HTML, DOM & Events

Lesson Objectives

- Read and write simple HTML5 document.
- Learn to host web page on the GitHub pages.
- Learn to validate HTML document
- Basics of DOM and event handling

Lesson Outline

- Read and write simple HTML5 document.
- Basics of DOM and event handling: JavaScript in the browser

Hypertext Markup Language (HTML)

- It's a markup language (not really a programming language) used to demarcate sections of a web page.
  - Nearer to word processing application (e.g., MS Word) than the programming language.
- HTML5 is the latest version of the language

Structure of an HTML5 page

- The **header** describes the page, and the **body** contains the page's contents
  - An HTML page is saved into a file ending with extension **.html**
- **DOCTYPE** html tag tells browser to interpret our page's code as HTML5.
- HTML is case insensitive, but we follow conventions.

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

HTML Elements

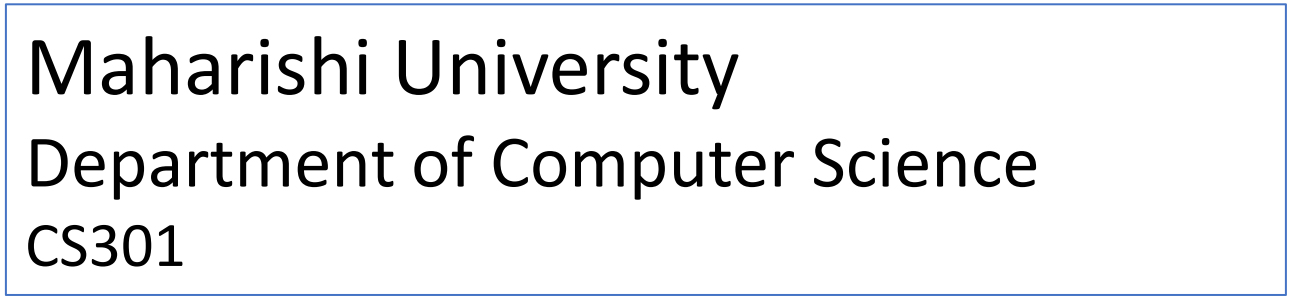
- HTML element is composed of opening tag, optional text, and closing tag.
  - Syntax: **`<element>`**content **`</element>`**
  - Example: **`<p>`**This is a paragraph**`</p>`**

- Most whitespaces are insignificant in HTML (ignored or collapsed to a single space)

Headings: `<h1>, <h2>, ..., <h6>`

Headings to separate major areas of the page (block)

```html
<h1>Maharishi University</h1>
<h2>Department of Computer Science</h2>
<h3>CS301</h3>
```



Maharishi University

Department of Computer Science

CS301

More headings examples

Paragraph: <p>

Paragraphs of text (block)

```html
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Nam enim nisl, adipiscing quis
ultrices a, egestas quis lorem. Pellentesque
ultrices nunc id mauris posuere pulvinar.</p>
```

- Placed within the body of the page

More paragraph examples

Phrase elements : `<em>`,`<strong>`

- em: emphasized text (usually rendered in italic)
- strong: strongly emphasized text (usually rendered in bold)

```
<p> HTML is <em>really</em>,
<strong>REALLY</strong> fun! </p>
```

HTML is *really*, **REALLY** fun!

- As usual, the tags must be properly nested for a valid page.

Links: <a>

- Links, or "anchors", to other pages (inline)
- **`href`** can be absolute or relative
- Anchors are inline elements; must be placed in a block element such as **`p`** or **`h1`**

```
<p> Search
       <a href="http://www.google.com/">Google</a> or our
       <a href="aboutme.html">About Me</a>.
</p>
```

Search Google or our About Me.

Exercise

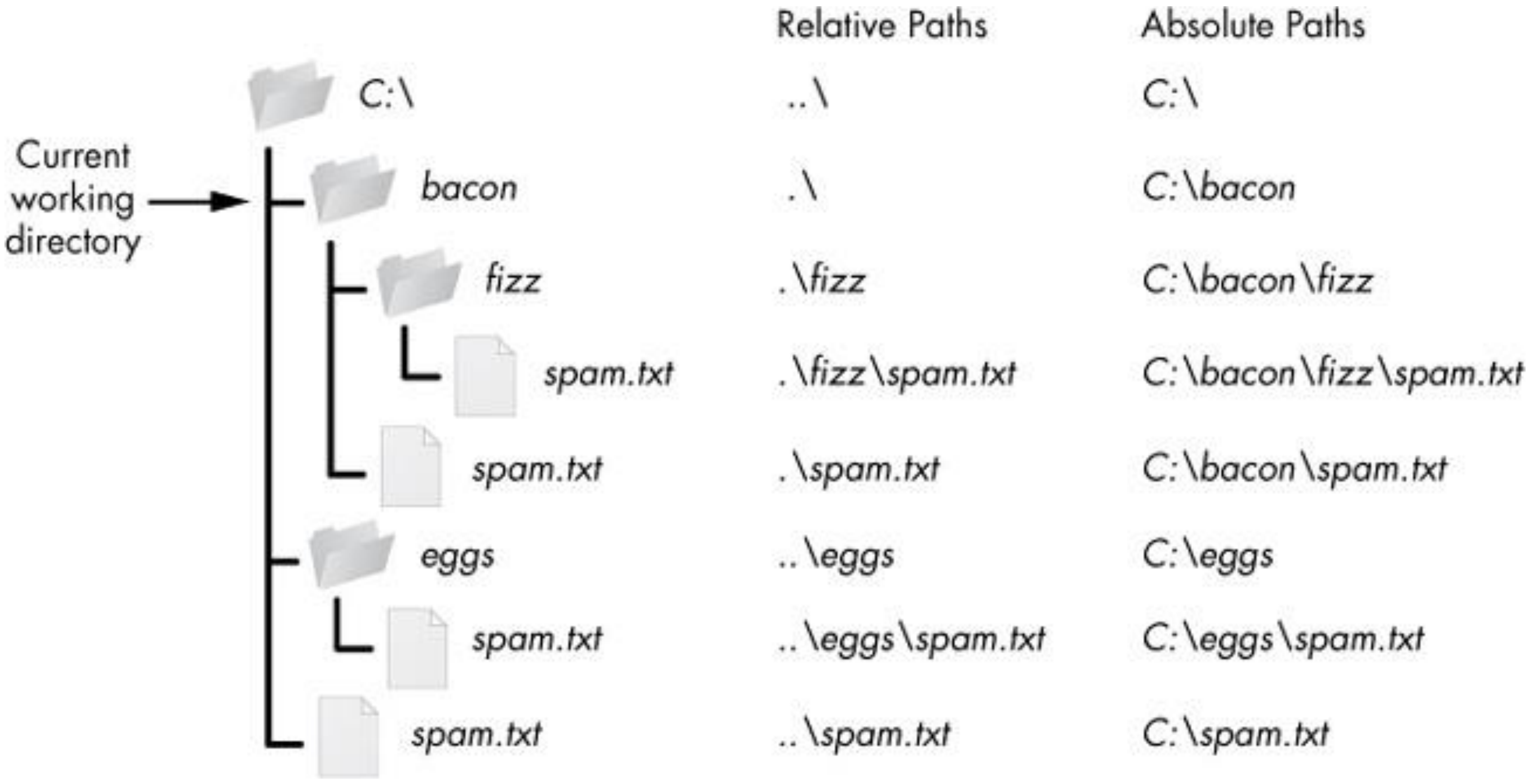
- Write a HTML page with your full name inside <h1> tag followed by a short paragraph about yourself.
    - Use phrase element to emphasize some texts.
- Inside the paragraph, add a link to the IMDB page for one of your favorite movie.

Images: <img>

- Inserts a graphical image into the page (inline)
- The `src` attribute specifies the image URL
- HTML5 also requires an `alt` attribute describing the image
- `title` attribute is an optional tooltip (on ANY element)

```html
<img src="../images/whiteLily.jpg" alt="White Lily" />
```

Relative vs absolute path



	Relative Paths	Absolute Paths
C:\	..\	C:\
bacon	.\	C:\bacon
fizz	.\fizz	C:\bacon\fizz
spam.txt	.\fizz\spam.txt	C:\bacon\fizz\spam.txt
spam.txt	.\spam.txt	C:\bacon\spam.txt
eggs	..\eggs	C:\eggs
spam.txt	..\eggs\spam.txt	C:\eggs\spam.txt
spam.txt	..\spam.txt	C:\spam.txt

Exercise

- Insert an image of your choice on the top of the page.
  - If the image you choose is too big you can also use `height` and `width` property of image tag to fix the size.

```
<img src="img.jpg" alt="alt text" width="500" height="600">
```

Displaying List Items

- Unordered List
- Ordered List

Unordered list: `<ul>, <li>`

- `ul` represents a bulleted list of items (block)
- `li` represents a single item within the list (block)

```html
<ul>
   <li>No shoes</li>
   <li>No shirt</li>
   <li>No problem!</li>
</ul>
```

- No shoes
- No shirt
- No problem!

Ordered list: `<ol>`

- `ol` represents a numbered list of items (block)
- We can make lists with letters or Roman numerals using CSS (later)

```
<p>RIAA business model:</p>
<ol>
   <li>Sue customers</li>
   <li>Profit!</li>
</ol>
```

RIAA business model:

1. Sue customers
2. Profit!

Exercise

- Add a list of your 2-3 favorite movies to your page and link each of the list item (movie) to the IMDb.

Form input elements

Tag	Purpose
`<input type="text">`	Make a text field (single row)
`<input type="button" value="Click Me">`	Makes a button that user can click with label "Click Me"
`<button>Click Me</button>`	Makes a button that user can click with label "Click Me"
`<textarea></textarea>`	Makes text area (multiple row)

Exercise

- Quickly add a textbox, a textarea and a button to your html page.

Block vs Inline elements

- A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). e.g., <p>, <h1>
- An inline element does not start on a new line and only takes up as much width as necessary. e.g., <input/>, <textarea>
  - Need to use line break **`<br/>`** to move to the new line.

Exercise

- User `<br/>` tag to fix spacing between the elements.

Comments: `<!--    -->`

- Comments to document your HTML file or "comment out" text
- Many web pages are not thoroughly commented (or at all)
- Useful at the top of page and for disabling code

```
<!-- My web page, by Suzy Student CSE 190 D, Spring 2048 -->
<p>CSE courses are <!-- NOT --> a lot of fun!</p>
```

W3C HTML Validator

- Checks your HTML code to make sure it follows the official HTML syntax
- Pickier than the browser
- Only works if the page opens from a public web server,
  - not a file in a browser
  - Not Live Server

```
<p>
    <a href="https://validator.w3.org/check/referer">Validate</a>
</p>
```

Main points

- HTML is a very simple language, once you know the content and organization of your document, you simply use the right set of tags to demarcate content you want to display in browsers. *Science of Consciousness*, *Transcendental Meditation is a very simple meditation technique, you simply take the correct angle and let go*.

Lesson Outline

- Read and write simple HTML5 document.
- Basics of DOM and event handling: JavaScript in the browser

JavaScript in browser

- JavaScript was originally developed to work with HTML pages in a browser application.

`<script>` element

- The `<script>` element is used to embed JavaScript codes.
- It can go anywhere in the HTML page, but by convention it is placed in the head section.

window

- When JavaScript runs on a browser, it runs inside the global environment called window (object).
  - `alert()` and `prompt()` are methods (functions) of window object for alerting output and displaying prompt for user input.

Running a JavaScript program

- The computer (browser) runs or executes JavaScript code when certain events happen.
  - When HTML document is loaded in a browser.
  - When user clicks a button
  - When user types in a textbox
  - and more…

HTML Event Attributes

```
<button onclick = "doSomething()">Do it</button>

<script>
    function doSomething(){
        // code to do something.
    }
</script>
```

Example 1

- Example, hello world on button click.
  - Note: `alert()` is being used for output instead of `console.log()`

Example 2

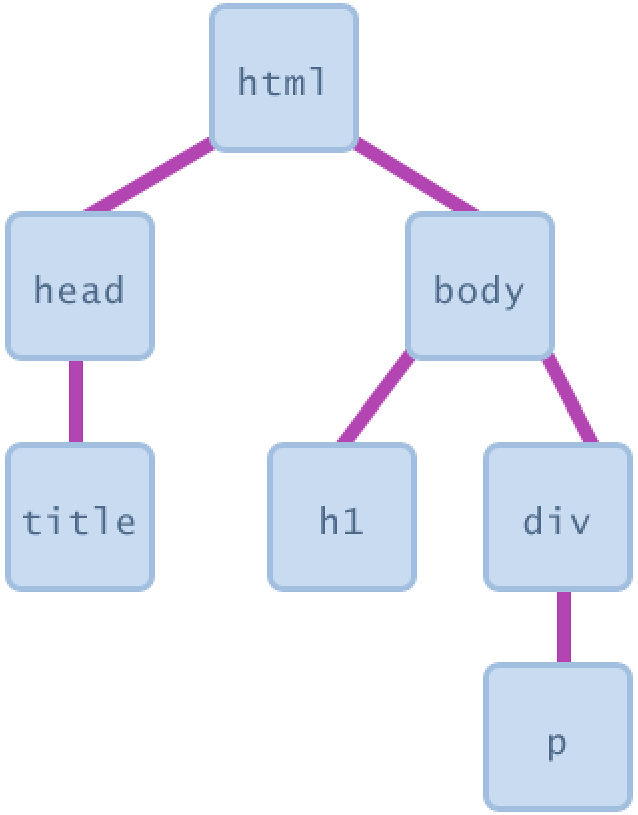
- Program to ask for name using prompt and display greeting using alert.
  - Note: don’t require Node module anymore
  - now running JavaScript inside browser’s environment.

DOM Interactions

- Prompt and Alert may work for simple input/output requirements but when writing real web applications, we very rarely use Prompt and Alert.
- Instead, we use form input tags for getting user inputs and update the webpage based on some events to change outputs.
    - For being able to do this we need to learn to use something called DOM.

Document Object Model (DOM)

- All HTML elements are represented in browsers as objects
- All objects are nested together in one tree (DOM tree)
- Elements can have parents, siblings and children
- Most JS code manipulates elements (objects) on the DOM
  - it can change state (insert some new text into a span)
  - it can change styles (make a paragraph red)



HTML element's id attribute

- The id attribute specifies a unique identifier for a HTML element (the value must be unique within the HTML document)
- The id attribute is used to target elements in JavaScript (via. the HTML DOM)

Getting a DOM element using its id

- `document.getElementById("id")`
  - Get the element with the specified id.

- Example, Program to get first name and last name from the input fields and display full name inside span element on button click.

Other DOM selection APIs

- getElementsByName("name")
  - Get all the elements with the specified name
- getElementsByTagName("tag")
  - Get all the elements in the document with the specified tag name
- querySelector("selector")
  - Get the first element in the document that matches the specified CSS selector(s) in the document.
- querySelectorAll("selector")
  - Returns all the elements in the document that matches a specified CSS selector(s)

Main point

- JavaScript programs run on events, which makes JavaScript an event driven programing language. We also perform our actions based on events that are either external or internal. *When we establish our self at the field of pure consciousness, we perform conscious actions rather than the compulsive ones.*

JavaScript in a separate file

- JS code can be placed directly in the HTML file's body or head
  - but this is not a good practice.
- script code should be stored in a separate .js file
  - script tag in HTML should be used to link the .js files

```
<script src="filename" type="text/javascript"></script>
```

- When more than one script file is included
  - interpreter treats them as a single file;
  - share global context.
  - order in which file files are loaded matters
    - interpreter executes code as soon as it hits the `<script>` tag.

Assignments

- assignment.docx

Readings

- Introduction to HTML (w3schools.com)
- HTML Basic (w3schools.com)
- HTML Elements (w3schools.com)
- HTML Attributes (w3schools.com)
- HTML Headings (w3schools.com)
- HTML Paragraphs (w3schools.com)
- HTML Forms (w3schools.com)