Class Outline

Day 1

Basic Terms

- Internet and World Wide Web
- Web Servers and Web Browsers
- Peer-to-Peer and Client/Server Architecture
- HTTP and HTTPS
- Web Page and HTML
- URL and FQDN
- IP Address and TCP Port
- Static Web Pages and Dynamic Web Pages
- Server-side Script and Client-side Script

Day 2

Writing a Very Basic Web Page

- Tools needed
- Writing the content
- Verifying the syntax
- Checking the output
- Checking the page source from the browser

Text Layout in HTML

- Headings: `<h1>` to `<h6>`
- New Line: `<br>`
- Non-breaking Space: ` `
- Paragraph: `<p>`
- Horizontal Rule: `<hr>`

Text Formatting in HTML

- Bold, Strong: `<b>, <strong>`
- Italics, Emphasized: `<i>, <em>`
- Underline, Inserted Text: `<u>, <ins>`
- Strike-through, Deleted Text: `<s>, <del>`
- Subscript: `<sub>`
- Superscript: `<sup>`
- Pre-formatted Text: `<pre>`
- Text Highlight: `<mark>`
- Small Text: `<small>`
- Quote, Block-quote: `<q>, <blockquote>`
- Abbreviation: `<abbr>`

Lab Work

1. Create a Web Page which shows how to use *Comments* in HTML.
2. Create a Web Page to demonstrates the use of all the *Heading tags*.
3. Check how to use the following tags, and create a Web Page to show their output: `<strong>`, `<em>`, `<ins>`, `<s>`, `<del>`, `<pre>`, `<sub>`, `<sup>`, `<small>`, `<mark>`, `<q>`, `<blockquote>`, `<abbr>`, `<code>`, `<samp>`, `<kbd>`, `<var>`
4. Check how to insert special characters (eg. Copyright symbol, Trademark Symbol etc.), and Emojis in a Web Page.

Day 3

Lists in HTML

- Unordered Lists: `<ul>, <li>`
- Ordered Lists: `<ol>, <li>`
- Description Lists: `<dl>, <dt>, <dd>`

Using image in a Web Page: `<img>`

Links in HTML (Hyperlinks)

- The anchor tag: `<a>`
- Absolute vs. Relative link
- Target of a link: `_self, _blank, _parent, _top`
- Image as a link
- Link to E-mail address
- Bookmark in HTML

Table in HTML: `<table>, <tr>, <th>, <td>`

Lab Work

1. Create a Web Page to show how to use *Ordered Lists* and *Description Lists*.
2. Create a Web Page to demonstrate the use of *Nested Lists*.
3. Create a Web Page to show the use of *image as a link*.
4. Create a Web Page to create a *link to an e-mail address*.
5. Create a Web page to show the use of *bookmark*.
6. Create a table in a Web Page to show the usage of `<caption>` tag, and `colspan, rowspan` properties.

Day 4

Inline Frame in HTML: `<iframe>`

Image Map in HTML: `<map>`

Playing audio & Video in an Web Page: `<audio>, <video>`

Page Redirect in HTML

Semantic elements in HTML5.

Lab Work

1. Create a Web Page to show how to load a HTML document in an iFrame, when the user clicks on a link (i.e. iFrame as the target of a link).
2. Create a Web Page to show how to play a YouTube video in the page.
3. Create a Web Page to play a video file in the page.
4. Create a Web Page to show how to include a *Favicon*.

Day 5

Working with Forms: `<form>`

- Form attributes:
  - `method`:
    - `get` (default)
    - `post`

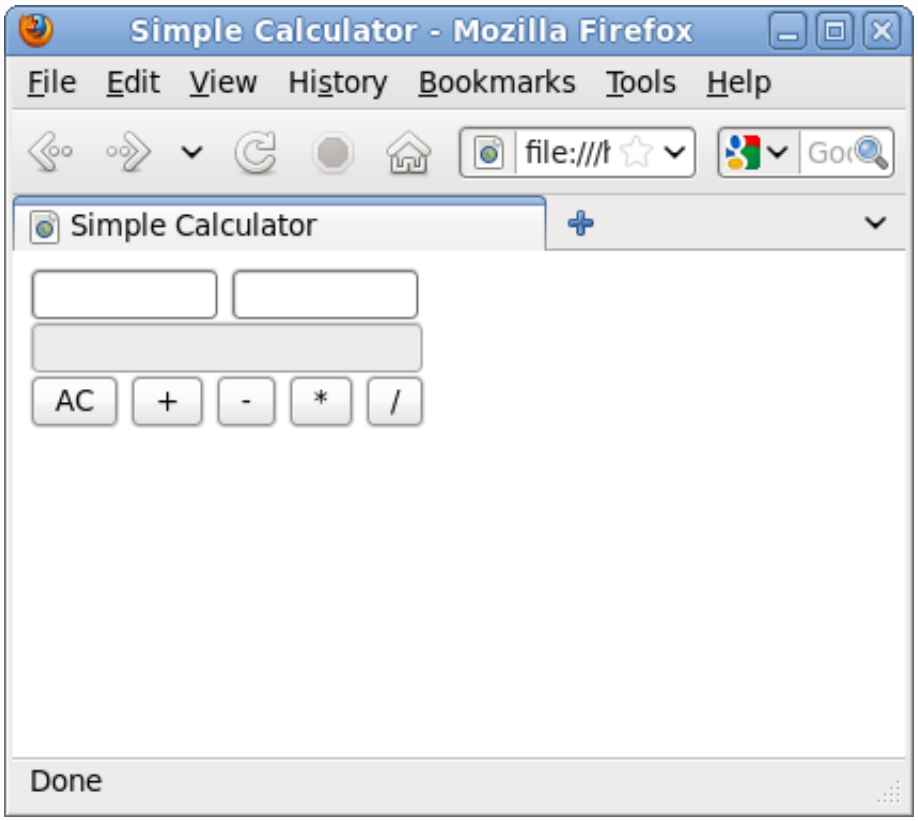
    - `enctype:` (for 'post' method only)
        - `application/x-www-form-urlencoded` (default)
        - `multipart/form-data`
        - `text/plain`
    - `name`
    - `action` , and few more...
- Form elements:
    - `<input>`
    - `<label>`
    - `<select>`
    - `<textarea>`
    - `<button>`
    - `<option>`
    - `<optgroup>`
    - `<fieldset>`
    - `<legend>` , and few more...
- `<input>` types:
    - `text` (default)
    - `password`
    - `button`
    - `checkbox`
    - `radio`
    - `date`
    - `color`
    - `file`
    - `hidden`
    - `submit`
    - `reset` , and few more...
- `<input>` attributes:
    - `value`
    - `readonly`
    - `disabled`
    - `size`

- `maxlength` , and few more...

Lab Work

1. Create a Web Page to demonstrate the following `<form>` elements:
   - `<fieldset>` and `<legend>`
   - input types: `date, file` and `color`
   - `<datalist>`
   - `<optgroup>`
2. Create a Web Page to generate a form as shown below. The form should have the following features:

   - The first two text boxes must not allow input of more than 10 characters.
   - The contents of the third text box should not be modifiable.
   - Clicking on the 'AC' button should clear the contents of the first two text boxes.



Day 6

Introduction to CSS (Cascading Style Sheets)

- What is CSS
- Why use CSS
- A Simple Demo

Ways to include styling information in HTML (i.e. different ways of using CSS)

1. Inline (inside a tag)
2. Internal (within the <head> tag)
3. External CSS File

CSS Cascading Order

- Browser default / No style sheet (**Lowest priority**)
- Internal or External style sheet (based on order of inclusion)
- Inline styles (**Highest priority**)

Note: Multiple style sheets can be used in a single HTML file. Moreover, all the different types of style sheets (inline, internal, external) can also be used in a single HTML file. In those cases, unique style properties would be cascaded.

Day 7

HTML Generic Elements:

- (block level)

- (inline)

CSS Syntax

- Rules
- Selector
- Declaration
- Property
- Value

Some of the Selector Types:

1. Element Name (eg. p, h1, a)
2. Class (eg. .class_name)
3. ID (should be unique in a document) (eg. #id_name)
4. Pseudo-selector or State (eg. :hover)
5. Class or ID attached to an element (eg. h1.class_name)
6. Grouping of Selectors
7. The Default Selector (Specified with an asterisk: *)

The CSS Box Model

- Margin
- Border
- Padding
- Element

Lab Work:

1. Create a web page to demonstrate different ways to specify colors in CSS (*Predefined color names, RGB, RGBA, Hex, HSL, HSLA*).
2. Create a web page to demonstrate some of the common font

properties in CSS (eg. `font-family`, `font-style`, `font-weight`, `font-variant`, `font-size`).
3. Create a web page to demonstrate how to use *Web Fonts* in CSS.

Day 8

Comments in CSS (`/* */`)

Example usage of `<div>`

Web Page layout using `<div>` **(along with the `'float'` & `'clear'` properties)**

The Grid Layout

Lab Work:

1. Create a web page to demonstrate how to create *rounded corners, box shadow,* and *text shadow* in CSS.
2. Create a web page to demonstrate some of the *2-D Transforms* in CSS (eg. `translate()`, `rotate()`, `scale()`, `skew()`).
3. Create a web page to demonstrate *transitions* in CSS.
4. Create a web page to demonstrate how to *animate* objects using `@keyframes` in CSS.

Day 9

Media Query: `@media`

- Syntax:

```
@media not|only mediatype and (expressions) {
CSS-Code;
}
```

Responsive Web Design

- The viewport syntax:
  `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

Lab Work:

1. Create a web page to demonstrate the use of *Media Queries* in CSS.
2. Create a web page to demonstrate *Responsive Web Design* in CSS.

Day 10

Introduction to JavaScript (from Wikipedia)

- A high-level scripting language
- It is one of the *core* technologies of the World Wide Web, along with HTML and CSS
- All current major web browsers natively supports JavaScript
- More popular on the client side, though server side code can also be written (eg. Node.js)
- Uses JIT (Just-in-Time) compiler for improving execution

performance
- Conforms to the ECMAScript standard
- Object oriented (prototype-based)
- Supports *event-driven, functional*, and *imperative* programming styles
- Uses *Dynamic Typing*, and is *Weakly Typed*
- **JavaScript ≠ Java**
- Some third-party JavaScript libraries / web frameworks:
  - jQuery
  - React.js (Created and used by Facebook/Meta. Also used by Twitter)
  - AngularJS, Angular (Created by Google and used in YouTube, Gmail)

Some of the things that JavaScript can do: **(from w3schools)**

- Change HTML content
- Change HTML attribute values
- Change styles (CSS)
- Show / Hide HTML elements

Fundamental Syntactical Features:

- It is a *Case-sensitve* language
- Variables uses dynamic typing
- Variable names can contain $, _, letters, and numbers. Numbers can not be the first character in variable names
- Data Types:
  1. Primmitive Types:
     - Boolean
     - Null
     - Undefined
     - Number (64-bit, Double Precision)

    - String (immutable)
    - Symbol: +Infinity, -Infinity, NaN
2. Objects

Writing simple client-side JavaScript code:

- Tools needed
- Checking the output
- Browser support
- Debugging

Ways to include JavaScript in a Web Page:

- Inside HTML elements
- In the <head> section of HTML
- In the <body> section of HTML
- As an external file

Day 11

Comments in JavaScript (// and /* */)

Operators in JavaScript:

- Arithmetic Operators:
    - +
    - -
    - *
    - /
    - %

    - ** *(exponentiation)*
    - ++
    - --
- Assignment Operators:
    - =
    - +=
    - -=
    - *=
    - /=
    - %=
    - **=
- String Operator:
    - + *(concatenation)*
- Comparison Operators:
    - ==
    - === *(equal to value and type)*
    - !=
    - !== *(not equal to value and type)*
    - <
    - >
    - <=
    - >=
    - ?
- Logical Operators:
    - &&
    - ||
    - !
- Type Operators:
    - `typeof`
    - `instanceof`
- Bitwise Operators:
    - &
    - |

- `~`
- `^`
- *<< (zero fill left-shift)*
- *>>> (zero fill right-shift)*
- *>> (signed right-shift)*

Output in JavaScript

- `alert()`
- `document.write()`
- `console.log()`
- `document.getElementById().innerHTML`

Declaration of Variables

- No declaration
- Using the `var` keyword
- Using the `let` keyword
  - Can not be redeclared
  - Must be declared before use
  - Have *block* scope

The `const` keyword

- Syntax:

```
const <identifier_name> = <value>;
```

JavaScript Strict Mode:

- No undefined object
- No delete of variables, objects, functions
- Can not write to read-only properties
- Can not use future reserved words as identifier name

- Syntax:

```
<script>
  "use strict";
</script>
```

Decision Making and Looping in JavaScript

- Syntax of `if`, `if-else`, `switch`, `while`, `do-while`, `for` are similar to those in C/C++/Java.

Lab Work:

1. Create a web page to demonstrate the use of JavaScript `alert()`, `confirm()`, and `prompt()`
2. Using JavaScript, check if a given integer is positive, negative, or neither (i.e. zero).
3. Using JavaScript, check if a given positive integer is prime or not.
4. Using JavaScript, calculate and print the first 20 numbers of the Fibonacci series.

Day 12

- Using the + operator with Numbers and Strings
- Use of the `let` keyword for declaring variables before use
- Using Functions
  - Simple functions
  - Functions with arguments
  - Functions with arguments and return value
- Scope of Variables

    - Global Variables
    - Local variables
    - Block scope using the `let` keyword
- Reading and Setting Form input values:
    - Using `document.<form_name>.<element_name>.value`
    - Using `document.getElementById('id').value`

Lab Work:

1. Add functionality to the 'Simple Calculator' created in *Day 5*, so that the arithmetic operations can be carried out. You may also add some styling so that the numbers and the components are properly aligned, and are of proper size.

Day 13

Arrays in JavaScript

- Syntax:
  `<identifier_name> = [val1, val2, ...];`
  or
  `<identifier_name> = new Array(val1, val2, ...);`
- Array Property: `<array_name>.length`
- Array Methods:
    - `concat()`
    - `indexOf()`
    - `lastIndexOf()`
    - `join()`
    - `pop()`

- `push()`
- `reverse()`
- `shift()`
- `unshift()`
- `slice()`
- `splice()`
- `sort()` and some more...

JavaScript Strings

- Syntax:
  `<identifier_name> = 'chars';`
  or
  `<identifier_name> = "chars";`
  or
  `<identifier_name> =`chars`;`
  or
  `<identifier_name> = new String("chars");`
- String Property: `<string_name>.length`
- String Methods:
  - `charAt()`
  - `concat()`
  - `startsWith()`
  - `endsWith()`
  - `indexOf()`
  - `lastIndexOf()`
  - `match()`
  - `slice()`
  - `split()`
  - `substr()`
  - `toLowerCase()`
  - `toUpperCase()`
  - `trim()` and some more...

JavaScript Date Object

Lab Work:

1. Create a web page to demonstrate some of the methods of JavaScript arrays (eg. `concat()`, `join()`, `indexOf()`, `sort()`, `pop()`, `push()` etc).
2. Create a web page to demonstrate some of the methods of JavaScript strings (eg. `charAt()`, `indexOf()`, `concat()`, `match()`, `replace()` etc).
3. Create a web page to demonstrate some of the methods of the Date object of JavaScript (eg. `getTime()`, `getMonth()`, `getDate()` etc).

Day 14

The *Document Object Model (DOM)* (From w3chools.com)

- It is a W3C standard for accessing documents
- It is subdivided into 3 parts:
  1. *Core DOM* (for all types of documents)
  2. *XML DOM* (for XML documents)
  3. *HTML DOM* (for HTML documents)
     - A standard *object* model and *programming interface* for HTML
     - We can manipulate all the *elements*, and their *properties, methods,* and *events* in a HTML document with it

The HTML DOM

- The `document` Object

- target
- getTargetRanges()
- getModifierState() and many more...

Using the Events

- Use as event attribute
  eg. `<h1 onclick='statement/function;'>`
- Using HTML DOM
  eg. `document.getElementById('id_name').onclick = statement/function;`
- Using `addEventListener`
  eg. `element.addEventListener('event_name', function);`

Changing Style Information (CSS)

- Using style object properties (eg. `style.color`) (Ref)
- Using `style['cssText']`

Lab Work:

1. Create a Web Page containing an image. When the user moves the mouse pointer over the image, the image should change to a new image. When the mouse pointer is moved away from the image, the original image should be shown. Use JavaScript for the implementation.
2. Create a Web Page to do form validation using JavaScript. Create a form for entering credit/debit card details. It should have a name field, card number field, and an expiry date (month & year). Use JavaScript to do the following:
   - The name field must not be blank
   - The card number field must have 16 digits

- The month in the expiry date field must only take values from 1 to 12
- The year in the expiry date field must only accept two digits

---

Day 15

Lab Work:

Let us assume that you are given the responsibility of creating a new user registration Web application. As a first step, you are required to do the following:

1. Write a simple HTML page to take user's input for registration. The page contains one input box for specifying the user id, and two input boxes to specify the password and retype the password. The input boxes should have relevant labels. The page also contains two buttons to submit the user input, and to reset the input.
2. Add an external style sheet, which sets the fonts of the page to 'sans-serif', sets the color of the texts to (hex) '#0066ff', and sets the size of the texts to `16pt`.
3. Use JavaScript to disable the 'Submit' button, until all the input boxes contains some texts.

Day 16

Basic concepts of Web Servers and Web Clients (Revisit)

- Why a Web Server is required, along with examples of some popular web servers
- What types of clients are required to access a Web Site from a Web Server, along with examples of some popular web clients (Web Browsers)
- Concept of IP Address and TCP Ports
- Basic Protocols used in the web communication, along with the standard port numbers

Installing, configuring, & testing a Web Server (Apache Web Server) in Ubuntu:

- Install the required package(s):
  - `sudo apt update`
  - `sudo apt install apache2`
- Check that the Web Server is running:
  - `systemctl status apache2`
- Host a Web Page in the Server:
  - In Ubuntu, by default a HTML file named `index.html` is present in the `/var/www/html/` directory (the document root). So, for testing purpose, we need not create our own file.
  - Run any Web Browser in the same machine, and type `localhost` or `127.0.0.1` in the address bar of the browser. The content of the HTML file should be displayed in the browser
- Enable HTTPS in the Web Server:

  By default, port 443 for `https` protocol is not enabled in the server. This can be verified with the following command:

  - `sudo apt install net-tools`
  - `sudo netstat -ant`

  So, if in the Browser, `https://localhost` is typed, the page

would not be served. To solve it, we need to enable the `ssl` module, and enable the ssl configuration:

- `sudo a2enmod ssl`
- `sudo a2ensite default-ssl`
- `sudo systemctl restart apache2`

Now, the `netstat` command would show the port number 443 in the output:

- `sudo netstat -ant`

Also, typing `https://localhost` in the address bar of the Web Browser would show the content of the page. (A warning may be shown regarding the certificate. Ignore the warning for now)

Installing, configuring, & testing a Web Server (Apache Web Server) in Fedora:

- Install the required package(s):
  - `sudo su -`
  - `dnf install httpd mod_ssl`
- Start the Web Server:
  - `systemctl start httpd`
- Verify that the Web Server is running:
  - `systemctl status httpd`
- Host a Web Page in the Server:

  Create a sample Web Page in the *Document Root* (`/var/www/html`) of the server (or copy a html file there):

  - `cd /var/www/html/`
  - `vim index.html`
  - Run any Web Browser in the same machine, and type `localhost` or `127.0.0.1` in the address bar of the browser
  - The content of the HTML file should be displayed in the

browser
- In Fedora, no additional configuration is required to enable `https`, after installing the `mod_ssl` packge. So, typing `https://localhost`in the address bar of the browser should show the output. (With a warning message. Ignore it for now)

Install PHP:

- `sudo apt install php` (in Ubuntu)
- `sudo dnf install php` (in Fedora)

Day 17

Access the hosted Web Page from a different machine (The firewall rules may have to be adjusted in some OS)

Verify that PHP is working properly in the Web Server

- Write a simple (e.g. Hello, World) PHP script, place it in the *document root* of the Web Server, and check the output in a Web Browser
- Debugging a PHP script
- The PHP interactive shell

Basics of PHP:

- PHP basic syntax
- `echo` and *short echo tag*
- Using *comments* in PHP (`//`, `#`, and `/* */`)
- Variables and Different *Arithmetic Operators*

- `if-else-elseif`
- `switch-case`
- `while`
- `do-while`
- `for`
- User Defined Functions
- Indexed Arrays, and Associative Arrays
- `foreach` with Indexed Arrays and Associative Arrays
- Including other files in PHP
- Date and Time Functions
- Page Redirection

Lab Work:

1. Using PHP, check if a given integer is positive, negative, or neither (i.e. zero).
2. Using PHP, check if a given positive integer is prime or not.
3. Using PHP, calculate and print the first 20 numbers of the Fibonacci series.