

Lab Exercise: Multimedia & Geolocation

Overview of the lab exercise:

Add a video element to a webpage using pug, and how to control various aspects of the video using JavaScript, such as play/pause, size, mute/unmute, and looping.

Use the Geolocation API to get the current geographical location of the user and display it on a Google Map using JavaScript and pug

Part 1: Embedding the Video

1. Create a pug file (i.e. vid.pug) in your current project.
2. Add <video> tag into your webpage. Add source file, you may use any video for this exercise. Ensure the id is "Vid".

```
video#vid(width='1000')
  source(src='pathtoyourvideo/videofilename.mp4', type='video/mp4')
p Your browser doesn't support HTML5 video.
```

3. Create a JavaScript file with this code, that allows you to control various aspects of a video through JavaScript, such as play/pause, size, mute/unmute, and looping.

```
var video = document.getElementById("vid");
var video_status =
document.getElementById("vid_status");
function playPause() {
  if (video.paused)
    video.play();
  else
    video.pause();
}

function bigSmall() {
  if (video.width == 1000)
    video.width = 500;
  else
    video.width = 1000;
}

function muteUnmute(){
  if (video.muted==false)
    video.muted=true;
  else
    video.muted=false;
}
```

var video = document.getElementById("vid"); - This line gets the video element with the id "vid" from the webpage and stores it in the video variable.

var video_status = document.getElementById("vid_status"); - This line gets another element with the id "vid_status" from the webpage and stores it in the video_status variable. This element is used to display the status of the video loop.

function playPause() {...} - This function plays the video if it's paused, and pauses it if it's playing.

function bigSmall() {...} - This function changes the width of the video. If the video's width is 1000, it changes it to 500. If it's not 1000, it changes it to 1000.

<pre>function loopNoloop() { var status = ""; if (video.loop == true){ video.loop = false; status = "off"; } else { video.loop = true; status = "on"; } video_status.innerHTML = "Loop " + status; }</pre>	<p>function muteUnmute(){...} - This function mutes the video if it's not muted, and unmutes it if it's muted.</p> <p>function loopNoloop() {...} - This function toggles the looping of the video. If the video is currently set to loop, it turns looping off and updates the video_status element to display "Loop off". If the video is not set to loop, it turns looping on and updates the video_status element to display "Loop on".</p>
---	---

4. Add the buttons to control the events and an element to show the loop status.
5. Link the JavaScript file above into vid.pug

Part 2: Geolocation

1. Create a pug file (i.e. map.pug) in your current project.
2. Add the button (to trigger the JavaScript function, a <div> to show the latitude and longitude, and another <div> to show the embedded map from Google Map API

```
button(onclick="getLocation()") Get Coordinate
div#latlong
div#embeddedmap
```

3. Create a JavaScript file and include a JavaScript code that uses the Geolocation API to get the current geographical location of the user and display it on a Google Map.

<pre>var x = document.getElementById("latlong"); var y = document.getElementById("embeddedmap"); function getLocation() { if (navigator.geolocation) { navigator.geolocation.getCurrentPosi tion(showPosition); } else { x.innerHTML = "Geolocation is not supported by this browser."; } } function showPosition(position) {</pre>	<p>var x = document.getElementById("latlong"); - This line gets the HTML element with the id "latlong" and assigns it to the variable x.</p> <p>var y = document.getElementById("embeddedmap"); - This line gets the HTML element with the id "embeddedmap" and assigns it to the variable y.</p> <p>function getLocation() {...} - This function is used to get the current geographical location of the user.</p>
---	---

```

x.innerHTML = 'Latitude: ' +
position.coords.latitude + '<br>Longitude: ' +
position.coords.longitude;
y.innerHTML = '<iframe width="300"
height="170" frameborder="0" scrolling="no"
marginheight="0" marginwidth="0"
src="https://maps.google.com/maps?q=' +
position.coords.latitude + ',' +
position.coords.longitude +
'&hl=es;z=14&output=embed"> </iframe>';
}

```

if (navigator.geolocation) {...} - This if statement checks if the Geolocation API is supported in the user's browser.

navigator.geolocation.getCurrentPosition(showPosition); - If the Geolocation API is supported, this line gets the current geographical location of the user and passes it to the showPosition function.

else {...} - If the Geolocation API is not supported, the else block executes and the innerHTML of the x element is set to "Geolocation is not supported by this browser."

function showPosition(position) {...} - This function is used to display the user's geographical location.

The latitude and longitude are displayed in the x element.

A Google Map with the user's location is displayed in the y element. The map is embedded in an iframe and the src attribute of the iframe is set to the URL of the Google Map with the user's latitude and longitude.

4. Link the JavaScript file above into map.pug