DBI-011
HyperText Markup Language (HTML)

COMP1048: Databases and Interfaces

Matthew Pike (matthew.pike@nottingham.edu.cn)

Dylan Shen (linlin.shen@nottingham.edu.cn)

Today

- ❓ What makes a web-page?
- 🏭 How do we produce one? How should we organise our code?
- ⚙️ What tools do we need?
- 🖊️ By the end of this lecture, you should be able to write a simple HTML page (and understand how it works!)
- 🎉 For fun, I decided to implement these slides in HTML also!

Client -- Server Paradigm

- Web browsers communicate with web servers using HTTP
- Web servers wait for client request messages, process them when they arrive, and reply to the web browser with an HTTP Response message
    - You may be familiar with HTTP Response status codes:
        - 200 - OK
        - 404 - Not Found
        - 503 - Service Unavailable (Moodle's Favourite)

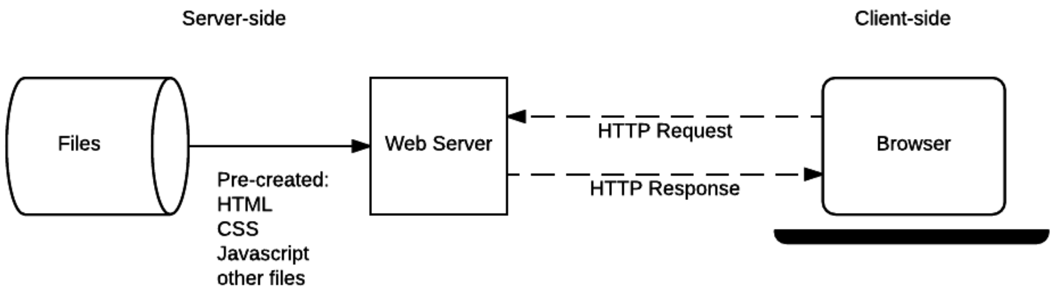


Image Credit - https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview

What makes a webpage?

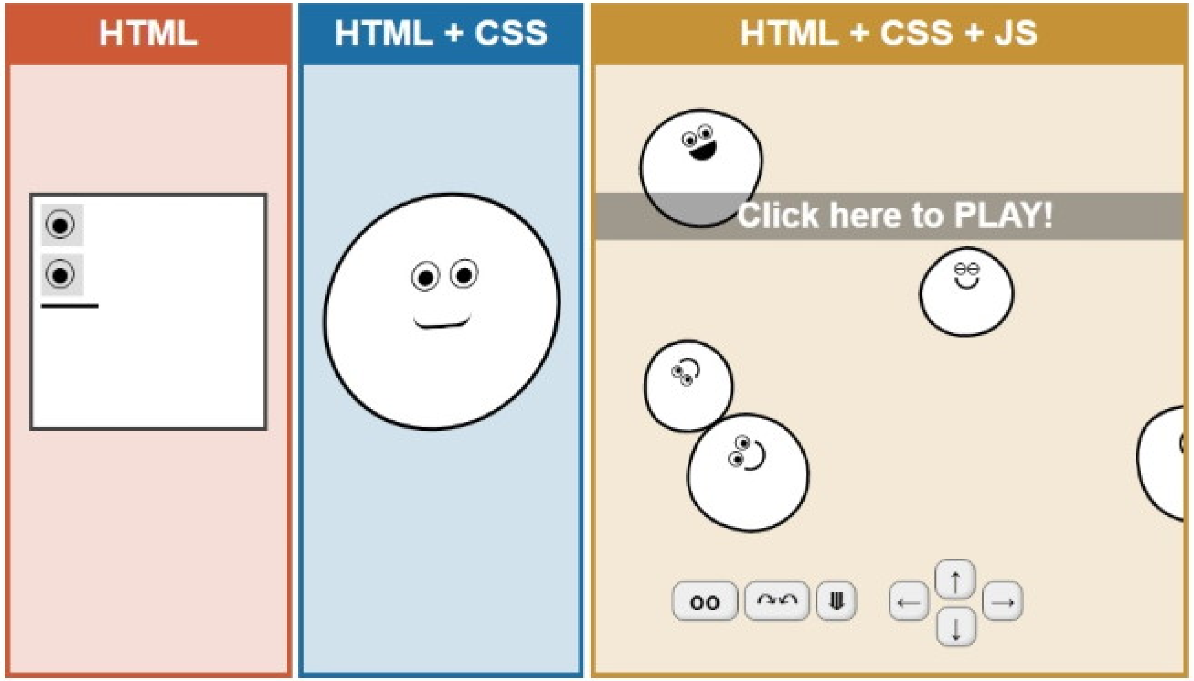
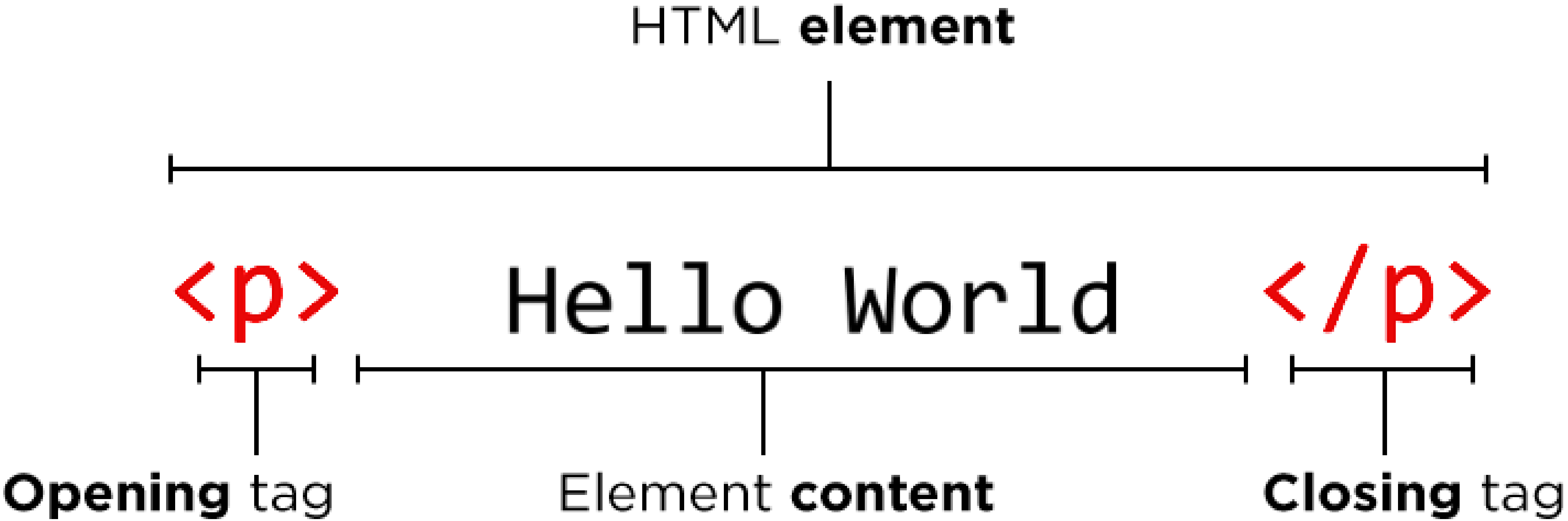


Image Source: https://html-css-js.com/images/og.jpg

What is HTML?

- HTML is a *markup language* that defines the **structure** of your content.
- HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.
- The HTML Language Specification is maintained by the W3C Group
    - HTML 5.2 Specification - https://www.w3.org/TR/html52/
- **Note**: There are many different versions of HTML. This module will only consider the latest (and greatest!) - HTML5.



HTML **element**

`<p>` Hello World `</p>`

Opening tag

Element **content**

Closing tag

Hello World

```html
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Hello COMP1048</title>
</head>
<body>
        Hello World!
</body>
</html>
```

Source available on Moodle - examples/DBI_011.Example_001.HelloWorld.html

HTML Hierarchy

- `<!DOCTYPE html>` — The doctype. It is a required preamble. Indicates which HTML version is being used.
- `<html></html>` — the root element. Wraps around all content. The document's language can also be specified using the `lang` attribute.
- `<head></head>` — Includes metadata, CSS and character set declarations (among many other things). Not visible to users.
- `<meta charset="utf-8">` — Tells the browser which character set your webpage is using. UTF-8 should be fine for most of your webpages.
- `<title></title>` — Sets the title of your page, which appears in the browser tab.
- `<body></body>` — This is where your content lives. This will be visible to users.

Nested HTML

- **Nesting** - Putting elements inside other elements.
- Ensure that your elements are properly nested:
  - `<p> Don't do <i> this </p> </i>`
- You may find that this code works ... but it is invalid - Test your code here

```html
<div>
<p>
  Hello <strong> World! </strong>
</p>
<p>
  Welcome to HTML.
</p>
</div>
```

Images

- All modern web browsers can display images e.g. PNG, JPEG, GIF
- The `src` attribute is required - the path to the image
- The `alt` attribute holds a text description of the image
    - Accessibility: screen readers read this description out to their users
    - Alt text is also displayed on the page if the image can't be loaded
- We aren't required to close image tags, but we can - `<img src="..." alt="...: />`

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Images</title>
</head>
<body>
        <!-- Image from local filesystem -->
        <img src="images/trees.jpg" width="500"
                 alt='Colourful forest trees.'>

        <!-- Image from URL -->
        <img src="https://www.sqlite.org/images/sqlite370_banner.gif"
                 alt='The SQLite Database Logo'>

</body>
</html>
```

Source available on Moodle - examples/DBI_011.Example_002.Images.html

Headings, Text and Lists

- Headings may range from levels 1-6 e.g., `<h1>` `<h2>` ... `<h6>`
- Text can be structured into paragraphs using the `<p>` element
- Lists may be:
  - **Ordered** - `<ol>` - order matters (e.g. 1, 2, 3)
  - **Un-Ordered** - `<ul>` - order does not matter (e.g. Bullet Points)
- List items are specified using `<li>` regardless of ordering.

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Headings, Text and Lists</title>
</head>
<body>
        <h1> Joe Bloggs </h1>
        <p> Welcome to my personal website.</p>

        <h2> About me</h2>
        <p>Hello, my name is Joe Bloggs.</p>
        <p>My hobbies include:</p>
        <ul>
                <li>Reading</li>
                <li>Writing</li>
                <li>Playing Football</li>
        </ul>

        <h2> My Favourite Things </h2>
        <h3> Books </h3>
        <p>My Favourite Books:</p>
        <ol>
                <li>Book A</li>
                <li>Book B</li>
                <li>Book C</li>
        </ol>
        <h3> Football Teams </h3>
        <ol>
                <li>Swansea City</li>
                <li>Swansea City</li>
                <li>Swansea City</li>
        </ol>
</body>
</html>
```

Source available on Moodle - examples/DBI_011.Example_003.HeadingsTextLists.html

Hyperlinks

- The `<a>` anchor element creates a hyperlink
- `href` attribute specifies the destination, which could be:
    - web-pages
    - files
    - email addresses
    - etc

```html
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Hyperlinks</title>
</head>
<body>
<a href="https://www.nottingham.edu.cn/">

<img
        src="https://www.nottingham.edu.cn/siteelements/images/core/logo-english-dark.png"
        alt="University of Nottingham logo"
>

</a>
<p>

        <a href="DBI_011.Example_002.Images.html"> Trees </a>

</p>
</body>
</html>
```

Source available on Moodle - examples/DBI_011.Example_004.Hyperlinks.html

Tables

- A table is a structured set of data made up of rows and columns
- A table is split into heading (`<thead>`), body (`<tbody>`) and footer (`<tfoot>`)
- `<th>` indicates a table header element
- Rows are defined using `<tr>` , with individual cells defined by `<td>`
- `<caption>` helps people navigating with the aid of assistive technology such as a screen reader

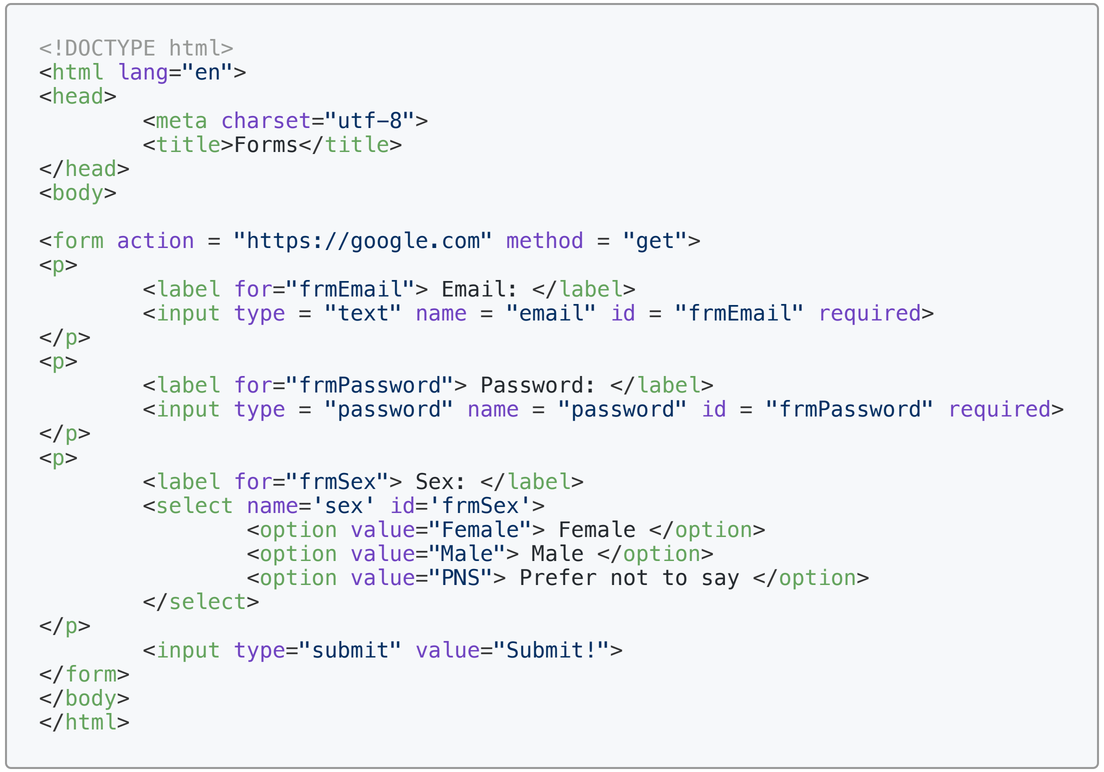
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Tables</title>
    <!-- We'll cover this next lecture -->
    <link href ="style.css" rel="stylesheet">
</head>

<body>
<table>
    <caption> DBI Class Schedule </caption>
    <thead>
        <tr>
            <th>Week</th>
            <th>Topic</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>Introduction to DB</td>
        </tr>
        <tr>
            <td colspan="2">
                    <b> <u> Break!</u> </b>
            </td>
        </tr>
    </tbody>
</table>
</body>
</html>
```

Source available on Moodle - examples/DBI_011.Example_005.Tables.html

Forms

- The `<form>` element is an interactive controls for submitting information
- Clicking the Submit button will send the form’s values to the server
- The only required attribute of `<form>` is `action` which specifies the URL to be called when the Submit button is clicked
- The `name` attribute is the name of the associated data point submitted to the server



```html
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8">
        <title>Forms</title>
</head>
<body>

<form action = "https://google.com" method = "get">
<p>
        <label for="frmEmail"> Email: </label>
        <input type = "text" name = "email" id = "frmEmail" required>
</p>
<p>
        <label for="frmPassword"> Password: </label>
        <input type = "password" name = "password" id = "frmPassword" required>
</p>
<p>
        <label for="frmSex"> Sex: </label>
        <select name='sex'  id='frmSex'>
                <option value="Female"> Female </option>
                <option value="Male"> Male </option>
                <option value="PNS"> Prefer not to say </option>
        </select>
</p>
        <input type="submit" value="Submit!">
</form>
</body>
</html>
```

Source available on Moodle - examples/DBI_011.Example_006.Forms.html

GET vs POST

- `POST` method sends data to the server
  - The data sent to the server with POST is stored in the request body of the HTTP request
  - The request is indicated by the `Content-Type` header
- `GET` method requests a representation of the specified resource.
  - Requests using `GET` should only be used to request data
  - they shouldn't include data

    > Does our previous example follow this guidance?

Practical Hints and Tips

- Ensure that you use a 'English' language keyboard input when developing your HTML.
  - Web-browsers can be sensitive to special characters
- Use a *good* web-browser:
  - Google Chrome
  - Mozilla Firefox
- Use a code editor
  - Sublime Text
  - VS Code
- Start simple. Don't try to run before you can walk.

Resources & Further Reading

- Mozilla Developer Network (MDN) - An Excellent Resource
  - The Basics of HTML
- HTML 5 Specification
- The examples presented here are available on Moodle