HTML

Course Roadmap

Web Client

Request

Response

Web Server

Frontend development

- HTML for page structure
- CSS for styling
- JavaScript for interaction

Backend development

- Web API
- Data Management

Outline

- Introduction to HTML
- Page Structure
- Tables
- Media tags
- Forms

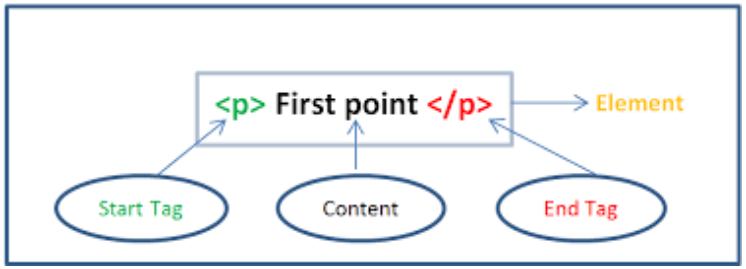
Introduction to HTML

HTML, CSS & JavaScript

- **HTML**
  - HTML stands for Hyper Text Markup Language
  - An HTML file is a text file containing markup tags
  - Describe the **content** & the **logical** **structure** of a page using markup tags
- HTML page consists of a *base HTML-file* which may includes *several referenced resources* such as:
  - **CSS** is a style sheet language used to control the **presentation and formatting** of an HTML document
  - **Javascript** used for client side scripting of **behavior/ functionality** such as validation, animation and partial page refresh (by asynchronously getting content from the server)
  - Images, audio files, etc.

Html page has head and body

- HTML uses tags to differentiate between document content
- Tags are enclosed in angle brackets **< … >**
- Generally come in pairs <tag> … </tag>



Page settings

```
<head>
    <title>Page Title</title>
    <meta name="description"
    content="This is an
    example.">
</head>
```

Page Content

```
<body>
  <h1>Heading 1</h1>
  <h2>Sub heading 2</h2>
  <h3>Sub heading 3</h3>
  <p>First paragraph</p>
  <p>Second paragraph</p>
</body>
```

HTML – Example

```
<!DOCTYPE html>
<html>
 <head>
  <title>HTML Example</title>
   <meta charset="UTF-8">
   <script src="script.js"></script>
   <link rel="stylesheet" href="style.css">
 </head>
 <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <p>This is a paragraph</p>
    <div>This is a div</div>
 </body>
</html>
```

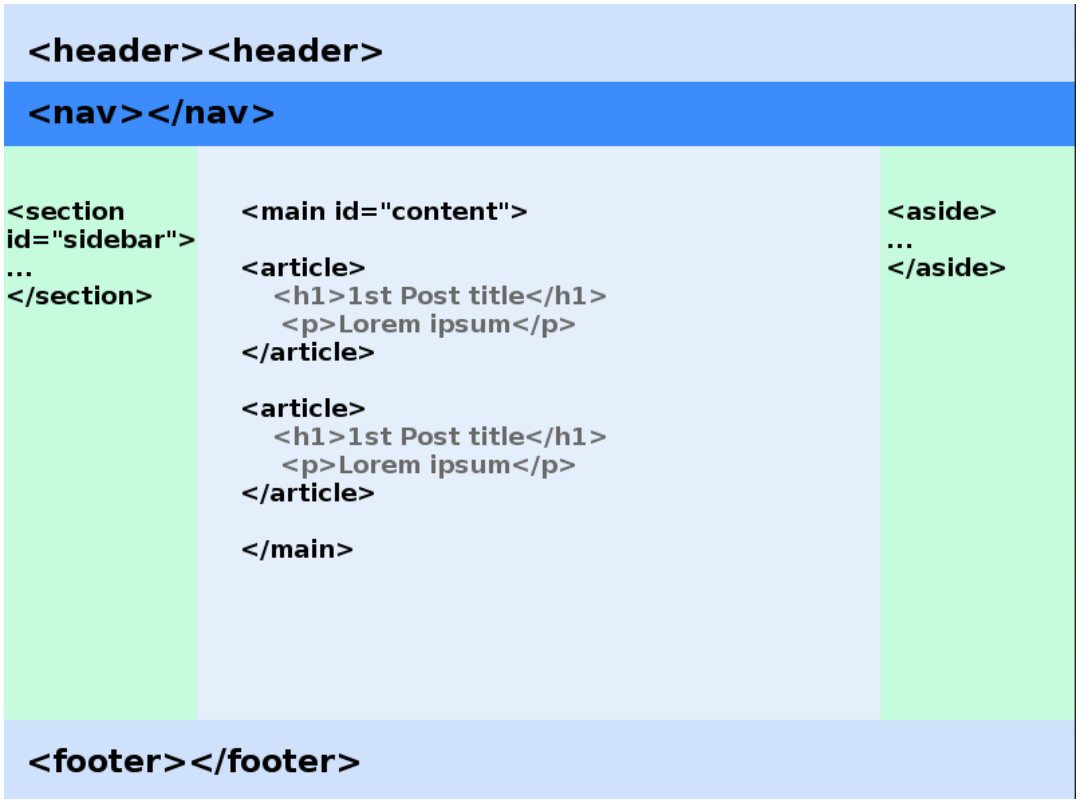
HTML key capabilities

- Define the logical structure of the HTML document
- Collect input from users using Forms
- Display data using tables
- Embed media (e.g., audio and video) into HTML documents
- Other capabilities such drawing graphics in canvas, etc.

Page Structure

Page Structure

<header>
<nav>
<main>
<section>
<article>
<aside>
<footer>



- We can use Semantic Tags to **define the logical structure of the page**
- We can use **css** to arrange elements into the desired layout

Which one to use? http://html5doctor.com/downloads/h5d-sectioning-flowchart.png

Header, Nav, Aside & Footer

<header>

a container for introductory content, logo or a set of navigational links

<nav>

contains primary navigation (frequently inside a header)

<footer>

contains information about copyright, contact info, facebook/twitter links etc.

<aside>

may contain sidebars, pullquotes, ads, etc.

(can be removed without reducing the meaning of the main content)

Article & Section

- `<article>`

defines self-contained content that would make sense when read on its own (e.g., blog post). e.g., https://www.w3schools.com/tags/tag_main.asp

- `<section>`

a thematic grouping of content. It is used to **divide either the page into different subject areas**, or to section an individual article.

Think of it like a newspaper: You will find **many articles in each section**. e.g., Film review articles in the Entertainment section.

<main> contains blog entries. Each blog entry would make sense when read on its own

```
<main>
    <article>
      <!-- first blog post -->
    </article>

    <article>
      <!-- second blog post  -->
    </article>

    <article>
      <!-- third blog post -->
    </article>
</main>
```

An article can be structured using sections. For example, a blog post can have an introduction, a body content and a summary.

```
<article>
  <section id="introduction">
  </section>

  <section id="content">
  </section>

  <section id="summary">
  </section>
</article>
```

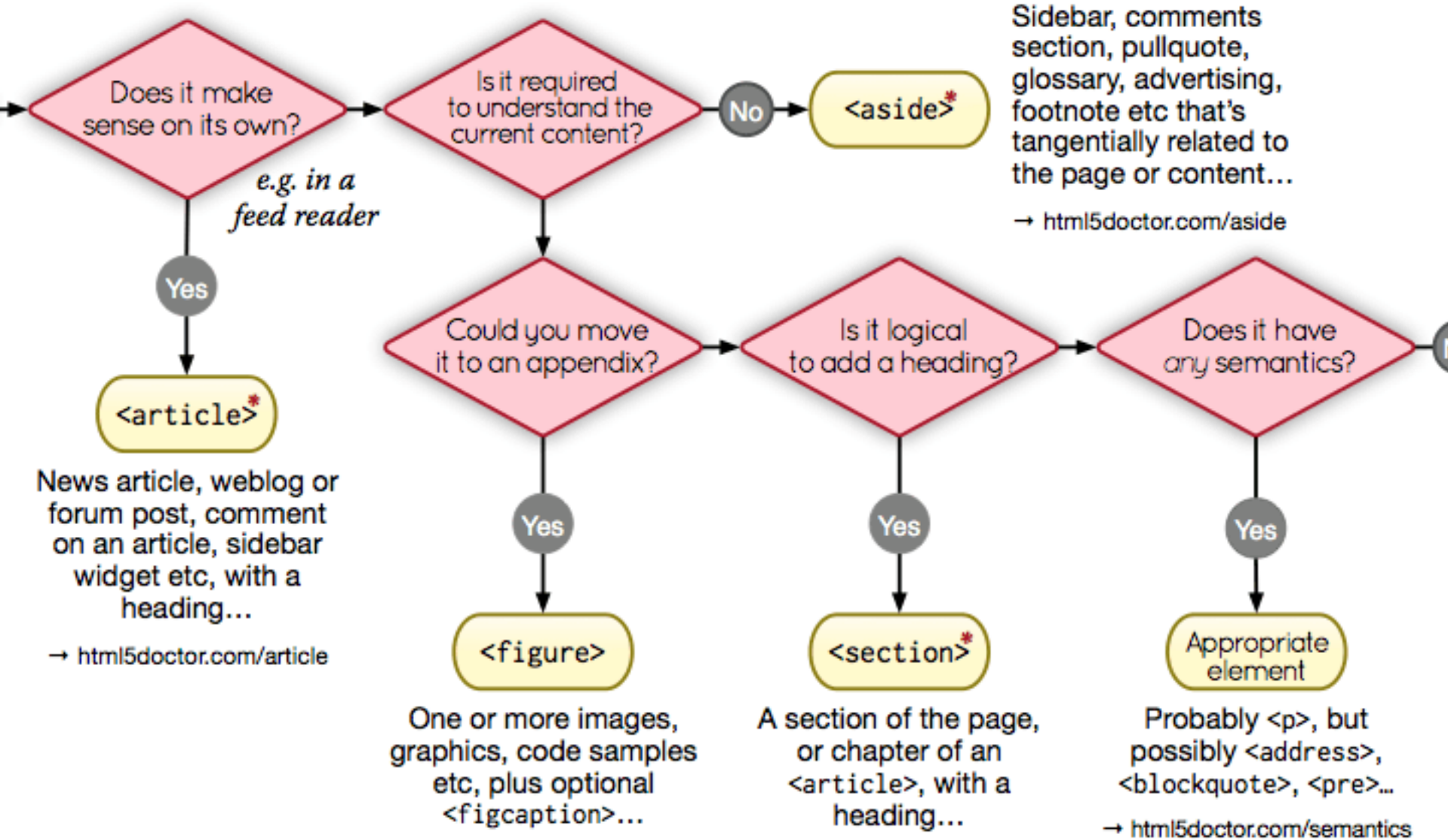
vs. Semantic tags

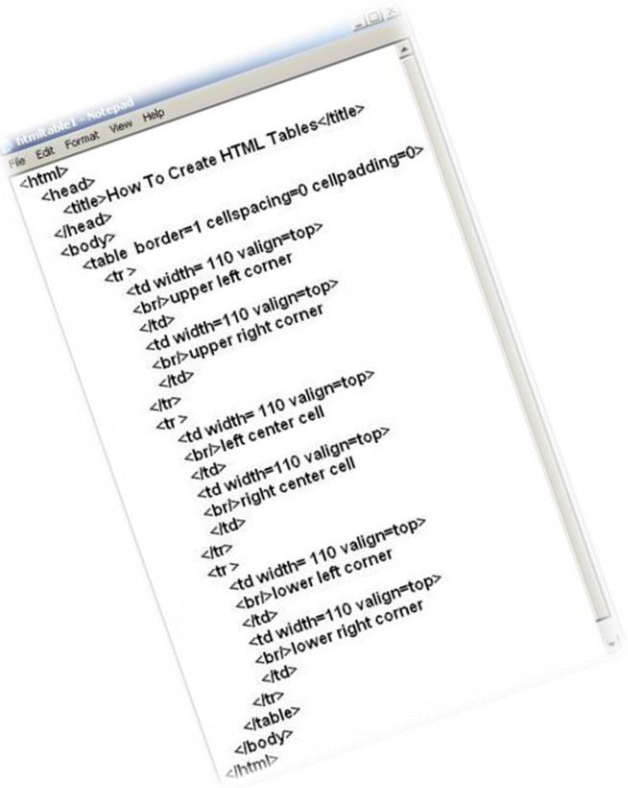
  - a ** ** element is used a generic container of other elements
  - semantic tags are designed to describe the page content in a more meaningful way

- Use semantic tags to **"mark up" content in a meaningful way**
  - The elements that you choose to mark up your content **should *describe* the content**
- **Mark your document elements based on on their role**
  - If you need an element to describe a **paragraph of content** then use a **<p>**
  - **If you need a generic container then use a **

By @riddle & @boblet
www.html5doctor.com

ock of flow content
inline phrasing content)





Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data

HTML Tables

HTML Tables

- Tables represent tabular data
  - A table consists of one or several rows
  - Each row has one or more columns
- Tables comprised of several core tags:
  `<table></table>`: begin / end the table
  `<tr></tr>`: create a table row
  `<td></td>`: create a table column

Simple HTML Tables – Example

```
<table>
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td><img src="zip.gif"></td>
    <td><a href="lecturedemos.zip">
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```

Complete HTML Tables

- Table rows split into three semantic sections: header, body and footer
  - `<thead>` denotes table header and contains `<th>` elements, instead of `<td>` elements
  - `<tbody>` denotes collection of table rows that contain the data
  - `<tfoot>` denotes table footer but comes BEFORE the `<tbody>` tag

Complete HTML Table: Example

```
<table>

    <thead>
      <tr><th>Column 1</th><th>Column 2</th></tr>
    </thead>
    <tfoot>
      <tr><td>Footer 1</td><td>Footer 2</td></tr>
    </tfoot>
    <tbody>
      <tr><td>Cell 1.1</td><td>Cell 1.2</td></tr>
      <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
    </tbody>
 </table>
```

header

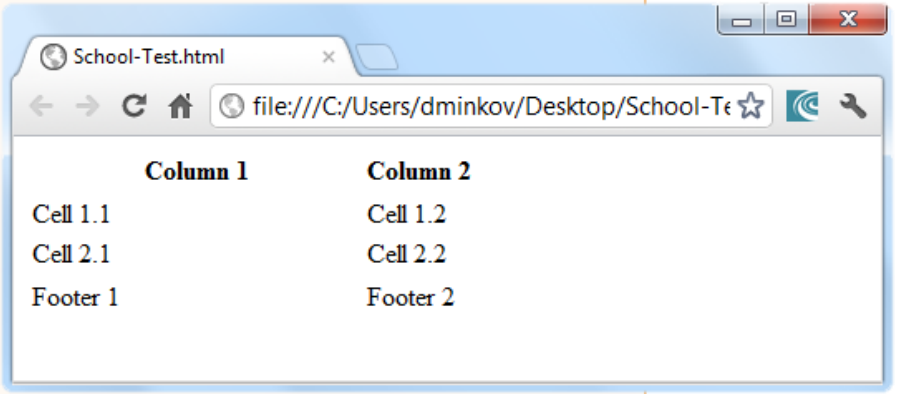
th

footer

Last comes the body (data)

Complete HTML Table: Example (2)

```
<table>
    <thead>
      <tr><th>Column 1</th><th>Column 2</th></tr>
    </thead>
    <tfoot>
      <tr><td>Footer 1</td><td>Footer 2</td></tr>
    </tfoot>
    <tbody>
      <tr><td>Cell 1.1</td><td>Cell 1.2</td></tr>
      <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
    </tbody>
</table>
```



Although the footer is before the data in the code, it is displayed last

Media Tags

Img / Audio / Video Tag

```
<img src="cat.jpg" alt="A black, brown, and white cat">
```

```
<h1>Video player with controls</h1>
<video controls width="360" height="240"
    src="http://www.w3schools.com/html/movie.mp4">
</video>

<h1>Audio player with controls</h1>
<audio controls
       src="http://download.quranicaudio.com/quran/mahmood_khaleel_al-husaree_doori/001.mp3">
</audio>
```

Forms

Forms

- Forms are used to collect input from the user and submitting it to a Web server
- A form can have many input elements each has a name and id
  - Name identifies the input when the form is submitted
  - Id is used to access the element from JavaScript or CSS

Text input

Textbox

- <input type="text" />

Password

- <input type="password"/>

Text Area

- <textarea />

Hidden

- <input type="hidden" />

HTML 5 input fields

- Input element enables multiple input types

<input type="**search**">	search box
<input type="**number**">	spinbox
<input type="**range**">	slider
<input type="**color**">	color picker
<input type="**tel**">	telephone number
<input type="**url**">	web address
<input type="**email**">	email address
<input type="**date**">	calendar date picker
<input type="**month**">	month
<input type="**week**">	week
<input type="**time**">	time
<input type="**datetime**">	date time
<input type="**datetime-local**">	local date and time

Selections

- Select, checkbox and radio enable pre-defined input

Select list

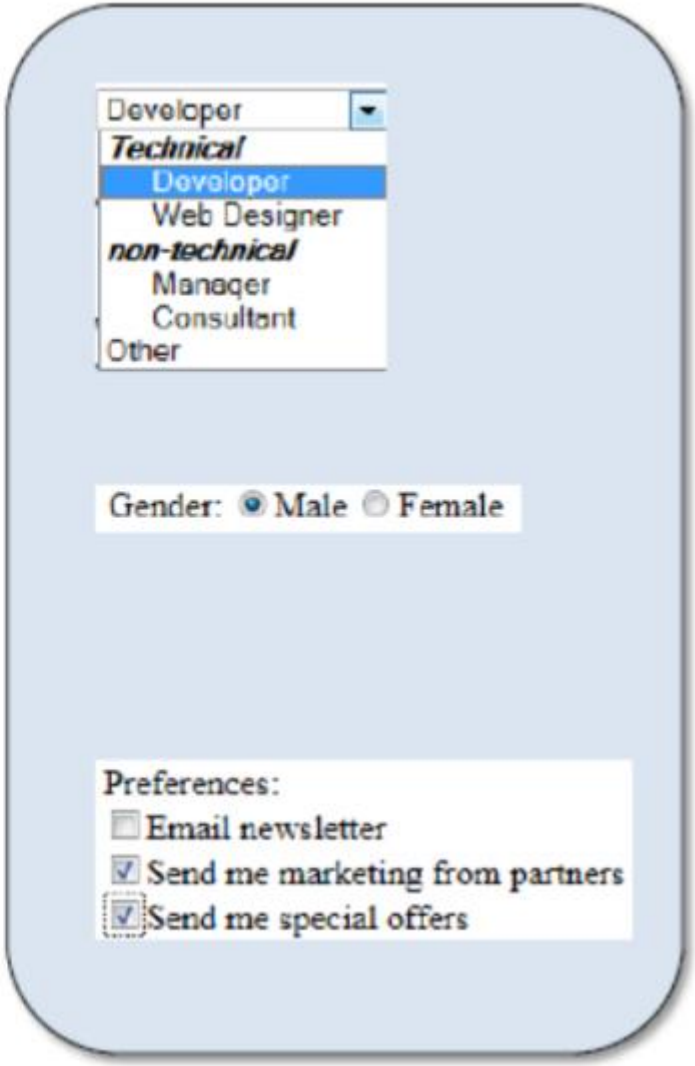
- Single/multi select

Radio buttons

- Grouped by name

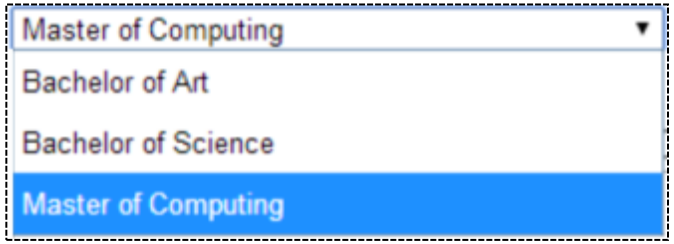
Check boxes

- Multiple allowed



Dropdown and List Examples

- **Dropdown**




```
<select name="degree" id="degree">
    <option value="BA">Bachelor of Art</option>
    <option value="BS">Bachelor of Science</option>
    <option value="MC" selected="selected">Master of Computing</option>
</select>
```

- **Multi selections list**

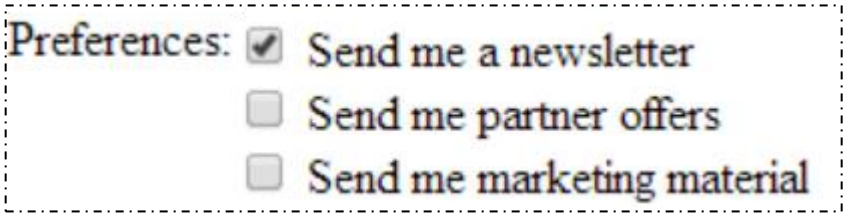
```
<label for="occupation">Occupation:</label>
<select id="occupation" name="occupation" multiple size="4">
    <optgroup label="Technical">
        <option label="Developer" value="developer"></option>
        <option label="Web Designer" value="designer"></option>
    </optgroup>
    <optgroup label="Non-technical">
        <option label="Consultant" value="consultant" selected></option>
        <option label="Manager" value="manager"></option>
    </optgroup>
    <option label="Other" value="other"></option>
</select>
```

Radio button and Checkbox Examples

- **Radio button**

```
<label style="text-align: left">
    <input type="radio" id="male" name="gender" value="male">Male
</label>
<label style="text-align: left">
    <input type="radio" id="female" name="gender" value="female" checked>Female
</label>
```

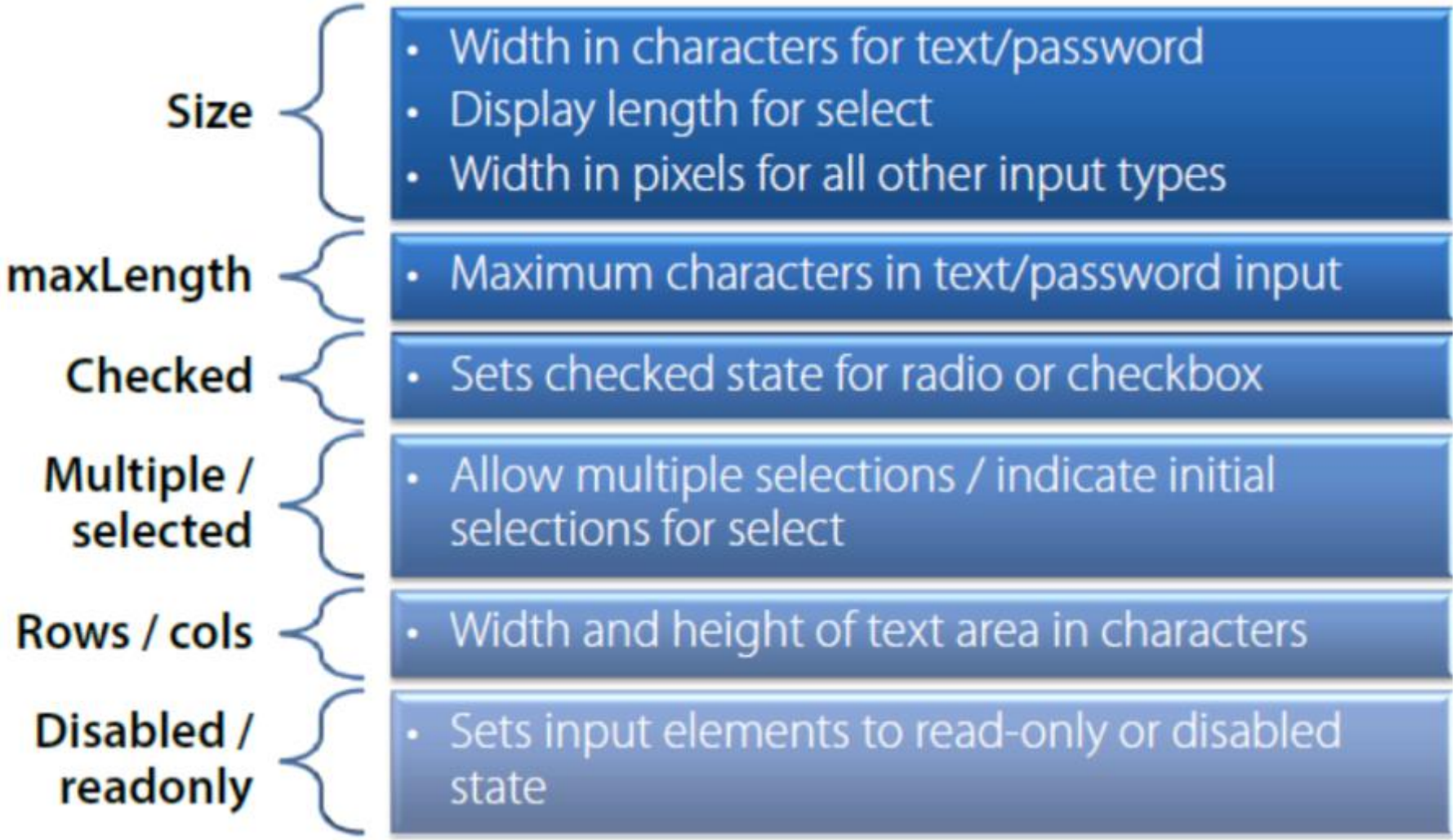
- **Checkbox**




```
<label>
    <input type="checkbox" id="newsletter" name="prefs" value="newsletter" checked>
    Send me a newsletter
</label><br/>
<label>
    <input type="checkbox" id="partners" name="prefs" value="partners">
    Send me partner offers
</label><br/>
<label>
    <input type="checkbox" id="marketing" name="prefs" value="marketing">
    Send me marketing material
</label><br/>
```

Input attributes

- Apply attributes to control **rendering**



Input commands

Reset

- Set inputs to original values

reset

Submit

- Submits the form to the server

submit

Button

- <input type="button"> = push button
- <button type="submit">

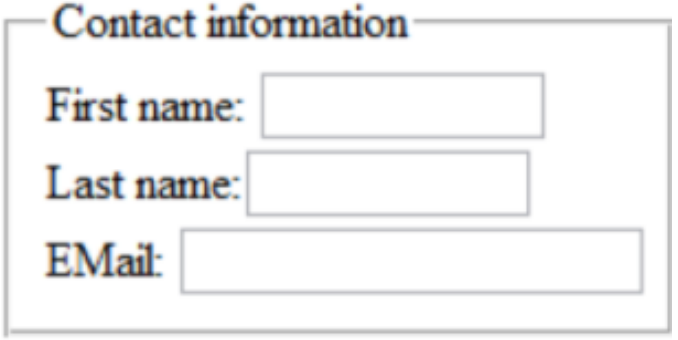
Button with content

Image

- Image button

Form organization

- **Labels**
  - Text explicitly associated with an input
  - Interaction with label moves focus to input
- **Fieldsets**
  - Groups form input fields
  - Optionally label the group



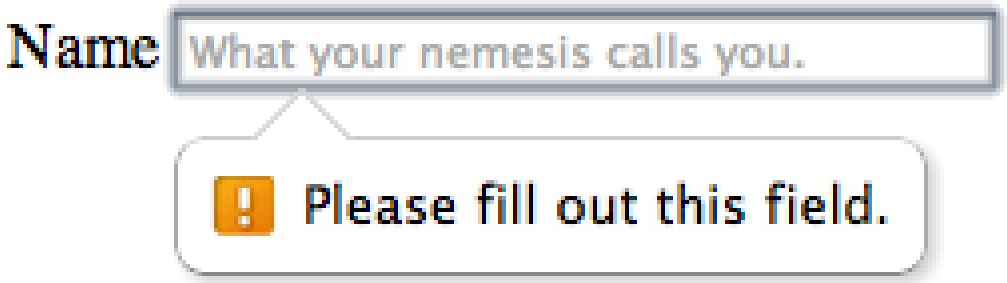
HTML 5 Input Fields

<input placeholder="Full Name">

- Disappears as the user types.
- NOT a replacement for a proper label

<input required>

- Validated by the browser

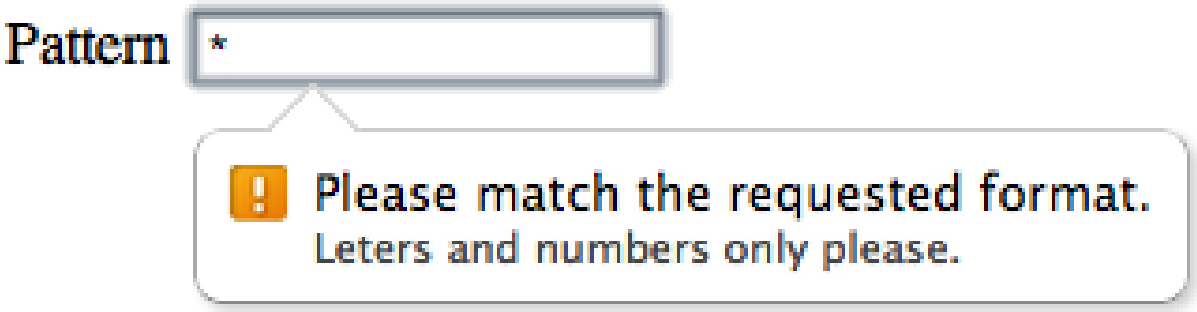


<input autofocus>

- Auto selects the first input field with autofocus
- Will scroll the page to give it focus.

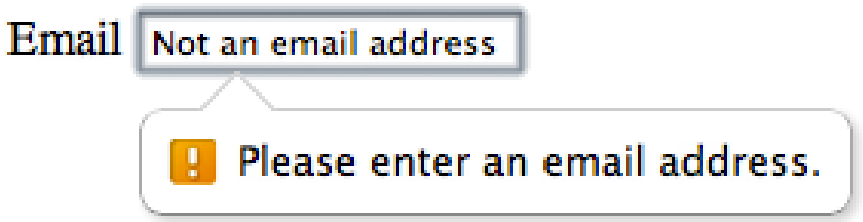
<input pattern="[a-zA-Z0-9]+" title="Letters and numbers only please">

- Matches a regular expression
- Only validates if something has been entered
- Error message is non-specific. Some browsers will use title attribute to explain
- Use the title attribute to add additional help text
  - This works with all the input types



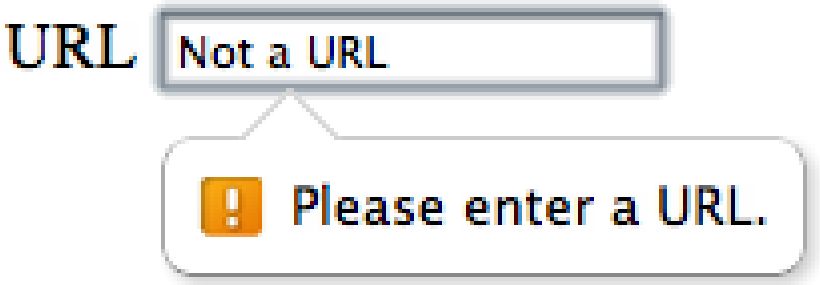
<input type="email">

- For email addresses
- Is validated as an email address
- Gives email keyboard



<input type="url">

- For urls
- Gives url keyboard
- Is loosely validated as a url
  - Use in combination with pattern if you want something specific



<input type="tel">

- For phone numbers
- Gives number pad
- Very loosely validated
  - Handy since the nice big number pad is handy for inputting any number so you can use it for anything else you like
  - Use with pattern if you have something specific in mind

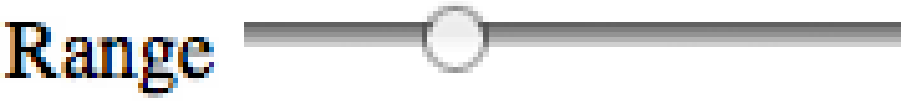
<input type="number">

- For numbers. Also called a "spinbox"
- Gives number keypad
- Special attributes:
  - min
  - max
  - step

Number 42

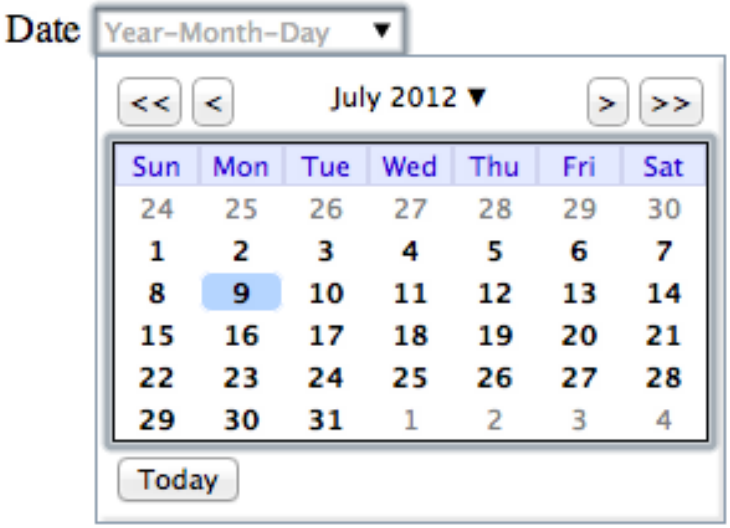
<input type="range">

- For numbers. Also called a "slider"
- Exact number not displayed to user
- Special attributes:
  - min
  - max
  - step



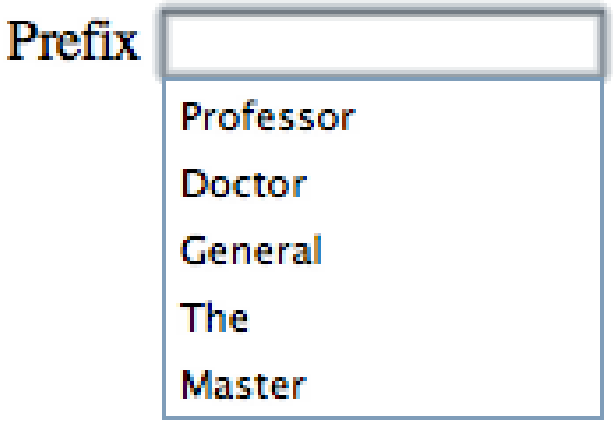
<input type="date">

- Displays a date picker
- Configurable formats:
  - type="date"
  - type="datetime"
  - type="datetime-local"
  - type="month"
  - type="week"
  - type="time"



```
<input type="text"
list="sources">
<datalist id="sources">
    <option>Professor</option>
    <option>Master</option>
</datalist>
```

- Text box with filtered list of suggestions
- Entire list isn't usually visible, appears as user types, filtered by what they've entered



References

- Mozilla Development Center HTML5
  - https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5
  - https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML
- HTML tutorial
  - http://www.w3schools.com/html/
- Cheat sheet
  - https://htmlcheatsheet.com/