

HTML

# About HTML

- Stands for HyperText Markup Language
- Is a language for describing web pages
- Current Standard version HTML 5
- World Wide Web Consortium (W3C)
- WHATWG (Web Hypertext Application Technology Working Group)

# First HTML page

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#):

### [What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), X11 [Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#).)

### [Technical](#)

Details of protocols, formats, program internals etc

### [Bibliography](#)

Paper documentation on W3 and references.

### [People](#)

A list of some people involved in the project.

### [History](#)

A summary of the history of the project.

### [How can I help ?](#)

If you would like to support the web..

### [Getting code](#)

Getting the code by [anonymous FTP](#), etc.

# Evolution of HTML

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

# HTML DOCTYPE

- **HTML 5**
  - `<!DOCTYPE html>`
- **HTML 4.01 Strict**
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- **HTML 4.01 Transitional**
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
- **HTML 4.01 Frameset**
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">`

# Browser Mode

- **Quirks mode** refers to a technique used by some web browsers for the sake of maintaining backward compatibility with web pages designed for old web browsers
- **Standards mode** strictly complying with **W3C** and **Internet Engineering Task Force (IETF)** standards

# Anatomy of HTML Elements

HTML elements are written with a **start** tag, with an **end** tag, with the **content** in between:

- `<tagname>content</tagname>`
- Eg. `<h1>Manipal University</h1>`

## HTML Tag types

- Container Tags come in pairs (paired tags)
  - Eg. `<P>This is in Center</P>`
- Empty Tags do not
  - Eg. `<BR/>`, `<HR/>` (Break and Horizontal line)

# HTML Element attributes

- Attributes provide additional information about HTML elements
- ATTRIBUTE="value", the value should really be in quotes.
- For example, <p style="color:green;">
- Tags can also have default attributes.
- Some browsers don't support the some tags and some attributes.



# Basic Document Structure

- The bare minimum HTML document
- HTML version declaration
- `<html> – <meta> – <head> – <title> – <body>`
- Sequential (top to bottom) rendering

# Semantic html element

- Element that implies some meaning to the content
- Human and/or machine can understand meaning of content surrounded by a semantic element better
- May help search engine ranking, i.e., SEO
- Semantic elements allow for a more meaningful expression of the structure of our HTML page.

# Content Models

A description of the element's expected contents

## Block-Level Elements

- Render to begin on a new line (by default)
- May contain inline or other block-level elements
- Roughly Flow Content (HTML5 category)

## Inline Elements

- Render on the same line (by default)
- May only contain other inline elements
- Roughly Phrasing Content (HTML5 category)

**W3C content models:** <https://www.w3.org/TR/2011/WD-html5-20110525/content-models.html>

# HTML validator

- W3C Markup Validation Service: <https://validator.w3.org/>
- HTML/CSS component support: <https://caniuse.com/>

# HTML Character Entity References

Result	Description	Entity Name
	non-breaking space	&nbsp;
<	Less than	&lt;
>	Greater than	&gt;
&	Ampersand	&amp;
“	Quotation mark	&quot;
©	Copyright	&copy;
¼	One Fourth	&#188
½	Half	&#189

# HTML Character Entity References

- Help avoid rendering issues
- Safeguard against more limited character encoding
- Provide characters not available on a keyboard

# List elements

- Lists provide a natural and commonly used grouping of content
- Very often, lists are used for structuring navigation portions of the web page
- Lists are segregated into three types, namely :
  - **Ordered lists** : lists are numbered according to the standard chosen
  - **Unordered lists** : lists are arranged in bulleted sequence
  - **Definition lists** : list consists of a term followed by its definition

# Unordered Lists

- They are delimited by the <UL> and </UL> tags. Each item in the list is delimited by the <LI> and </LI> tags.
- The syntax of the <UL> tag is:  
    <UL TYPE=" " > <LI> </LI> </UL>

## Attributes

- TYPE= "DISC", "SQUARE" or "CIRCLE".  
    disc (solid bullet), square (solid block) or a circle (hollow bullet)
- The default appearance for list is disc
- </LI> is optional



# Ordered Lists

- They are used when the items in the list have a natural order.
- To make an ordered list, simply change the <UL> tag to <OL>.

## Attributes

TYPE= (1 | A | a | I | i) – changes the style of the list

1 – Arabic numbers

A – Uppercase alphanumeric

a – lowercase alphanumeric

I - Uppercase roman numerals

i – lowercase roman numerals

- Start = value – indicates where the list numbering should begin

# Creating Links

- A link moves us from the current page to a destination that is specified in the HTML page or resource
- The Anchor tag (`<a></a>`) is used to create links between different objects like HTML pages, files, web sites etc.

## Attribute

- HREF -- defines the destination of the link.
  - TARGET -- is used to specify where to open the linked document
  - NAME -- attribute is used to create a bookmark inside an HTML document
- Internal linking to other pages in the site
- External linking to other web sites
- Linking to sections of a document

# Images

- Images can enhance your site
- Remember to specify width and height attributes whenever possible
- Specify the alternate attribute

# Path Convention

## *Absolute Vs. Relative Pathnames*

- An absolute pathname includes the full pathname of the file.
- This means that if you move your files or change your directory, you have to edit every IMG tag in your document.

- *Example*

*<IMG SRC="D:\htmleg\Images\picture.jpg">*

*<IMG SRC="http://xyz.com/Images/picture.jpg">*

# Path Convention

- A relative pathname indicates the pathname of the image file relative to the pathname of the file. This is the recommended naming convention.
- *Example : <IMG SRC="picture.jpg">*
- *Example : <IMG SRC="images/picture.jpg">*
- *If your image file is stored one directory level up from your HTML file, use two dots(..) in the pathname to move up a directory level.  
Example : <IMG SRC="../picture.jpg">*

# Table tags

Tag	Description
<a href="#"><code>&lt;table&gt;</code></a>	Defines a table
<a href="#"><code>&lt;th&gt;</code></a>	Defines a header cell in a table
<a href="#"><code>&lt;tr&gt;</code></a>	Defines a row in a table
<a href="#"><code>&lt;td&gt;</code></a>	Defines a cell in a table
<a href="#"><code>&lt;caption&gt;</code></a>	Defines a table caption
<a href="#"><code>&lt;colgroup&gt;</code></a>	Specifies a group of one or more columns in a table for formatting
<a href="#"><code>&lt;col&gt;</code></a>	Specifies column properties for each column within a <code>&lt;colgroup&gt;</code> element
<a href="#"><code>&lt;thead&gt;</code></a>	Groups the header content in a table
<a href="#"><code>&lt;tbody&gt;</code></a>	Groups the body content in a table
<a href="#"><code>&lt;tfoot&gt;</code></a>	Groups the footer content in a table

# HTML Form

- Web form is container with labels and the form fields
- Example forms used in websites and applications
  - Login/registration
  - Purchase order
  - Ticket reservation
  - Hotel Room booking
  - Payment checkout
  - Newsletter/subscription
  - Consultation/audit request
  - Donation
  - Survey
  - Custom form.

# HTML Form design

- Usability issues
  - Group Similar categories
  - Easy access to the form elements
- Design issues
  - Text emphases
    - Use headings
    - Input specifiers
    - Messages
      - Tooltips
      - Help messages
      - Validation messages
  - Visual separate groups
  - Use line breaks and spacing appropriately
- The information collected from the form can then be used for processing



# HTML Form Designing best practices

- **Single-column layout**

- Read from top to bottom which creates a simple interaction with a form
- Avoid zigzag field placements

- **Path to completion and align**

- Left-align is better than center-align as the best way to reduce the path to completion time
- Left-align eyes don't need to jump across the page

# HTML Form Designing best practices..

- **Use wizard for long forms and progress bar**
- **Group related information**
- **Mention optional and required field/s**
- **Use autocomplete**
- **Choose the input format wisely**
- **Call to action**
- **Use inline form field validation**
- **Use styles effectively**

# Form tag

- The key attributes within the form tag are
  - ACTION – indicates the program on the HTTP server that will process the output from the form
  - METHOD – tells the browser how to send the data to the server with the POST and the GET method.
- Ex:
- `<FORM METHOD = "POST" ACTION = "http://www.XYZ.reg.php">`  
Input elements.....  
`</FORM>`

# Form methods

- GET: Form data is sent in URL of GET request
- POST: Form data is sent in HTTP message body of POST request

## NOTE:

- GET request can be cached
  - GET request remains in browser history and or server log
  - GET request can be bookmarked
- GET request cannot be used to send sensitive data
- GET request have length restriction
- application/x-www-form-urlencoded encoding type is used in GET

# Form methods..

- POST request cannot be cached
  - POST parameters are not stored in browser history and or server log
  - POST request cannot be bookmarked
- POST request can be used to send sensitive data
- POST request has no length restriction
- application/x-www-form-urlencoded or multipart/form-data encoding type is used in GET

# Label

```
<label for="male">Male</label>
```

```
<input type="radio" name="gender" id="male" value="male"><br>
```

```
<label for="female">Female</label>
```

```
<input type="radio" name="gender" id="female" value="female"><br>
```

```
<label for="other">Other</label>
```

```
<input type="radio" name="gender" id="other" value="other">
```

- It provides a usability improvement for mouse users
- If the user clicks on the text within the <label> element, it toggles the control.

# INPUT tag

- The input element is used to select user information.
- An input element can vary in many ways, depending on the type attribute.
- An input element can be of type text field, checkbox, password, radio button, submit button, and more.
- The attributes for the INPUT tag are
  - **type** = (text | password | checkbox | radio | file | hidden | button | image | submit | reset)
  - **name**: is used to specify a name
  - **value**: “text” is used along with radio buttons and checkboxes
  - **size**: “n” – indicates visible size.
  - **maxlength** : “n” in characters
  - **checked**: specifies element should be pre-selected when loaded

# Examples

Example for text box, radio button & check box

<FORM>

First Name : <input type="text" name="fname" size="20" maxlength="30"/>

<input type="radio" name="gender" value="male" checked /> Male<br >  
<input type="radio" name="gender" value="female" /> Female

<input type="checkbox" name="vehicle" value="Bike" /> I have a bike<br />  
<input type="checkbox" name="vehicle" value="Car" /> I have a car

</FORM>



# Password & Hidden fields

- Password fields work like text fields but display “\*” in browser
- `<input type = “password” name=“newpass” size=“10” maxlength=“10” />`
- Hidden fields are not visible in the browser but can be used to pass information to programs receiving input.
- `<input type=“hidden” name = “success” value = “http://www.xmission.com/...”`
  - Better to place them on the top after FORM tag

# To insert a button

- `<input type="submit" />` defines a submit button.
  - A submit button is used to send form data to a server.
  - The data is sent to the page specified in the form's action attribute.
  - The file defined in the action attribute usually does something with the received input:
- `<form name="input" action="html_form_action.asp" method="get">`  
    Username: `<input type="text" name="user" />`  
    `<input type="submit" value="Submit" />`  
    `</form>`
- To define an ordinary button
  - `<input type="button" value="Hello world!">`
- To define an image instead of submit button
  - `<input type="image" src="submit.gif" alt="Submit" />`

# File Field & Text Area

- File fields allow visitors to upload files like pictures, scanned documents, spreadsheets etc.
  - `<input type="file" name="pic" />`
- Text areas are places within a form for extensive text input. `<textarea>`
- Attributes include
  - name, rows, cols
- Ex:
  - `<textarea name="comments" cols="30" rows="5">` pl type comments if any
  - `</textarea>`

# Select Fields

- `<select>`: Sets an area in a form for a select field that can look like a drop down list or a larger select field
- Attributes include
  - **name** – used for form processing
  - **size** – sets visible size for the select field
  - **multiple** – accepts more than one selection
  - **selected** – indicates default selection
- Elements to include
  - Option – used to specify list values or items
  - Optgroup- is used to group related options in list

# Example for select

`<form>`

`<select name="referral" multiple>`

`<option value ="print" selected>print ads </option>`

`<option value ="visit">instore visits </option>`

`<option value ="rec">recommendation </option>`

`</select>`

`<input type="submit" value="submit">`

`<input type="reset" value="reset">`

`</form>`

# Example on optgroup

```
<select name="c">  
  <optgroup label="8.01 Physics I: Classical Mechanics">  
    <option value="8.01.1">Lecture 01: Powers of Ten  
    <option value="8.01.2">Lecture 02: 1D Kinematics  
    <option value="8.01.3">Lecture 03: Vectors  
  <optgroup label="8.02 Electricity and Magnetism">  
    <option value="8.02.1">Lecture 01: What holds our world together?  
    <option value="8.02.2">Lecture 02: Electric Field  
    <option value="8.02.3">Lecture 03: Electric Flux  
  <optgroup label="8.03 Physics III: Vibrations and Waves">  
    <option value="8.03.1">Lecture 01: Periodic Phenomenon  
    <option value="8.03.2">Lecture 02: Beats  
    <option value="8.03.3">Lecture 03: Forced Oscillations with Damping  
</select>
```

# Example for field set

```
<html>
<body>
<form>
  <fieldset>
    <legend>Health information:</legend>
    Height <input type="text" size="3">
    Weight <input type="text" size="3">
  </fieldset>
</form>
</body>
</html>
```

# INPUT element new types

- color
- date
- datetime-local
- time
- email
- url
- month
- week
- number
- range



# INPUT element new attributes

- autocomplete
- autofocus
- list
- min
- max
- disabled
- readonly
- multiple
- pattern
- placeholder
- required
- step

# Example on Datalist

```
<input list="items">  
  <datalist id="items">  
    <option value="Item1">  
    <option value="Item2">  
    <option value="Item3">  
    <option value="Item4">  
    <option value="Item5">  
  </datalist>  
</form>
```

# Summary

- Understood semantics of the element and its attribute set
- Compared the block-level and inline-level content elements
- Discussed the importance of UX best practices when designing the form
- Advantages of using relative path conventions