
CSC309H1S

Programming on the Web

Winter 2023

Lecture 2: Hypertext Markup Language

Instructor: Kuei (Jack) Sun

Department of Computer Science
University of Toronto

Markup Language

- Language that provides control over organization of document content
- Allows specification of its structure and various components
 - E.g., headings, paragraphs, etc.
- Can help with formatting of text or multimedia components
 - E.g., Markdown
- Extensible Markup Language (XML)
 - Provides a standard for storage and transmission of arbitrary data
 - Labels, categorizes, and organizes information
 - Now used commonly for interchange of data over the Internet

Hypertext Markup Language

- A special form of XML for interchange of web documents
- Browser *renders* an HTML file to display a web page
- Emphasis on *structural semantics* of elements
- Elements
 - Building blocks of a web page
 - Wide variety of elements are supported
 - E.g., images, videos, embedded PDFs, interactive objects
 - Declared using a tag

- E.g., image tag
 - ``
 - ↑ tag
 - ↑ attribute name
 - ↙ attribute value

HTML Syntax

- Two types of elements

1. Regular elements

- Element can have nested elements or text
- Most HTML tags are in this category
- Requires a closing tag
- E.g., section tag

```
• <section>  ← opening tag
    <p>first paragraph</p>
    <p>second paragraph</p>
  </section> ← closing tag
```

} section element

2. Void elements

- Cannot have nested elements; does not require closing tag. E.g., `img`

An Example HTML file

```
<!-- My first HTML file -->  ← This is a comment
<!DOCTYPE html>  ← Not HTML. It declares the format of the document
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My first HTML file</title>
</head>
<body>
  <header>Hello World</header>
  <main>
    <h1>Welcome!</h1>
    <p>I am Mishu.</p>
    
  </main>
  <footer>Bye</footer>
</body>
</html>
```

Basic HTML tags

- `<html>`
 - The root element. All other tags must be declared within `<html>`
- `<head>`
 - The “invisible” part of an HTML document
 - Specifies various information about the document
 - E.g., `<title>` goes in here to specify the title of the document
- `<body>`
 - The “visible” part of an HTML document
- `<header>`, `<main>`, `<footer>`
 - Analogous to the header and footer of a printed document
 - `main` is the “meat” of the content

Basic HTML tags

- Visit www.w3schools.com/html/ for a full list
- Headings: <h1> to <h6>
 - Typically used to name a section of the document
 - Headings usually have larger font size
 - By default, h1 is largest, h6 is smallest
- Paragraph: <p>
 - A block of text
- Anchor: <a>
 - Defines a hyperlink
 - `Go to Google`

HTML Attributes

- Identifiers
 - `id`: specifies name of a unique element in the document
 - `class`: specifies name(s) of class(es) in which elements with same class share the same style and/or behavior
 - Used extensively by CSS to style elements
 - Hyperlink can include id to jump to that element
 - Example:
 - https://en.wikipedia.org/wiki/HTML#HTML_5
- Some tags have required attributes
 - E.g., `` requires the `src` attribute to specify the URL of the image
- You can “make your own” attributes, which are ignored by renderer

Whitespaces in HTML

- Whitespaces in HTML files are **ignored**.

- Example:

```
<p>Hello
```

World

```
</p>
```

- when rendered, becomes:

Hello World

- forced line break `
`
 - Seldom used. Why?

Preformatted Text

- Prefomatted text: `<pre>`
 - Asks browser to *not ignore* whitespaces inside the element
 - Often used to show source code
 - Problem: what if the text uses the greater than or less than character?
- HTML entities
 - Represent characters reserved by the HTML language
 - Starts with ampersand, followed by entity name or number
 - These 4 ASCII characters must be escaped
 - Greater than `>` becomes `>` ;
 - Less than `<` becomes `<` ;
 - Ampersand `&` becomes `&` ;
 - double quote `"` becomes `"` ;
 - Other entities: https://www.w3schools.com/html/html_entities.asp

Organizing Elements

- Division `<div>` tag
 - Block level element
 - Changes to a new line wherever defined
 - Used extensively for organizing and styling elements
 - Can be styled to provide spacing and alignment of child elements
- Span `` tag
 - Inline element
 - Allows styling of an inline element or text
 - `<p>Hello World!</p>`
 - Alternatives include `` (emphasis) and ``
 - Do not use: strike through `<strike>`, bold* ``, italic* `<i>`

HTML Table

- `<table>` tag
 - Creates a table of rows and columns
- Each row is specified by the `<tr>` tag
- Each cell is specified by the `<td>` or `<th>` tag
 - `<td>`: table data
 - `<th>`: table header
- Number of cells in row should match expected number of columns
- `colspan` and `rowspan` attributes
 - Allows a cell to span multiple columns or rows
- More rigid than `<div>` and ``

HTML Lists

- Unordered list: ``
 - Each list element is prefixed with a symbol, e.g., bullet
- Ordered list: ``
 - Each list element is prefixed with an ordinal value, e.g., Arabic number
- List item: ``
 - An item inside the list. Must be a child of either `` or ``
- Description list: `<dl>`
 - A list of key value pairs
 - Child elements must alternate between term `<dt>` and description `<dd>`
 - Hint: very good for designing forms

HTML Form

- Primary way to send user data to server
- `<form>` tag
- Input elements

```
<input class="mystyle" type="text" name="first_name" size="60" required>
```

↑
boolean attribute

- Many other types
 - text-based: `password`, `email`
 - file upload: `file`
 - button-like: `radio`, `checkbox`, `submit`
- `<textarea>` tag for multiline input
- When user presses submit, a request can be sent

HTML Form

- `action` attribute defines the URL of the HTTP request
- `method` attribute defines which HTTP method will be used
 - GET: usually used for queries and filters
 - E.g., Google search
 - Data are appended to the end of the URL
 - POST: data sent in body of HTTP request
 - Usually used for secure transfer, e.g., if password is in the form
- Form data consists of key-value pairs of input name and their values
- Example:

```
<form action="/submit" method="POST">  
  ...  
</form>
```

HTML Validation

- Helps you identify issues with your HTML code
- Go to <https://validator.w3.org/>
- Detects many types of issues
 - Syntax error
 - E.g., missing a closing tag
 - Semantic error
 - E.g., missing required elements, such as <title>
 - Warnings
 - Not critical but would be nice to fix
 - tag should always have the alt attribute for accessibility reasons