gt;   &lt;h1&gt;Heading&lt;/h1&gt;   &lt;p&gt;Some content...&lt;/p&gt; &lt;/body&gt;</pre>
Sectioning	Used to structure the content of a web page and to help with layout	<pre>&lt;section class="highlight col"&gt;   Some content ... &lt;/section&gt;</pre>

# HTML Contents

Type	Description	Example
Phrasing	Elements for marking up content within a paragraph element such as text and typography	<pre>&lt;p&gt;   &lt;b&gt;Emphasized text&lt;/b&gt;   and some normal text &lt;/p&gt;</pre>
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	<pre>&lt;h1&gt;Main heading&lt;/h1&gt; &lt;h2&gt;Sub-heading&lt;/h2&gt;</pre>
Embedded	Embedded content includes media, such as video, audio, and images	<pre>&lt;img src="media/monster.png" alt="A cute monster image" width="80%" /&gt;</pre>
Interactive	Elements that a user can interact with such as media elements with controls, form inputs, buttons, and links	<pre>&lt;input type="text" name="username" required /&gt; &lt;input type="password" name="pwd" required /&gt;</pre>

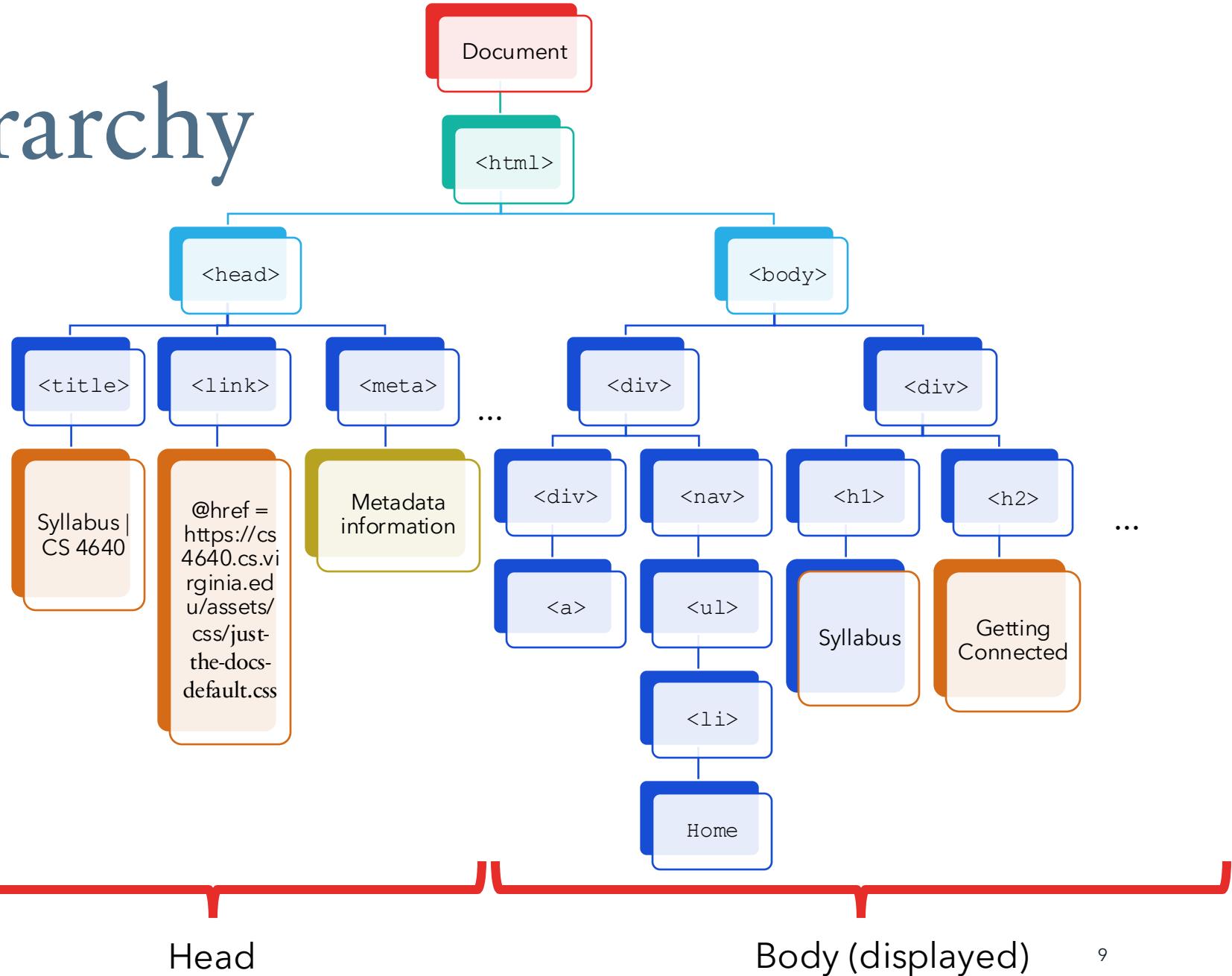
# HTML Example



# HTML Hierarchy

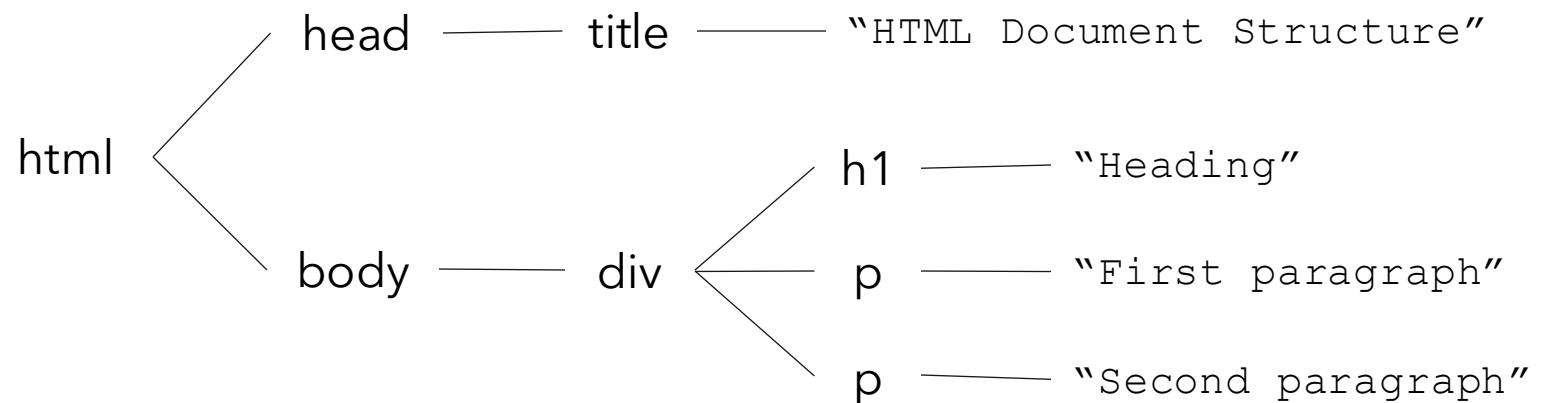
The screenshot shows a web browser window titled "Syllabus | CS 4640". The page content includes:

- A sidebar with links: Home, Syllabus, FAQ, Office Hours, Schedule, Project, Homework, Resources.
- The main title: "Syllabus".
- "Getting Connected" section with contact info for Robbie Hott.
- "Course Meetings" section with times and location.
- "Course Overview" section with a note about the course's focus on web development.



# HTML Hierarchy - 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>
```



# Body Text

```
1 <!doctype html>
2 <html>
3 <head></head>
4 <body>
5
6 <h1>Level 1 Heading</h1>
7 <h2>Level 2 Heading</h2>
8 <h3>Level 3 Heading</h3>
9 <h4>Level 4 Heading</h4>
10 <h5>Level 5 Heading</h5>
11 <h6>Level 6 Heading</h6>
12
13 Text can be made <b>bold</b> and <i>italic</i>,
14 or <sup>super</sup> and <sub>sub</sub> scripts.
15 White space collapsing removes all sequences of
16 two more spaces and line breaks, allowing
17 the markup to use tabs and whitespace for
18 organization.
19 Space can be added with &nbsp; &nbsp; &amp;nbsp;
20 <br />
21 <br />New line can be added with &lt; br /&gt;
22
23 <p>A paragraph consists of one or more sentences
24 that form a self-contained unit of discourse.
25 By default, a browser will show each paragraph
26 on a new line.
27 </p>
28
29 <hr />
30 Text can also be offset with horizontal rules.
31
32 </body>
33 </html>
34
```

## 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 <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;

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

- 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

```
<!doctype html>
<html>
<head>
</head>
<body>
Absolute link
<br />
Relative URL
<br />
Email Prof Hott
<br />
Open in new tab/window
<br />
Navigate to HTML anchor idName on this page
</body>
</html>
```

[Absolute link](https://cs4640.cs.virginia.edu)

[Relative URL](schedule.html)

[Email Prof Hott](mailto:jrhott@virginia.edu)

[Open in new tab/window](schedule.html)

[Navigate to HTML anchor idName on this page](#idName)

# Embeds: Image, Audio, Video

- HTML has built-in support for:
  - Images: <img> - .jpg, .png, .gif, .svg, etc
  - Audio: <audio> - .mp3, webm
  - Videos: <video> - .mp4, webm
- Always use an alt attribute on images to ensure accessibility

```

```

# Embeds: Image, Audio, Video

- Important attributes for <video> elements
  - 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

```
<video width="400" controls>
    <source src="doctor_who.mp4" type="video/mp4" />
    Your browser does not support HTML5 video.
</video>
```

# Tables

```
1 <!doctype html>
2 <html>
3 <head>
4   <title>Example: Table</title>
5 </head>
6 <body>
7   <table border="2" cellspacing="2" bgcolor="lightyellow" width="70%" align="center">
8     <tr>
9       <th>&nbsp;</th>
10      <th>Monday</th>
11      <th>Tuesday</th>
12      <th>Thursday</th>
13    </tr>
14    <tr>
15      <th>1pm-2pm</th>
16      <td rowspan="2">Intro to Programming</td>
17      <td>Calculus</td>
18      <td>&nbsp;</td>
19    </tr>
20    <tr>
21      <th>2pm-3pm</th>
22      <td>&nbsp; <!-- why &nbsp; here ? -->
23      <td>Physics</td>
24    </tr>
25  </table>
26 </body>
27 </html>
```

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

rowspan



# Forms

Send form data to  
`cs4640.cs.virginia.edu/formHandler.php`

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

```
<form action="https://cs4640.cs.virginia.edu/formHandler.php" method="post" >  
    Username: <input type="text" name="username" value="blank" /> <br />  
    Password: <input type="password" name="pwd" /> <br />  
    <input type="hidden" name="sectionid" value="002" />  
    <input type="submit" value="Submit" />  
</form>
```

Username:	<input name="username" type="text" value="blank"/>
Password:	<input name="pwd" type="password"/>
<input type="button" value="Submit"/>	

Transfer method

Method attribute specifies how data is transmitted to server.

`method="get"` sends data appended to URL.

`method="post"` sends data in body of the request

- 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    <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:

Text area:

Checkbox:  Checked  Unchecked

Drop down list box:

Multiple select Box:

File input box:  No file selected.

Image button: 

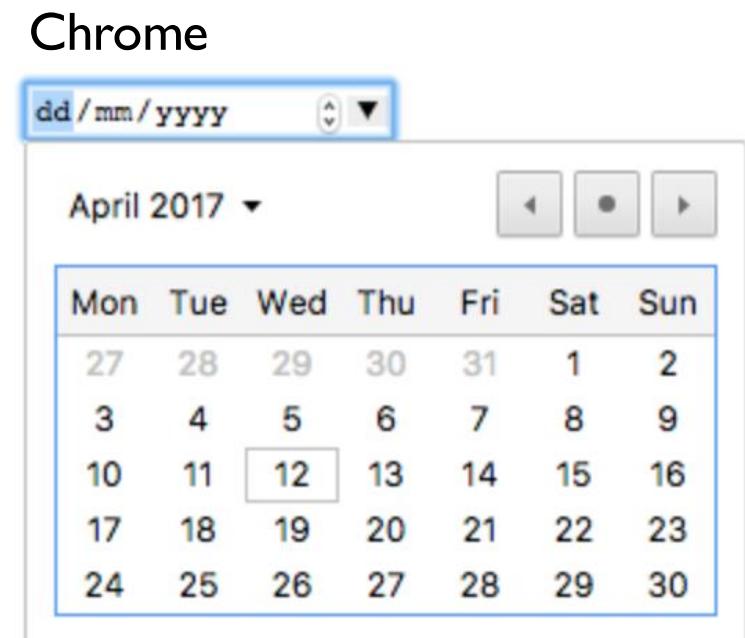
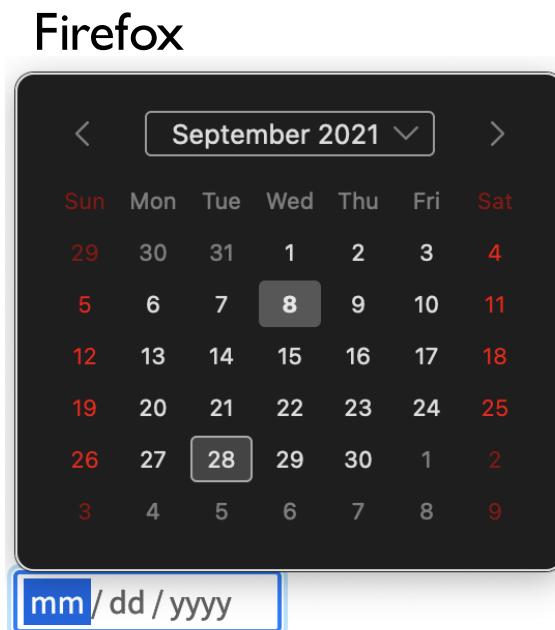
Button:

Range input:

# Specialized Controls

- date: may appear differently depending on browser or operating system

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



# Specialized Controls

- time: may appear differently depending on browser or operating system



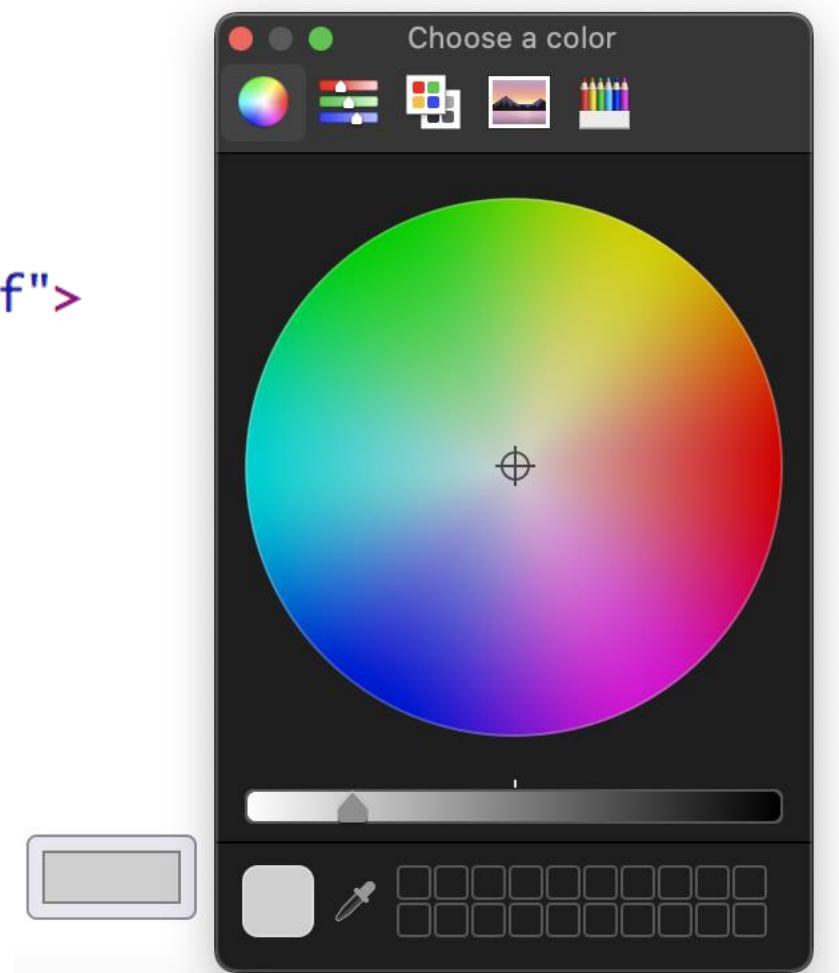
- number: can specify min and max values



# Specialized Controls

- color: allow the user to choose colors

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



# Labeling Inputs

- Placeholder suggestions can be given inside text boxes

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

Input box:

Disappear after the user begins typing

Input box:

- Labels **should** be attached to inputs (accessibility)

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

Label on input box:

# Validating Inputs (HTML)

Certain inputs will display errors on invalid input immediately, making it easier for users 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 (HTML)

Certain inputs will display errors on invalid input immediately, making it easier for users to fix errors

- 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

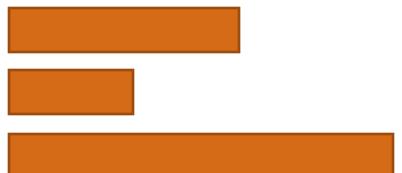
```
<label>Readonly text:  
  <input type="text" readonly />  
</label>
```

Readonly text:

# Block and Inline Elements

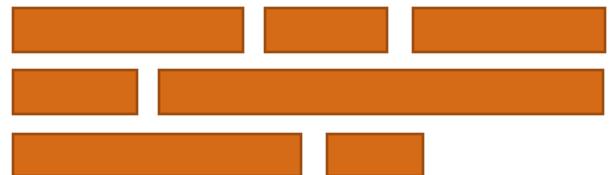
## Block elements

- 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>



# Grouping Elements

- Elements can be grouped with generic or semantic elements
  - Creates a “parent” or “container” for a set of child elements
  - Useful for styling elements together or referencing elements in scripts
- Generic elements:
  - `<div>` : generic block element
  - `<span>` : generic inline element

# Grouping Elements

- Semantic block layout elements are associated with meaning
  - Popular examples:
    - <header>, <footer>, <nav>, <article>, <aside>, <section>, <figcaption>

```
<!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>
```

# Best Practices: Styling HTML

- 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
- Indent your code well based on tree-structure

# Best Practices: HTML

- Use specialized controls or input validation where applicable
- Use label or placeholder for labeling controls
- Use alt to make images accessible