Web Programming
EECS 348 Lab 4 — 2/20/2025

Harlan Williams

Electrical Engineering and Computer Science
University of Kansas

Connecting to the lab computers with SSH

- Linux or macOS use the terminal and `ssh`, Windows users can use PuTTY (https://putty.org/)
- Connect using your KU id as your username and the cycle server, *e.g.*, `ssh a123z789@cycle1.eecs.ku.edu`
  - Can replace `cycle1` with `cycle2`, `cycle3` or `cycle4`
- You will be prompted for a password, which is the same password you log into Canvas with. With SSH, your input won't be echoed but that doesn't mean the server is frozen; it's just hiding your password for privacy
- For your files to be displayed on your EECS people page, they must be in the `public_html` directory. By default the server looks for an HTML file named `index.html` to display

HTML (HyperText Markup Language)

- Plain-text file format for conveying both the **content** and **structure** of a web page
- Content (what is to be displayed) is stored in **elements**
- Structure is conveyed by nesting elements within a tree
- Designed to be easy to modify and generate programmatically

```html
<html lang="en-US">
  <head>
    <title>My Website</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

https://developer.mozilla.org/en-US/docs/Glossary/HTML

HTML syntax

An **element** is the basic unit of an HTML document. An element begins and ends with **tags** defining the type of the element, followed by optional key-value **attributes**, and then the **content** of that element.
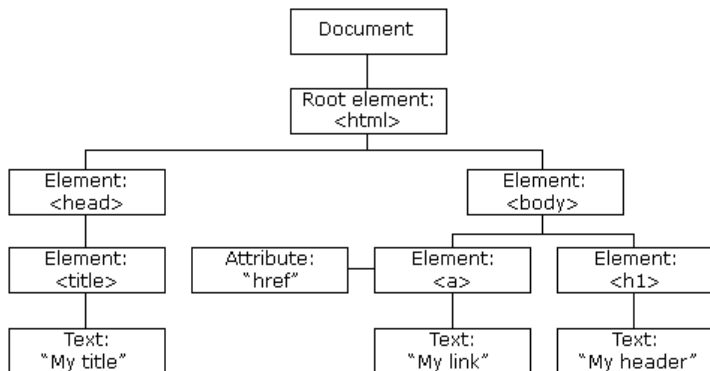
```
                              Attribute
HTML element :   <tag   attr="value" > content </tag>
              Opening tag                        Closing tag
```

Content can be empty, text, data, or other HTML elements.

DOM (Document Object Model)

The document object model specifies an API for accessing specific elements in an HTML document with code. It represents the document as a tree, with elements acting as **nodes**.

https://developer.mozilla.org/en-US/docs/Glossary/DOM

Various HTML element tags

- `<html>` The root element of the document.
- `<head>` Children of this element define metadata about the document
    - `<meta>`
    - `<title>` Specify the title of the web page in the user's browser
- `<body>`
    - `<h1>`, `<h2>`, `<h3>`, *etc.* specify different size of text headers
    - `<p>` Specifies a paragraph
    - `<a>` Contains a link inside an `href` attribute, and the displayed link text in the element's content
    - `<img>` Contains a link or data for an image
    - `<ul>`, `<ol>` Unordered (bulleted) and ordered (numbered) lists, which contain elements tagged with `<li>` for list item

HTML example

```
<!doctype html>
<html lang="en-US">
<head>
 <link href="style.css" rel="stylesheet"
       type="text/css">
 <title>My Website</title>
</head>
<body>
 <h1>Hello, world!</h1>
 <p>Here are some useful sites:</p>
 <ul>
  <li><a href="www.google.com">Google</a></li>
  <li><a href="www.github.com">GitHub</a></li>
 </ul>
</body>
</html>
```

CSS (Cascading Style Sheets)

An HTML document contains what is to be displayed and logical relations between elements with the DOM, but it doesn't specify **how** to display the document.

Attributes can control style of particular elements, but this leads to more clutter in the document. CSS organizes all style information about HTML elements by writing rules for how to style content inside each tag.

CSS can be embedded into an HTML document with
```
<link href="style.css" rel="stylesheet" type="text/css">
```

https://developer.mozilla.org/en-US/docs/Glossary/CSS

Writing CSS

A CSS file is a list of **rules** that define how sections of the webpage should be styled. Each rule is made up of a **selector** followed by `property: value;` declarations inside curly brackets.

E.g., the following applies the color red and font size 18 to all paragraphs on the page:

```
p {
    color: red;
    font-size: 18px;
}
```

Tags are the most common selector, but there are many others that may be used. See:

https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Basic_selectors

JavaScript

- Scripting language for making webpages more dynamic
- Uses the DOM to access elements in the document
- Embedded into HTML documents using the `<script>` tag
  - Keep scripts in a separate file and embed them with
    `<script type="text/javascript" src="scr.js"></script>`
- Event-driven programming style where functions are ran based on user interactions

```
<head>
  <script>
    function hi() { alert("Hello,world!"); }
  </script>
</head>
<body>
  <button onclick="hi()">Hello</button>
</body>
```

JavaScript example

In this example, the document is queried for the first <p> tag, and the function modify changes its style and content.

```
function modify() {
  const p = document.querySelector("p");
  p.textContent = "foo";
  p.style.color = "rgb(39,236,170)";
}


<body>
  <p>Some text to be modified</p>
  <button onclick="modify()">Modify</button>
</body>
```

JavaScript example 2

In this example, the document is queried for the first tag with the id attribute `modify`. This is more precise in the event a new paragraph is added before the one to be modified.

```
function modify() {
  const p = document.getElementById("modify");
  p.textContent = "foo";
  p.style.color = "rgb(39,236,170)";
}


<body>
  <p>Some text to be modified</p>
  <button onclick="modify()">Modify</button>
</body>
```

PHP

- Language for generating HTML programmatically
- Ran by the server when the user requests a page
- Written in `.php` files that can use a mixture of HTML, and PHP code delimited by the **pseudotag** `<?php ... ?>`

```
<head>
  <title>PHP Test</title>
</head>
<body>
  <?php
    for ($i = 1; $i <= 3; $i++) {
      echo '<p>Hello World</p>';
    }
  ?>
</body>
```

https://php.net/manual/en/tutorial.firstpage.php

Lost? See...

- Web page not updating? Use Ctrl+Shift+R (or ⌘+Shift+R on macOS) to perform a hard refresh of the page

- Forbidden even though you have files in `public_html`? Get the script `wget people.eecs.ku.edu/Ĩ367r860/grant_permission.sh`, and run it with `bash grant_permission.sh`

- https://developer.mozilla.org/en-US/docs/Web/HTML

- https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Getting_started

- https://developer.mozilla.org/en-US/docs/Web/JavaScript

- https://php.net/manual/en/tutorial.php