GCS 3205 – Mobile Web Application

Department of Computer Science
Gulu University

Daniel Ogenrwot

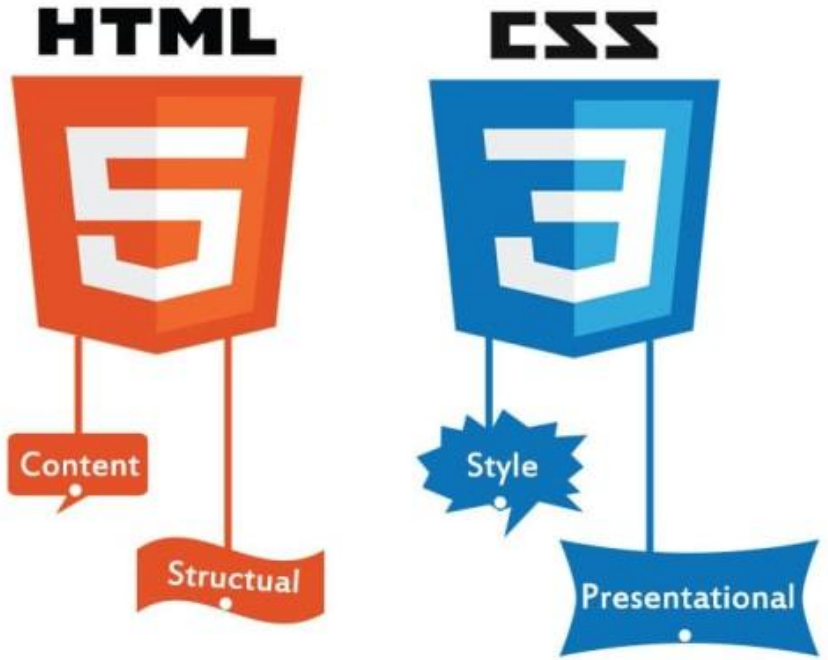
d.ogenrwot@gu.ac.ug
https://www.gu.ac.ug

October 20, 2020

Lecture-2 HTML5, CSS3

Outline

- HTML5
- CSS3



HTML5 - Overview

- HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1.
- HTML5 is a standard for structuring and presenting content on the World Wide Web.
- Browser Support: Apple Safari, Google Chrome, Mozilla Firefox, Opera, and Internet Explorer 9.0 >
- **The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.**

HTML5 – New Features

- New Semantic Elements - <headers>, <footer>, <section>, etc
- Forms 2.0 - new attributes have been introduced for <input> tag
- Persistent Local Storage - no need for third-party plugins
- WebSocket - bidirectional communication technology for web applications
- Server-Sent Events (SSE) - events which flow from web server to the web browsers
- Canvas - 2D drawing surface
- Audio & Video – no need for third-party plugins
- Geolocation – ability to share physical location with your web application
- Microdata - create your own vocabularies beyond HTML5
- Drag and drop - Drag and drop the items from one location to another on the same webpage.

HTML5 – Syntax

The DOCTYPE

DOCTYPEs in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

HTML 5 authors would use simple syntax to specify DOCTYPE as follows;

```
<!DOCTYPE html>
```

Character Encoding

```
<meta charset = "UTF-8">
```

HTML5 – Syntax

The <script> tag

BEFORE

```
<script type = "text/javascript" src=
"scriptfile.js"></script>
```

AFTER

```
<script src = "scriptfile.js"></script>
```

The <link> tag

BEFORE

```
<link rel = "stylesheet" type="text/css" href=
"stylefile.css">
```

AFTER

```
<link rel="stylesheet" href="stylefile.css">
```

HTML5 – markup overview

```
<!DOCTYPE html>
<html>
<head>
   <meta charset = "utf-8">
   <title>...</title>
</head>
<body>
<header role = "banner">
   <h1>HTML5 Document Structure Example</h1>
   <p>This page should be tried in safari, chrome or Mozilla.</p>
</header>
<nav>
   <ul>
    <li><a href = ”#">HTML</a></li>
    <li><a href = ”#">CSS</a></li>
    <li><a href = ”#">JavaScript</a></li>
   </ul>
</nav>
<article>
  <section>
   <p>Some paragraph</p>
  </section>
</article>
<aside> <p>This is aside</p> </aside>
<footer>
  <p>Created by <a href = ”#”>...</a></p>
</footer>
</body>
</html>
```

HTML5 – Web Forms 2.0

The <input> element in HTML5

HTML5 input elements introduced several new values for the **type** attribute.

datetime	range
datetime-local	email
date	url
month	time
week	number

The placeholder attribute

```
<input type = "text" name = "search" placeholder =
"search the web"/>
```

HTML5 – Audio & Video

- HTML5 features include native audio and video support without the need for Flash.
- The HTML5 `<audio>` and `<video>` tags make it simple to add media to a website. You need to set **src** attribute to identify the media source and include a controls attribute so the user can play and pause the media.

Embedding Video

```
<video src = "foo.mp4" width = "300" height = "200"
controls> Your browser does not support the <video>
element. </video>
```

Embeddding Audio

```
<audio src = "foo.wav" controls autoplay> Your browser
does not support the <audio> element. </audio>
```

HTML5 – Geolocation

- HTML5 Geolocation API lets you share your location with your favorite web sites.
- A JavaScript can capture your latitude and longitude and can be sent to backend web server.
- Do fancy location-aware things like finding local businesses or showing your location on a map
- Today most of the browsers and mobile devices support Geolocation API

```
var geolocation = navigator.geolocation;
```

The `geolocation` object is a service object that allows widgets to retrieve information about the geographic location of the device.

HTML5 – Geolocation Methods

- The `geolocation` object provides the following

Sr.No.	Method & Description
1	getCurrentPosition() This method retrieves the current geographic location of the user.
2	watchPosition() This method retrieves periodic updates about the current geographic location of the device.
3	clearWatch() This method cancels an ongoing watchPosition call.

```
function getLocation() {
   var geolocation = navigator.geolocation;
   geolocation.getCurrentPosition(showLocation,
errorHandler);
}
```

Here `showLocation` and `errorHandler` are callback methods which would be used to get actual position as explained in next section and to handle errors if there is any

HTML5 – Location Properties

Property	Type	Description
coords	objects	Specifies the geographic location of the device. The location is expressed as a set of geographic coordinates together with information about heading and speed.
coords.latitude	Number	Specifies the latitude estimate in decimal degrees. The value range is [-90.00, +90.00].
coords.longitude	Number	Specifies the longitude estimate in decimal degrees. The value range is [-180.00, +180.00].
coords.altitude	Number	[Optional] Specifies the altitude estimate in meters above the WGS 84 ellipsoid.
coords.accuracy	Number	[Optional] Specifies the accuracy of the latitude and longitude estimates in meters.
coords.altitudeAccuracy	Number	[Optional] Specifies the accuracy of the altitude estimate in meters.
coords.heading	Number	[Optional] Specifies the device's current direction of movement in degrees counting clockwise relative to true north.
coords.speed	Number	[Optional] Specifies the device's current ground speed in meters per second.
timestamp	date	Specifies the time when the location information was retrieved and the Position object created.

HTML5 – Geolocation

Demo

CSS3

- Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language.

- CSS3 is a latest standard of css earlier versions(CSS2). The main difference between css2 and css3 is follows;

  - Media Queries
  - Namespaces
  - Selector Level 3
  - Color

CSS3 - Modules

- CSS3 is collaboration of CSS2 specifications and new specifications, we can called this collaboration is **module**. Some of the modules are shown below;

  - Selectors
  - Box Model
  - Backgrounds
  - Image Values and Replaced Content
  - Text Effects
  - 2D Transformations
  - 3D Transformations
  - Animations
  - Multiple Column Layout
  - User Interface

CSS3 – Vendor Prefixes

- Some CSS rules wont work without the vendor prefix
  - Mozilla Browsers (Firefox)
    - -moz
  - Webkit Browsers (Safari, Chrome)
    - -webkit
  - Opera
    - -o
  - Internet Explorer
    - -ms
    - <!--[if IE] <![endif]-->
- Not all CSS rules work with all browsers:

CSS3 – Rounded Corners

- CSS3 Rounded corners are used to add special colored corner to body or text by using the border-radius property. A simple syntax of rounded corners is as follows

```
#rcorners {
  border-radius: 60px 15px;
  background: #FF0000;
  padding: 20px;
  width: 200px;
  height: 150px;
}
```

CSS3 – Selectors

- In CSS, selectors are patterns used to select the element(s) you want to style.

https://www.w3schools.com/cssref/css_selectors.asp

CSS3 – Selectors

- * - selects everything, every tag
- E:has(> F) – an E element that has an F child
- E[attr="str" i] – an E element with an attribute "attr" equal cu "str" (case insensitive)
- E[attr="str" s] – an E element with an attribute "attr" equal cu "str" (case sensitive)
- E:current, E:past, E:future - timeline selectors, select an element in a time-dimensional canvas; Ex.;

  :current(p) { background : yellow } - selects and colors the paragraph of text that is currently read in a speech rendering of the document

CSS3 – Selectors

- E[attr^="str"] - an E element whose "attr" attribute begins with "str"
- E[attr$="str"] - an E element whose "attr" attribute ends with "str"
- E[attr*="str"] - an E element whose "attr" attribute contains "str"
- E[attr="str"] – an E element whose "attr" attribute is equal to "str"
- E:nth-child(n) - an E element, the n-th child of its parent
- E:nth-last-child(n) - an E element, the n-th child of its parent, counting from the last one
- E:nth-of-type(n) - an E element, the n-th sibling of its type
- E:nth-last-of-type(n) - an E element, the n-th sibling of its type, counting from the last one
- E:first-child - an E element, first child of its parent (in CSS2.1)
- E:last-child - an E element, last child of its parent

CSS3 – Text shadow & box-shadow

- text-shadow : h-shadow v-shadow blur color;
- Ex.: div {

```
        text-shadow: 2px 2px 4px #ff00dd;
    }
```

- box-shadow : h-shadow v-shadow blur spread color
- Ex. :

```
    div {
        box-shadow: 4px 6px 6px -2px #aaaaff;
    }
```

CSS3 – Transitions

- add effects when changing from a style to another (e.g. when :hover is used), like flash or javascript
- Style properties used:
  - transition-property : comma separated list of property names to which transition is applied
  - transition-duration : how long it take the transition to be completed (ex.: transition-duration : 2s)
  - transition-delay : when transition will start (ex.: transition-delay: 1s)
  - transition-timing-function : defines the speed of the transition; values: linear|ease|ease-in|ease-out|ease-in-out|cubic-bezier(*n*,*n*,*n*,*n*)
  - transition : shorthand property for the above properties

CSS3 – Transitions Examples

- simple example:

```
div {
        transition-property: width;
        transition-duration: 5s;
}
div:hover { width: 100px }
```

- multiple properties example:

```
div {
        width: 20px;
        transition: width 3s, transform 3s;
}
div:hover {
        width: 100px;
        transform: rotate(90deg);
}
```

CSS3 – Beyond CSS (CSS Preprocessors)

- using functions, variables, inheritance, code reusability, operators, if, loops in CSS ?
- extend CSS syntax (compiled to standard CSS by a preprocessor written in node.js, ruby etc.)
- Stylus
- Compass
- Less
- Sass

CSS Preprocessors - less/scss

```
@font-size: 16px;

.bordered (@width) {
        border: @width solid #ddd;
        &:hover {
                border-color: #999;
        }
}

h1 { .bordered(5px);
     font-size: @font-size;
}
```

Responsive Web Design

- Document should look good on any device (desktop, tablet, phone)
- Use CSS to shrink, enlarge, hide or move html content in order to look good on any screen
- Setting viewport:

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

- sets the width of the page to follow the screen width of the device

- sets the initial zoom level to 1 (no zoom)

- Do not rely on a particular viewpoint
- Use relative (percentage, %) dimensions: width, height
- Use flexbox for layout
- Use media-querries to apply different styles to large/small screens

Responsive Web Design

- Use relative units for dimensions:
  - percentages: Ex. width: 100%
  - relative to fontsize: em (relative to the root of the document) and rem (relative to the parent's font-size)
  - relative to viewport: vw, vh, vmin, vmax
  - NOT: px, pt, cm, mm, in (these are absolute units)

Responsive Web Design

- Use @media to apply css style only if a condition is met

```
@media only screen and (max-width: 500px) {
    #div1 {
        width: 100%;
    }
}

@media only screen and (min-width: 500px) {
    ....
}
@media only screen and (orientation: landscape) {
    ...
}
```

Neat CSS icons: google icons

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">

<i class="material-icons" style="font-size:48px;color:red">folder</i>
<i class="material-icons" style="font-size:48px;color:red">
cloud_upload</i>
```

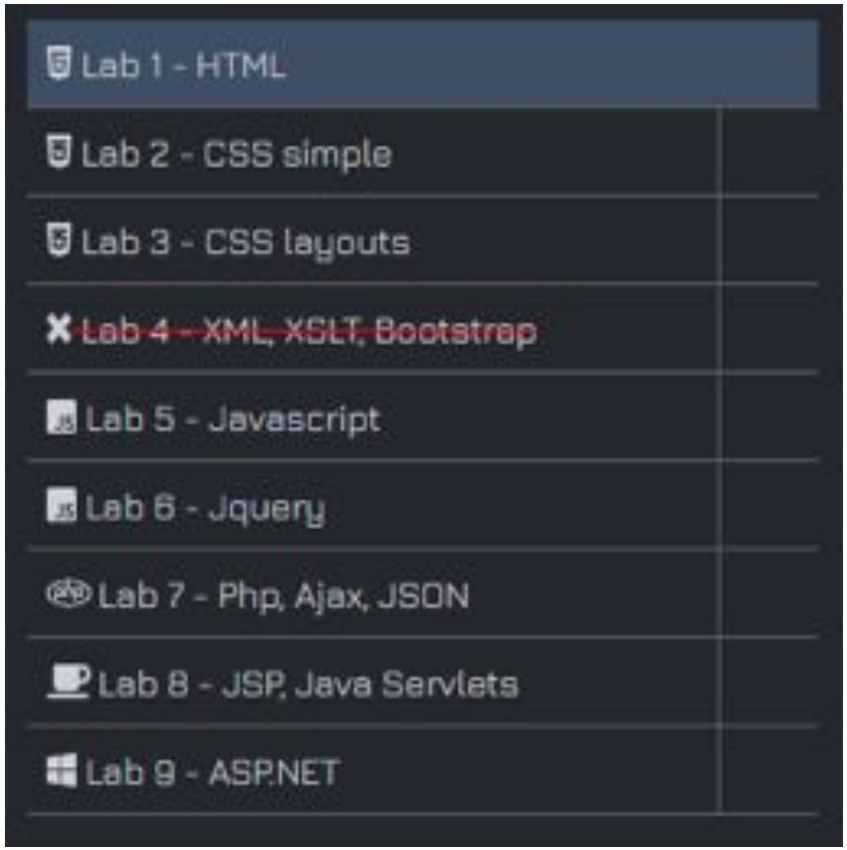
Neat CSS icons: fontawesome

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

<i class="fa fa-car" style="font-size:60px;color:red;"></i>
```

Neat CSS icons: fontawesome

```
<script defer src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
<li><div>
    <a href="#" id="1">
      <i class="fab fa-html5"></i>
       Lab 1 – HTML
    </a>
</div></li>
<li><div>
     <a href="#" id="2">
       <i class="fab fa-css3-alt"></i>
        Lab 2 - CSS simple
     </a>
</div></li>
...
```



THANK YOU