HTML

INTERNET ENGINEERING

Fall 2022

@1995parham

- Introduction
- HTML
- Body
- Semantic Tags
- Head
- HTML in Practice

- Introduction
- HTML
- Body
- Semantic Tags
- Head
- HTML in Practice

INTRODUCTION

- Remark: The idea of WWW is document sharing
- Main question: How to define the structure of document?
  - Text, tables, figures, link, ...
- In 1980s
  - Binary formats? Useless
    - Different machines, no popular graphical desktops, no such popular format such as PDF, Doc, ...

- Text format
  - It is okay, everyone knows ASCII
    - But how to describe structure, layout, and formatting?
  - Add meaning to each part of text using special predefined markup, E.g., It is heading, It is paragraph, It is table ...

INTRODUCTION (CONT.)

- HTML (Hyper Text Markup Language)
  - A language to define structure of web docs
    - Tags specify the structure
- HTML
  - Was defined with SGML (Standard Generalized Markup Language)
  - Is not a programming language
    - **Cannot** be used to describe computations
  - HTML does/should not specify presentation
    - Font family, style, color, ...
    - Cascading Style Sheet (CSS) is responsible for presentation

INTRODUCTION (CONT.)

- HTML 1 (Berners-Lee, 1989): very basic, limited integration of multimedia
- 1993, Mosaic added many new features (e.g., integrated images)
- HTML 2.0 (IETF, 1994): tried to standardize these & other features
- 1994-96, Netscape & IE added many new, divergent features

- HTML 3.2 (W3C, 1996): attempted to unify into a single standard
- HTML 4.0 (W3C, 1997): attempted to map out future direction
- XHTML 1.0 (W3C, 2000): modified to conform to XML standards
- HTML 5 (Web Hypertext Application Technology Working Group, W3C): New version of HTML4, XHTML 1.0

HTML BASICS: TAGS

- XHTML is a text document collecting elements
- Element: (usually) a tag pair (opening & closing) + content between them
  - E.g., <h1>This is header</h1>
  - Not all tags have content
- Tags specify markups for the content
- Tags
  - <tagname>: opening (start) tag
  - </tagname>: closing (end) tag
  - <tagname />: self-closing tag

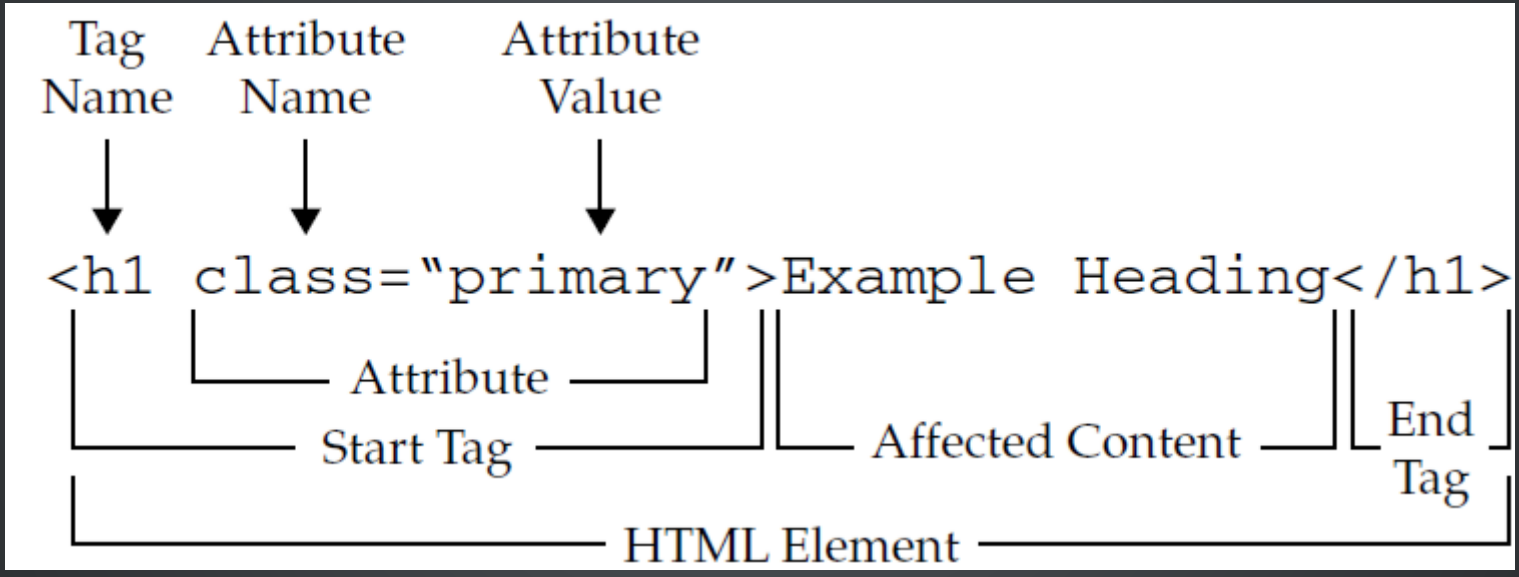
HTML BASICS: ATTRIBUTES

- Each tag can have some attributes
  - Attributes customize tags

```
<tagname attrib1=“setting” attrib2=“setting” ...>
...
</tagname>
```

- Core attributes can be used for most of elements
  - `id`: A unique identifier to element starting with "A-Z"
  - `class`: Assign a class to the element, multiple classes are allowed
  - `title`: Specifies extra information about an element
    - The information is most often shown as a tooltip text when the mouse moves over the element.

HTML BASICS: TAG & ATTRIBUTE & ELEMENT



HTML PROCESSING

- HTML is just a text file; How does it work?
- It is processed by applications for a specific purpose!
- Search engine objectives:
  - Analyze page, extract elements, prioritize, ranking, ...
  - Each tag has meaning, used for ranking
    - E.g., paragraphs are not as important as headings
- Web browser objectives:
  - Display the document to client
  - Rendering
    - Generate layout for the document
    - Display elements

HTML PROCESSING: RENDERING

- The processing of displaying HTML in browser
- Not all tags are to be displayed
  - E.g. Tags in <head>

- For tags which should be displayed
  - Tags by themselves are not displayed
  - Each tag has its own default presentation

- If tag has content, the presentation is applied to content

```
<em>I am important</em>
```

I am important

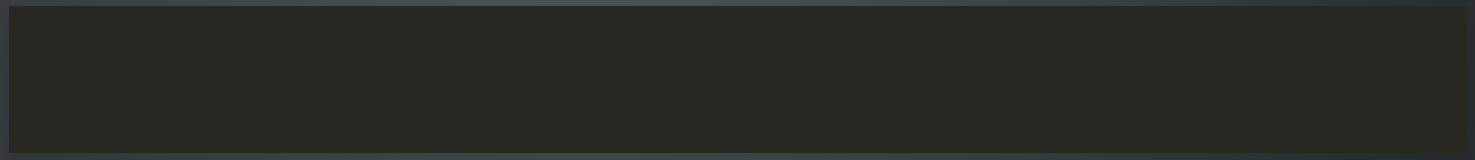
- If tag has not content, the presentation is displayed (if it is needed)

```
I am a <br /> break
```

I am a
break

HTML PROCESSING: RENDERING (CONT.)

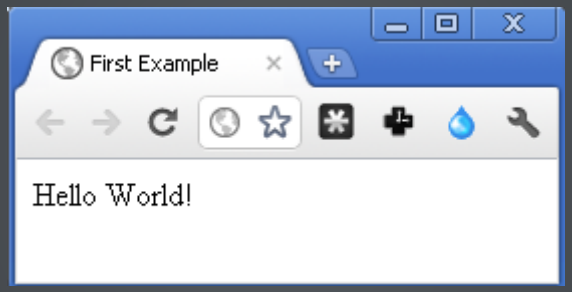
- Web browsers by default start placing elements from left-top corner
  - In-line elements are placed from left to right
  - A new line is created for each block-level element
- Web browsers ignore
  - Comments



  - Tags that don’t recognize
  - More than single whitespaces
    - E.g., Multiple newlines + tabs + spaces → single space

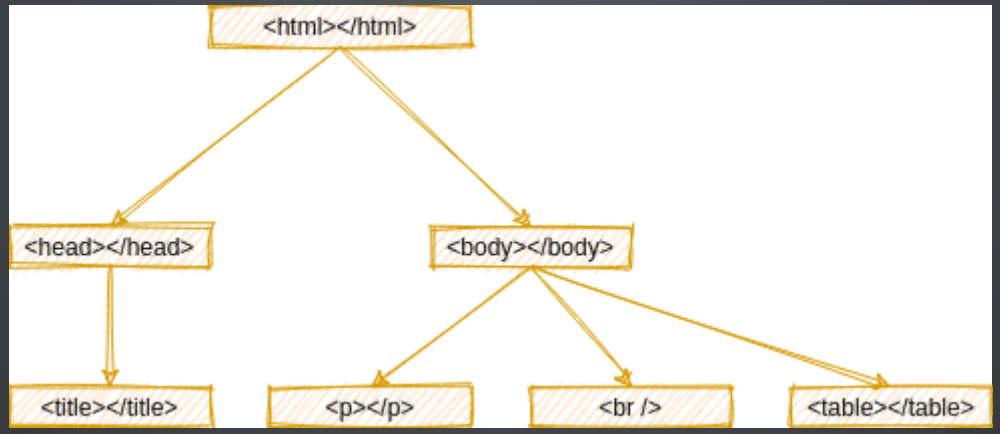
THE “HELLO WORLD” EXAMPLE

```
<html>
  <head>
    <title>First Example</title>
  </head>
  <body>
    <p> Hello World! </p>
  </body>
</html>
```


First Example
Hello World!

NESTED TAGS

- Nested Tags
  - Tree of elements
  - Parent & Child relationship



SPECIAL CHARACTERS/SYMBOLS

- Some characters and symbols are encoded
  - Because cannot be used directly in text files

Character	Coding	Number code
<	&lt;	&#60;
>	&gt;	&#62;
&	&amp;	&
' '	[1]	
©	&copy;	&#169;
λ	&lambda;	&#955;

[1] non breaking space

- Introduction
- HTML
- Body
- Semantic Tags
- Head
- HTML in Practice

XHTML

- HTML is an application of Standard General Markup Language (SGML)
- XHTML is an application of Extensible Markup Language (XML)
- XML is more restricted than SGML
    - XHTML has more restrictions vs. HTML
    - XHTML is more well-defined

XHTML RULES (VS. HTML)

- All tags have ending (closing) tags
  - Some tags are self-closing

```
<br />
```

- Tags cannot be overlapped

```
<strong><em>test</strong></em>
```

  - Who is parent? Who is child?
- All tags are lowercase
- Attributes' value must be in double quotation
- Browsers ignore unknown tags and attributes

- Layout (styles) are separated from markup
  - Markup is used for meaning & structure
  - The obsolete HTML Big Element (`<big>`) renders the enclosed text at a font size one level larger than the surrounding text.
  - The obsolete HTML Center Element (`<center>`) is a block-level element that displays its block-level or inline contents centered horizontally within its containing element.

ANATOMY OF AN HTML DOCUMENT

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <img src="images/firefox-icon.png" alt="My test image">
  </body>
</html>
```

ANATOMY OF AN HTML DOCUMENT (CONT.)

`<!DOCTYPE html>` THE DOCTYPE

- It is required preamble.
- In the mists of time, when HTML was young (around 1991/92), doctypes were meant to act as links to a set of rules that the HTML page had to follow to be considered good HTML, which could mean automatic error checking and other useful things.
- However these days, they don't do much, and are basically just needed to make sure your document behaves correctly.
- That's all you need to know for now.

`<html></html>`

- This element wraps all the content on the entire page and is sometimes known as the root element.

<head></head>

- This element acts as a container for all the stuff you want to include on the HTML page that isn't the content you are showing to your page's viewers.
  - keywords and a page description that you want to appear in search results
  - CSS to style our content
  - character set declarations
  - and more

`<meta></meta>`

- This element sets the character set your document should use to UTF-8.

`<title></title>`

- This sets the title of your page, which is the title that appears in the browser tab the page is loaded in.

<body></body>

- This contains all the content that you want to show to web users when they visit your page.

- Introduction
- HTML
- Body
- Semantic Tags
- Head
- HTML in Practice

HTML ELEMENTS

- Main root
- Document metadata
- Sectioning root
- Content sectioning
- Text content
- Inline text semantics
- Image and multimedia
- Embedded content
- Scripting

- Table content
- Forms
- Interactive elements
- Web Components

MAIN ROOT

- The HTML `<html>` element represents the root (top-level element) of an HTML document, so it is also referred to as the root element.
- All other elements must be descendants of this element.

SECTIONING ROOT

- The HTML <body> Element represents the content of an HTML document.
- There can be only one <body> element in a document.

CONTENT SECTIONING

- Content sectioning elements allow you to organize the document content into logical pieces.

- The HTML `<h1>`–`<h6>` elements represent six levels of section headings. `<h1>` is the highest section level and `<h6>` is the lowest.
- The HTML `<section>` element represents a standalone section — which doesn't have a more specific semantic element to represent it — contained within an HTML document.
- Content you are sectioning must have a natural heading within it (for example `h1-h6`).

- The HTML `<footer>` element represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.
- The HTML `<header>` element represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also a logo, a search form, an author name, and other elements.

- The HTML `<nav>` element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.

TEXT CONTENT

- Use HTML text content elements to organize blocks or sections of content placed between the opening `<body>` and closing `</body>` tags.
- The HTML `<p>` element represents a paragraph.
- The HTML `<pre>` element represents preformatted text which is to be presented exactly as written in the HTML file.

- The HTML `<ol>` element represents an ordered list of items — typically rendered as a numbered list.
    - Tho ordered list uses a specific attribute to define a custom start number. The attribute syntax is start=”X”, where the X value is the list custom starting number.
- The HTML `<ul>` element represents an unordered list of items, typically rendered as a bulleted list.
    - The HTML `<li>` element is used to represent an item in a list.
- The HTML `<dl>` element represents a description list.
    - The element encloses a list of groups of terms (specified using the `<dt>` element) and descriptions (provided by `<dd>` elements).

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
</ol>
<dl>
  <dt>Hello</dt>
  <dd>Means salam</dd>
</dl>
```

- Item 1
- Item 2

1. Item 1
2. Item 2

Hello

Means salam

- The HTML Content Division element (`<div>`) is the generic container for flow content.

INLINE TEXT SEMANTICS

- Use the HTML inline text semantic to define the *meaning*, **structure**, or style of a word, line, or any arbitrary piece of text.
- The HTML `<span>` element is a generic inline container for phrasing content, which does not inherently represent anything.
- The HTML `<q>` element indicates that the enclosed text is a short inline quotation. Most modern browsers implement this by surrounding the text in quotation marks.
- The HTML Citation element (`<cite>`) is used to describe a reference to a cited creative work, and must include the title of that work.

- The HTML <a> element (or anchor element), with its `href` attribute, creates a hyperlink to web pages, files, email addresses (mailto), locations in the same page (fragment), or anything else a URL can address.
    - When scheme is not give in the URL & base is not set in <head>, it is assumed as a file in current domain
        - href="http://www.google.com" open Google
        - href="www.google.com" open a file in current directory named www.google.com
        - href= "/www.google.com" open a file in the root directory named www.google.com

- For paths in *current domain*, similar to filesystem, paths can be
  - Absolute: Path starts from web server root directory
  - Relative: Path starts from current directory
- Scheme can be every supported protocol
  - E.g. `mailto` for sending email
  - E.g. `javascript` to run code
- By default links are opened in the same window, to open link in new window
  - Attribute target="_blank"
- Everything between <a> is considered as link name
- Avoid spaces after <a> and before

- #frag part in URL is used to jump middle of a large document
- Step one: assign an *id* to the part

```
<a id="SctionResult">Results</a>
<h2 id="SctionResult">Results</h2>
```

- The *id* is case sensitive
- Step two: create link using #frag feature

```
To see result <a href="xyz#SctionResult">click here</a>
```

- The HTML `<br>` element produces a line break in text (carriage-return).
- The HTML `<code>` element displays its contents styled in a fashion intended to indicate that the text is a short fragment of computer code.
- The HTML `<em>` element marks text that has stress emphasis.
- The HTML Strong Importance Element (`<strong>`) indicates that its contents have strong importance, seriousness, or urgency.

IMAGE AND MULTIMEDIA

- The HTML `<img>` element embeds an image into the document.
  - `src`: address of file (local or remote)
  - `alt`: alternative message shown if image cannot be displayed
  - align: alignment of image with respect to text line (deprecated, is controller by CSS)
  - There is no caption for images!!!Images are inline elements

EMBEDDED CONTENT

- The HTML Inline Frame element (`<iframe>`) represents a nested browsing context, embedding another HTML page into the current one.

SCRIPTING

- The HTML `<noscript>` element defines a section of HTML to be inserted if a script type on the page is unsupported or if scripting is currently turned off in the browser.
- The HTML `<script>` element is used to embed executable code or data; this is typically used to embed or refer to JavaScript code.

TABLE CONTENT

- Tables are created by `<table> </table>`
- Each row is created by `<tr> </tr>`
- Each column inside a row is created by `<td> </td>`
- Heading of a column is by `<th> </th>`
- Block-level element

```
<table>
  <caption>I am a lonely table</caption>
  <thead>
    <tr>
      <th>Head Col 0</th>
      <th>Head Col 1</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Row 0 Col 0</td>
      <td>Row 0 Col 1</td>
    </tr>
    <tr>
      <td>Row 1 Col 0</td>
      <td>Row 1 Col 1</td>
    </tr>
  </tbody>
```

I am a lonely table

Head Col 0	Head Col 1
Row 0 Col 0	Row 0 Col 1
Row 1 Col 0	Row 1 Col 1

CE & IT of AUT

h1-h6

Introduction

p

In this lecture, we want to learn basic concepts and syntax of HTML.

This is another paragraph

HTML provides many facilities to describe *structure* of document including

em

ol

1. Text
    - Paragraph
    - Lists including:

ul

dl

Ordered List
: This is for ordered list

Un-ordered List
: This is for un-ordered list

dt

Definition List
: This is for definition list

dd

2. Images
3. Tables
4. Forms
5. etc. ...

HTML is the basic technology of the web. It is the business of the major cor

Web Empires

table

#	Name	Logo	Sites			Business
1	Google	Google	Google	Gmail	Scholar	Everything
2	Facebook		Facebook			Social
3	Linkedin		Linkedin			

tr

td

img

FORMS

- Forms are used to **get** information from user
- HTML is only responsible to gather the information
    - It does not responsible to process
    - Data are processed by server side scripts
        - However, some preprocessing can also be performed in client slide
        - As we are going to see, with Javascript we can even use forms' data without any server side processing
            - drawio

- Major form components
  - The form element
  - Inputs
    - Text, Checkboxes, radio buttons, select boxes, etc.
  - Buttons
    - Submit, cancel, etc.

- Forms are created by `form`
- Each form must have `action` and `method` attributes
  - `action` is a URL
    - Server side script that process the data
  - `method` is a HTTP method used to send data
    - `get`: User input data is sent through the *query part* of URL by HTTP GET method
    - `post`: User input data is sent as the *body* of HTTP message by HTTP POST method

- A from is composed of `input` elements
- Each component has `type`, `name`, and `value` attributes
    - `type` specifies the type of component
    - `name` is the name of the component
    - `value` (except buttons)
        - If not empty, is **the default value**
- On submission, `name`=*value* (user input or default) of the components in the form are sent to server (using the `action` and `method`: POST, GET)
    - Server processes the values according to the names
    - It must know the names

- Single-line text
  - `type="text"`
- Password (instead of real input, other character is shown)
  - `type="password"`
- Multi-line text
  - Instead of `<input>`, we use `<textarea> </textarea>`
  - `cols` & `rows` specifies # of columns & rows
- name=value of component is sent to server
  - Password in plain text format 😱

```
<form action="http://httpbin.org/get" method="get">
  Search:
  <input type="text" name="txtSearch" value="" size="20" maxlength="64" />
  <br />
  Password:
  <input type="password" name="pass" value="" size="20" maxlength="64" />
  <br />
  Text:
  <textarea name="inputtext" cols="30" rows="3">
Please enter your message</textarea>
</form>
```

- `type="checkbox"`
- If checked, its name=value is sent to server
  - User cannot change/enter value
    - The value attribute is needed in some cases
    - If not given, it is assumed “on”
- To be checked by default
  - `checked="checked"`
- To draw border around a group of components
  - `<fieldset> </fieldset>`
- To assign name to the group
  - `<legend> </legend>`

Web Development Skills

- [ ] HTML
- [x] XHTML
- [ ] CSS
- [ ] JavaScript
- [ ] ASP.Net
- [ ] PHP

Submit

```
<form action="http://httpbin.org/get" method="get">
  <fieldset>
    <legend><em>Web Development Skills</em></legend>
    <input type="checkbox" name="skill_1" value="html" />
    HTML<br />
    <input type="checkbox" name="skill_2" value="xhtml" checked="checked" />
    XHTML<br />
    <input type="checkbox" name="skill_3" value="css" /> CSS<br />
    <input type="checkbox" name="skill_4" value="JavaScript" />
    JavaScript<br />
    <input type="checkbox" name="skill_5" value="aspnet" />ASP.Net<br />
    <input type="checkbox" name="skill_6" />
    PHP <br />
    <input type="submit" value="Submit" />
  </fieldset>
</form>
```

- `type="radio"`
- Only one of button can be selected in a group of buttons with the same name
- name=value of the selected button will sent
- Again, user cannot change/enter the value
- If the value attribute is missing, the default value is “on”
- The value attribute is (almost always) needed
- The name is the same for all choices

University Grade

- BS
- MS
- PhD
- Post Doc

Submit

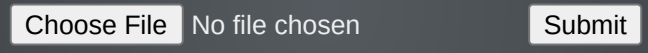
```
<form action="http://httpbin.org/get" method="get">
  <fieldset>
    <legend>University Grade</legend>
    <input type="radio" name="grade" value="B" /> BS <br />
    <input type="radio" name="grade" value="M" /> MS <br />
    <input type="radio" name="grade" value="P" /> PhD <br />
    <input type="radio" name="grade" value="PD" /> Post Doc <br />
    <input type="submit" value="Submit" />
  </fieldset>
</form>
```

- Select Boxes has the same functionality of radio buttons
- Created by:
  - `<select name="selname"></select>`
  - Options are given by `<option value="val"> text </option>`
- slename=val of the selected item is sent to server
- User cannot enter value
- If `value` attribute is omitted, the value is taken from the text content of the `option` element.

Select color: Red Submit

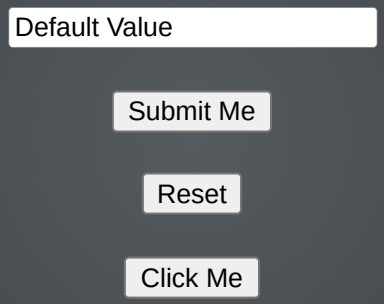
```
<form action="http://httpbin.org/get" method="get">
  Select color:
  <select name="selColor">
    <option value="r">Red</option>
    <option value="g">Green</option>
    <option value="b">Blue</option>
  </select>
  <input type="submit" value="Submit" />
</form>
```

- File Input
- In `<input>`
  - `type`="file"
  - `accept`= A MIME type to specify default acceptable file format
- In `<form>`
  - `method`="post"
  - `enctype`="multipart/form-data"
  - The type that allows file `<input>` element(s) to upload file data.


Choose File
No file chosen
Submit

```
<form
  action="http://httpbin.org/post"
  method="post"
  enctype="multipart/form-data"
>
  <input type="file" name="fileUpload" accept="image/*" />
  <input type="submit" value="Submit" />
</form>
```

- Buttons: <button> label </button>
  - `type` The default behavior of the button. Possible values are:
    - `"submit"`: To submit data to server.
    - `"reset"`: To reset all inputs to default values.
    - `"button"`: To run client side script.
- The `name` of the button, submitted as a pair with the button's value as part of the form data.
- Defines the `value` associated with the button's name when it's submitted with the form data.


Default Value
Submit Me
Reset
Click Me

```
<form action="http://httpbin.org/get" method="get">
  <input type="text" name="input" value="Default Value" /> <br />
  <button type="submit">Submit Me</button><br />
  <button type="reset">Reset</button> <br />
  <button type="button">Click Me</button> <br />
</form>
```

TABINDEX

- The *tabindex* **global attribute** indicates that its element can be focused, and where it participates in sequential keyboard navigation (usually with the Tab key, hence the name).

- Introduction
- HTML
- Body
- Semantic Tags
- Head
- HTML in Practice

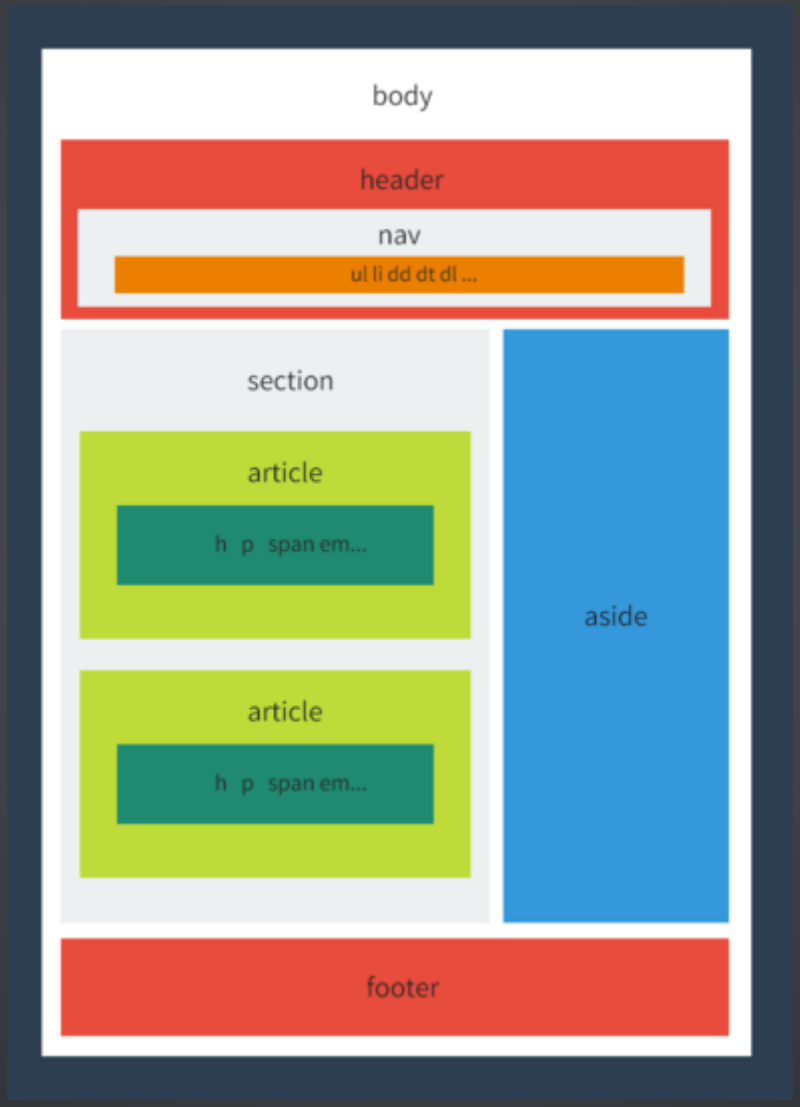
SEMANTIC TAGS

- Many HTML tags have semantic meaning
- Semantic tags convey some information about the type of content inside them.
- Semantic tags make it clear to the browser what the meaning of a page and its content is.
- They also help search engines, ensuring that the right pages are delivered for the right queries.

SOME SEMANTIC TAGS

- head
- body
- header
- nav
- main
- section
- article
- aside
- footer
- code

SEMANTIC TAGS EXAMPLE



- Introduction
- HTML
- Body
- Semantic Tags
- Head
- HTML in Practice

DOCUMENT METADATA

- Metadata contains information about the page.
- This includes information about styles, scripts and data to help software (search engines, browsers, etc.) use and render the page.
- Metadata for styles and scripts may be defined in the page or link to another file that has the information.

- The HTML Title element (`<title>`) defines the document's title that is shown in a browser's title bar or a page's tab.
- The HTML `<style>` element contains style information for a document, or part of a document.
- The HTML `<base>` element specifies the base URL to use for all relative URLs in a document.

- The HTML `<meta>` element represents metadata that cannot be represented by other HTML meta-related elements

- If the name attribute is set, the meta element provides document-level metadata, applying to the whole page.
  - `author`: the name of the document's author.
  - `description`: a short and accurate summary of the content of the page. Several browsers, like Firefox and Opera, use this as the default description of bookmarked pages.
  - `generator`: the identifier of the software that generated the page.
  - `keywords`: words relevant to the page's content separated by commas.

```
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
```

- If the http-equiv attribute is set, the meta element is a pragma directive, providing information equivalent to what can be given by a similarly-named HTTP header.
    - Usually is not processed by web-server
    - Browser simulates the behavior of the effect of the header

```
<!-- The number of seconds until the page should redirect to another
<meta http-equiv="refresh" content="3;url=https://www.mozilla.org">
```

- If the charset attribute is set, the meta element is a charset declaration, giving the character encoding in which the document is encoded.

```
<meta charset="UTF-8">
```

- If the itemprop attribute is set, the meta element provides user-defined metadata.

```
<meta itemprop="description" content="my description">
```

- The HTML External Resource Link element (`<link>`) specifies relationships between the current document and an external resource. This element is most commonly used to link to stylesheets, but is also used to establish site icons (both "favicon" style icons and icons for the home screen and apps on mobile devices) among other things.

- Introduction
- HTML
- Body
- Semantic Tags
- Head
- HTML in Practice

HTML REMARKS

- HTML is open source
  - We can find how others do amazing things in web
  - Learning by reading the others' codes
    - Copy/Paste is strictly prohibited (copyright)
- HTML is not a programming language
  - No compiler or interpreter
    - So, what happen if there is an error?
      - Depends on browser
      - Developer should check with multiple browsers

HTML DEVELOPMENT TOOLBOX

- A HTML editor
  - A simple text editor
    - Neovim with Apache License
    - Vim is Charityware
    - Atom with MIT License
    - VSCode with MIT License
    - ...
  - HTML source code editor (syntax highlight, etc)
    - WebStorm 💰
  - WYSIWYG editors (you have not work with tags) E.g. MS. FrontPage, Word (export to HTML), ...

- A rendering software
  - Common browsers Try different browsers
- Additional debugging tools
- HTML Debugging
  - Browser reads HTML document
    - Parses its tree
    - Document Object Model (DOM) tree
      - Shows how browser interprets your HTML file
  - Google Chrome **Inspect Element**
  - Firefox Developer Edition

REFERENCES 

- Prof. Bahador Bakhshi's Internet Eng. Course's Slides