

# HTML



# Course Roadmap



Web Client

Request

Response



Web Server

Frontend development

HTML for page content & structure



CSS for styling



JavaScript for interaction



Backend development

Web API

Web Pages

Data Management

NEXT.js



# Outline

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

# Introduction to HTML

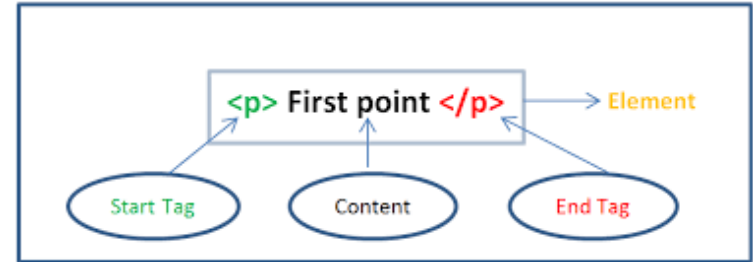


# HTML, CSS & JavaScript

- **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">
```

← Ensures proper display of text characters and symbols from all languages

```
<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>

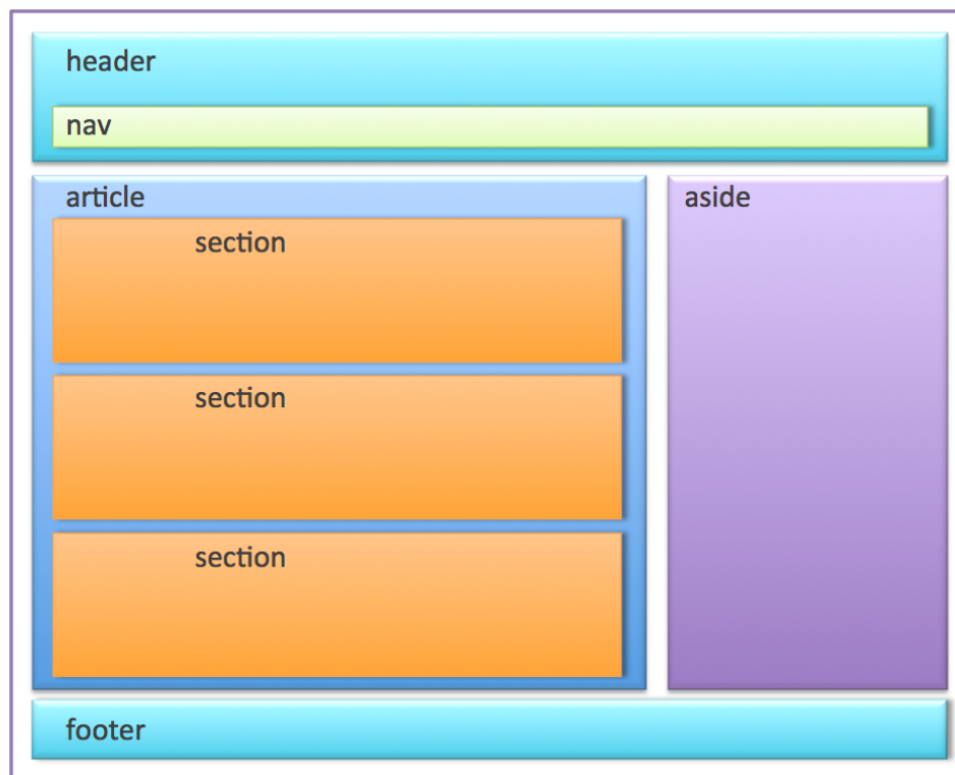
<main>

<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

# 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](https://www.w3schools.com/tags/tag_main.asp)

- **<section>**

represents a thematic grouping of content within a document. **Group related content** within a larger page, similar to chapters or sections in a book

- Has its own heading (<h1>-<h6>)

**<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>**

Here, the `<article>` represents the entire blog post. The `<section>` tags divide the post into sections for each city visited, and a nested `<section>` further divides the Tokyo section to focus on Shibuya

```
<article>
  <h1>My Trip to Japan</h1>
  <p>Introductory paragraph about the trip.</p>

  <section id="tokyo">
    <h2>Tokyo</h2>
    <p>Details about the time spent in Tokyo.</p>
    <section id="shibuya">
      <h3>Shibuya Crossing</h3>
      <p>Description of Shibuya Crossing.</p>
    </section>
  </section>

  <section id="kyoto">
    <h2>Kyoto</h2>
    <p>Details about the time spent in Kyoto.</p>
  </section>
</article>
```

```
<body>
  <header>...</header>

  <section id="about-us">
    <h2>About Us</h2>
    <p>Information about the company...</p>
  </section>

  <section id="our-services">
    <h2>Our Services</h2>
    <ul>
      <li>Service 1</li>
      <li>Service 2</li>
    </ul>
  </section>

  <section id="contact-us">
    <h2>Contact Us</h2>
    <form>...</form>
  </section>

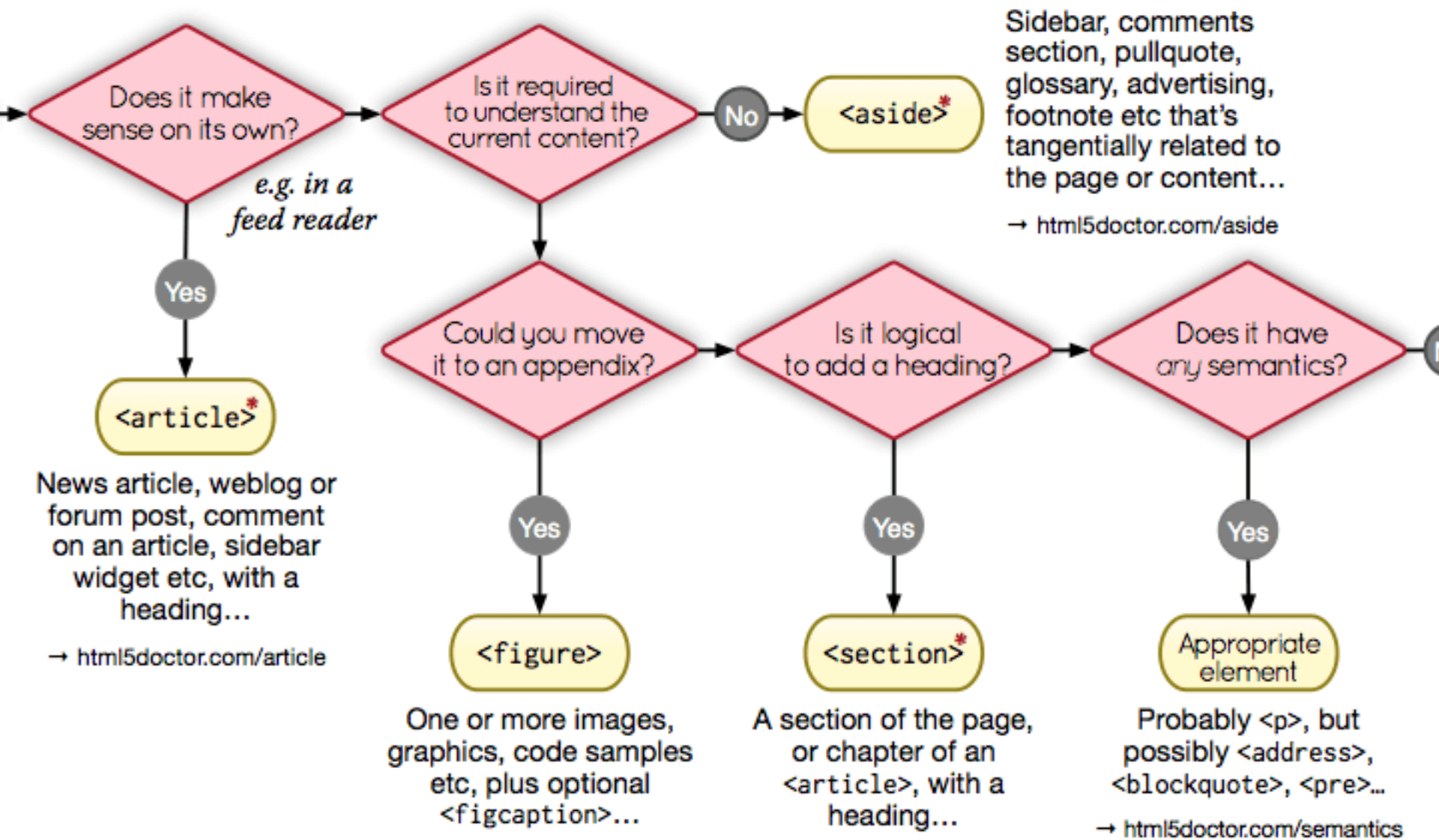
  <footer>...</footer>
</body>
```

```
<article class="product">
  <h1>Product Name</h1>
  <section id="product-description">
    <h2>Description</h2>
    <p>Detailed product description.</p>
  </section>
  <section id="product-specifications">
    <h2>Specifications</h2>
    <ul>
      <li>...</li>
    </ul>
  </section>
  <section id="customer-reviews">
    <h2>Customer Reviews</h2>
    <ul>
      <li>...</li>
    </ul>
  </section>
</article>
```

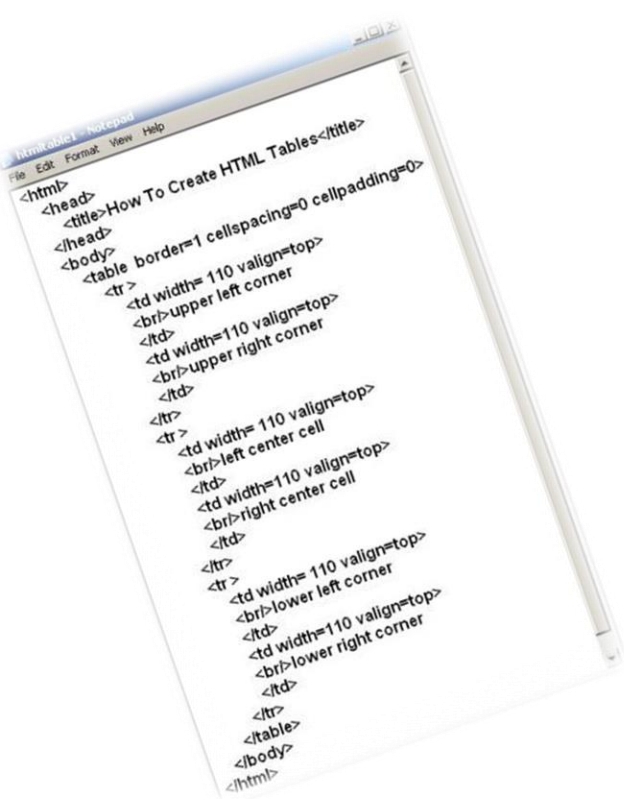
# <div> vs. Semantic tags

- a **<div>** element is used as 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 their role**
  - If you need an element to describe a **paragraph of content** then use a **<p>**
  - Make the page more accessible
  - **If you need a generic container then use a <div>**

Block 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></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td></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>`

header

th

`<tr><th>Column 1</th><th>Column 2</th></tr>`

`</thead>`

footer

`<tfoot>`

`<tr><td>Footer 1</td><td>Footer 2</td></tr>`

`</tfoot>`

Last comes the body (data)

`<tbody>`

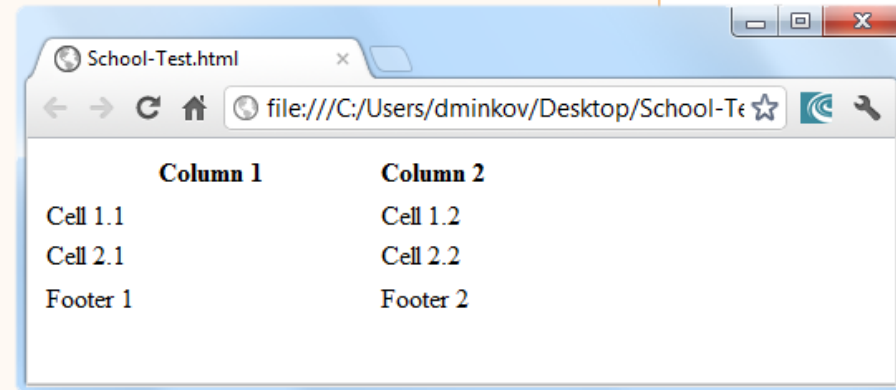
`<tr><td>Cell 1.1</td><td>Cell 1.2</td></tr>`

`<tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>`

`</tbody>`

`</table>`

# Complete HTML Table: Example (2)



A screenshot of a web browser window titled 'School-Test.html'. The address bar shows the file path 'file:///C:/Users/dminkov/Desktop/School-Te'. The browser displays a table with two columns, 'Column 1' and 'Column 2'. The table has three rows: a header row with 'Column 1' and 'Column 2', a body row with 'Cell 1.1' and 'Cell 1.2', and a footer row with 'Footer 1' and 'Footer 2'.

Column 1	Column 2
Cell 1.1	Cell 1.2
Cell 2.1	Cell 2.2
Footer 1	Footer 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

``

```
<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" />`



Hello forms

.....

Text areas can have  
more than one line.

# HTML 5 input fields

- Input element enables multiple input types

<code>&lt;input type="search"&gt;</code>	search box
<code>&lt;input type="number"&gt;</code>	spinbox
<code>&lt;input type="range"&gt;</code>	slider
<code>&lt;input type="color"&gt;</code>	color picker
<code>&lt;input type="tel"&gt;</code>	telephone number
<code>&lt;input type="url"&gt;</code>	web address
<code>&lt;input type="email"&gt;</code>	email address
<code>&lt;input type="date"&gt;</code>	calendar date picker
<code>&lt;input type="month"&gt;</code>	month
<code>&lt;input type="week"&gt;</code>	week
<code>&lt;input type="time"&gt;</code>	time
<code>&lt;input type="datetime"&gt;</code>	date time
<code>&lt;input type="datetime-local"&gt;</code>	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

Developer ▼

*Technical*

Developer

Web Designer

*non-technical*

Manager

Consultant

Other

Gender: ☒ Male ☐ Female

Preferences:

☐ Email newsletter

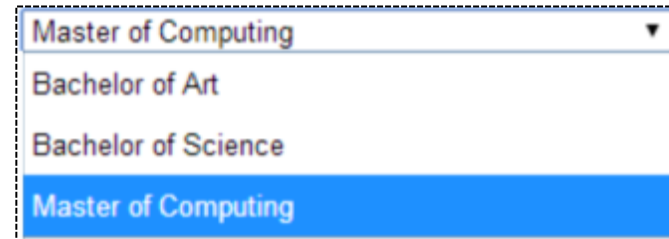
☒ Send me marketing from partners

☒ Send me special offers

# 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>
```



Master of Computing  
Bachelor of Art  
Bachelor of Science  
Master of Computing

- 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>
```



Occupation: Developer  
Web Designer  
Non-technical  
Consultant

# Radio button and Checkbox Examples

- Radio button

Gender: ☐ Male ☒ Female

```
<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

Preferences: ☒ Send me a newsletter  
☐ Send me partner offers  
☐ Send me marketing material

```
<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**

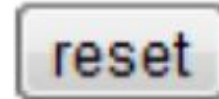
Size	<ul style="list-style-type: none"><li>• Width in characters for text/password</li><li>• Display length for select</li><li>• Width in pixels for all other input types</li></ul>
maxLength	<ul style="list-style-type: none"><li>• Maximum characters in text/password input</li></ul>
Checked	<ul style="list-style-type: none"><li>• Sets checked state for radio or checkbox</li></ul>
Multiple / selected	<ul style="list-style-type: none"><li>• Allow multiple selections / indicate initial selections for select</li></ul>
Rows / cols	<ul style="list-style-type: none"><li>• Width and height of text area in characters</li></ul>
Disabled / readonly	<ul style="list-style-type: none"><li>• Sets input elements to read-only or disabled state</li></ul>



# Input commands

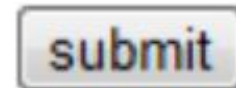
## Reset

- Set inputs to original values



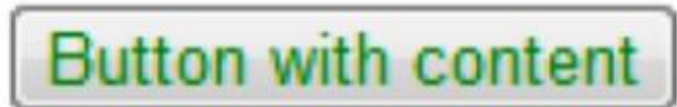
## Submit

- Submits the form to the server



## Button

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



## Image

- Image button



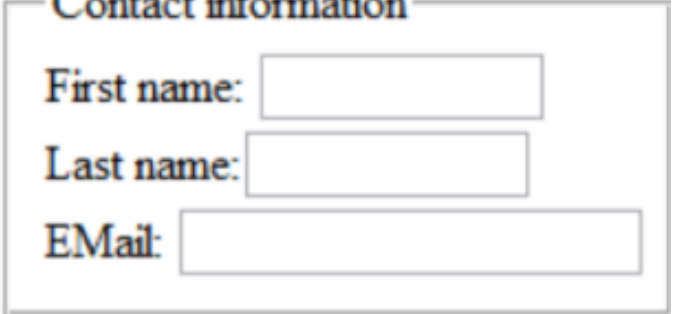
# Form organization

- **Labels**

- Text explicitly associated with an input
- Interaction with label moves focus to input

- **Fieldsets**

- Groups form input fields
- Optionally label the group



The image shows a form titled "Contact information" enclosed in a rectangular border. Inside the border, there are three rows of text and input fields. The first row is "First name:" followed by a rectangular input box. The second row is "Last name:" followed by a rectangular input box. The third row is "EMail:" followed by a rectangular input box. The text labels are in a dark blue font, and the input boxes are white with a thin grey border.

# HTML 5 Input Fields

## <input placeholder="Full Name">

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

Name

## <input required>

- Validated by the browser

Name

! Please fill out this field.

## <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

Pattern

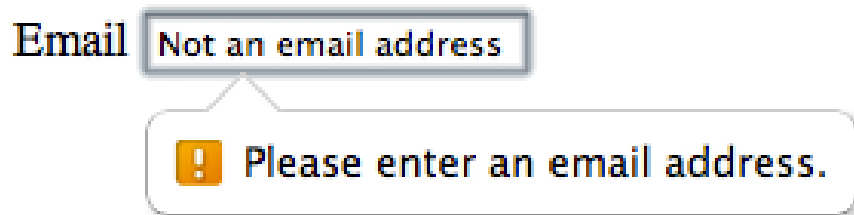
\*



Please match the requested format.  
Letters and numbers only please.

# <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

URL

Not a URL



Please enter a URL.



# <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



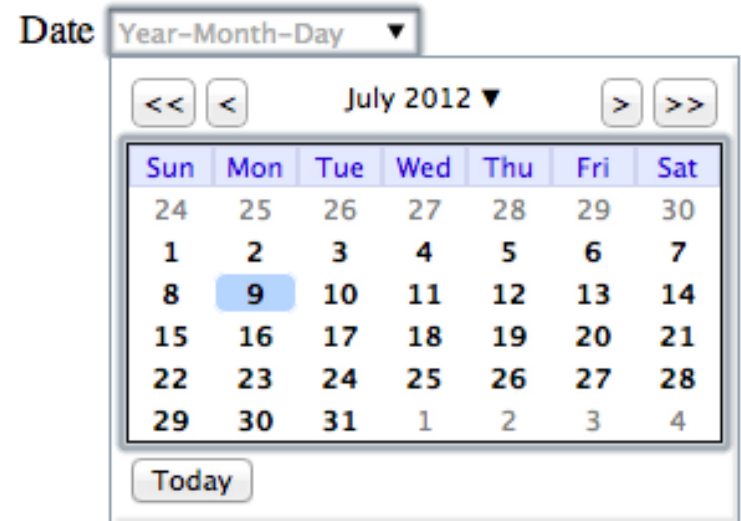
# <input type="range">

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



# <input type="date">

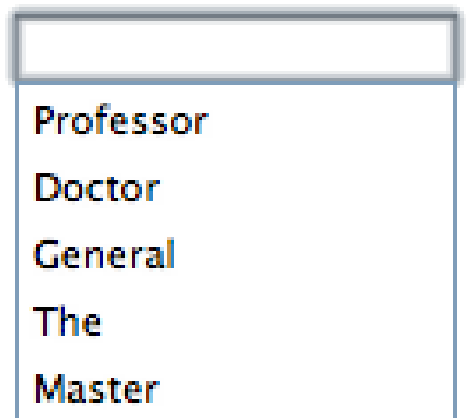
- Displays a date picker
- Configurable formats:
  - type="date"
  - type="datetime"
  - type="datetime-local"
  - type="month"
  - type="week"
  - type="time"



```
<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

Prefix



Prefix
Professor
Doctor
General
The
Master

# Submitting Form

- The Form should specify the **Method** and **Action**

**Method:** Specifies how the form data is sent to the server

**Action:** Specifies the URL where the form data is sent for processing.

**GET:** Appends data to the URL as query parameters (e.g., for search or non-sensitive data)

**POST:** Sends data in the request body (e.g., for sensitive data like passwords)

# How Form Data is Formatted when Submitted?



Form data is converted into a key-value pair format.



Characters that are not safe for URLs (like spaces or special symbols) are URL encoded (i.e., percent-encoding)

Example: A space ( ) becomes %20 or +



GET Method: Data is appended to the URL as query parameters

?key1=value1&key2=value2  
/search?query=qatar%20university



POST Method: Data is sent in the HTTP request body

key1=value1&key2=value2

# URL Encoding of Form Data

- According [RFC 1738](#), the characters allowed in URL are alphanumeric [0-9a-zA-Z] and the special characters \$-\_.+!\*'()
- Unsafe characters should be encoded, e.g.,

[/search?q=qatar%20university](#)

Commonly encoded values:

ASCII Character	URL-encoding
space	%20
!	%21
"	%22
#	%23
\$	%24
%	%25
&	%26

# 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](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML)
- HTML tutorial
  - <http://www.w3schools.com/html/>
- Cheat sheet
  - <https://htmlcheatsheet.com/>