

Multimedia

Video

- ❖ HTML 5.0 provides a standard for showing video. Using the `<video>` element we can easily embed video within our web page.
- ❖ The video formats the `<video>` element supports include the following:
 - MPG4 (with H.264 video codec and AAC audio codec)
 - OGG (with Theora video codec and Vorbis audio codec)

Video

Content we place in between the tags will be displayed when the browser doesn't support displaying video

```
<video src="myvid.ogg" controls="controls">
```

```
</video>
```

We can use the `width` and `height` attributes in order to specify the size

the `control` attribute is for adding the play, pause and volume controls

Video

```
<h1>HTML 5 Playing Video Sample</h1>
```

```
<video
```

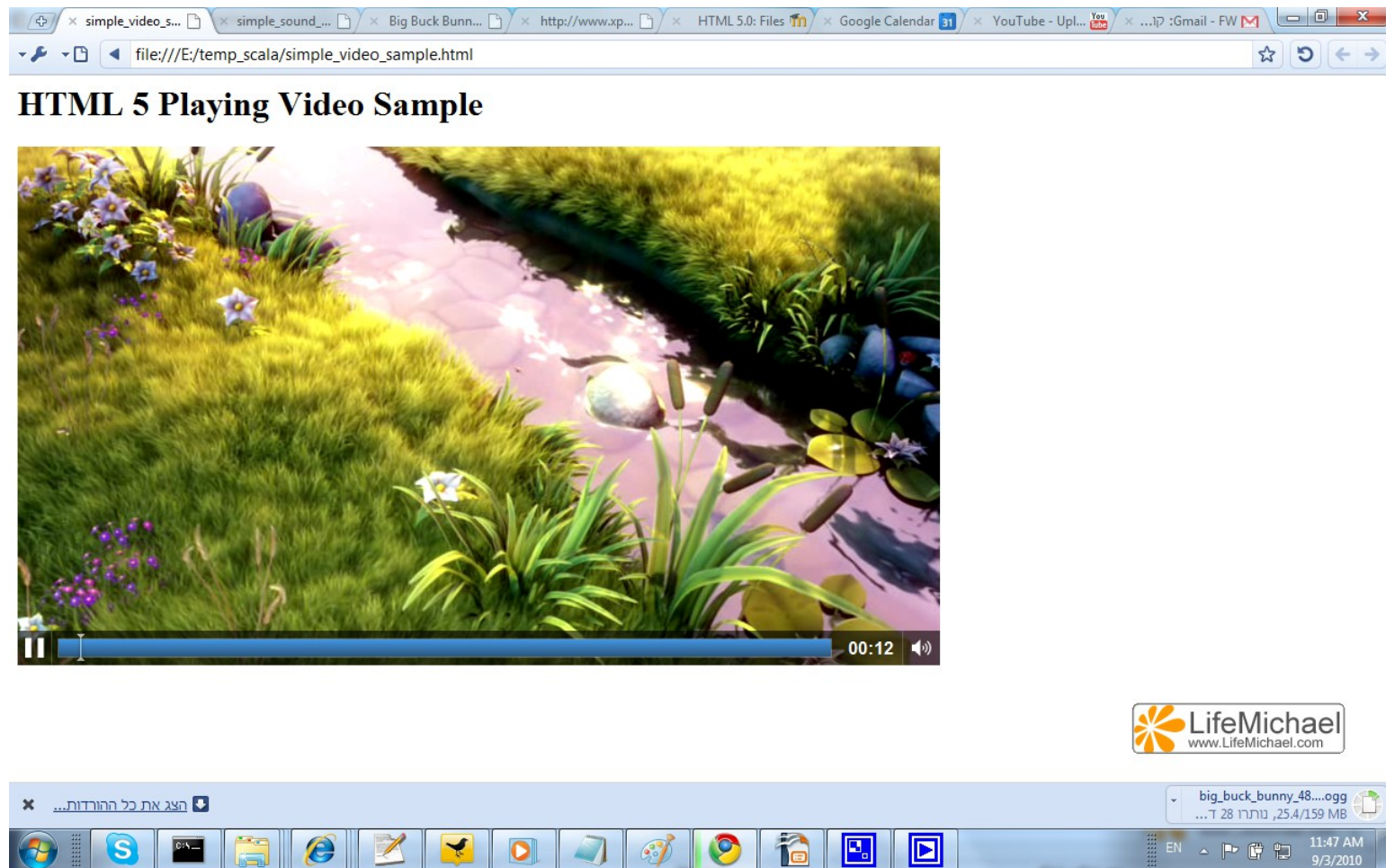
```
  src="http://mirror.bigbuckbunny.de/peach/bigbuckbunny_movies/big_buck_bunny_480p_stereo.ogg"  
  controls="controls"  
  width="854"  
  height="480">
```

```
  browser does not support html 5.0
```

```
</video>
```



Video



Video

❖ The HTML 5.0 specification supports the following attributes:

`autoplay`

we assign it with the value `automatic` in order to specify that we want the video to start playing as soon as it is ready.

`controls`

we can assign it with the value `controls` in order to specify that we want to have the video controls displayed on screen.

`height`

we can specify the height of the rectangle through which the video will be displayed.

Video

`width`

we can specify the width of the rectangle through which the video will be displayed.

`loop`

we can specify the number of times we want the video to be played.

`preload`

we can assign the preload value and by doing so specify that we want the video to be loaded when the page loads.

`src`

We use this attribute in order to specify the exact video file we want to play.

Audio

- ❖ The HTML 5.0 specification allows us playing sound using the `<audio>` element.

...

```
<audio src="mymusic.ogg" controls="controls">
```

```
</audio>
```

the control attribute adds the play, pause and volume controls

...

- ❖ The `<audio>` element can play sound files or an audio stream.

Audio

- ❖ The HTML 5.0 specification aims at supporting the following sounds formats: MP3, WAV and Ogg Vorbis.

Audio

- ❖ We can add the `<source/>` child elements in between the audio element tags. The browser will use the first supported format.

...

```
<audio controls="controls">
```

```
  <source src="mymusic.ogg" type="audio/ogg">
```

```
  <source src="mymusic.mp3" type="audio/mpeg">
```

```
  browser does not support html 5.0
```

```
</audio>
```

...

Audio

❖ The HTML 5.0 specification supports the following audio attributes:

`autoplay`

we assign it with the value `autoplay` in order to specify that we want the audio to start playing as soon as it is ready.

`controls`

we can assign it with the value `controls` in order to specify that we want to have the audio controls displayed on screen.

`loop`

we can specify the number of times we want the audio to be played.

Audio

`preload`

we can assign it with `preload` in order to specify that we want the audio to be loaded together with the page.

`src`

we assign this attribute with the URI of the audio file.

Audio

```
<h1>HTML 5 Playing Sound Sample</h1>
```

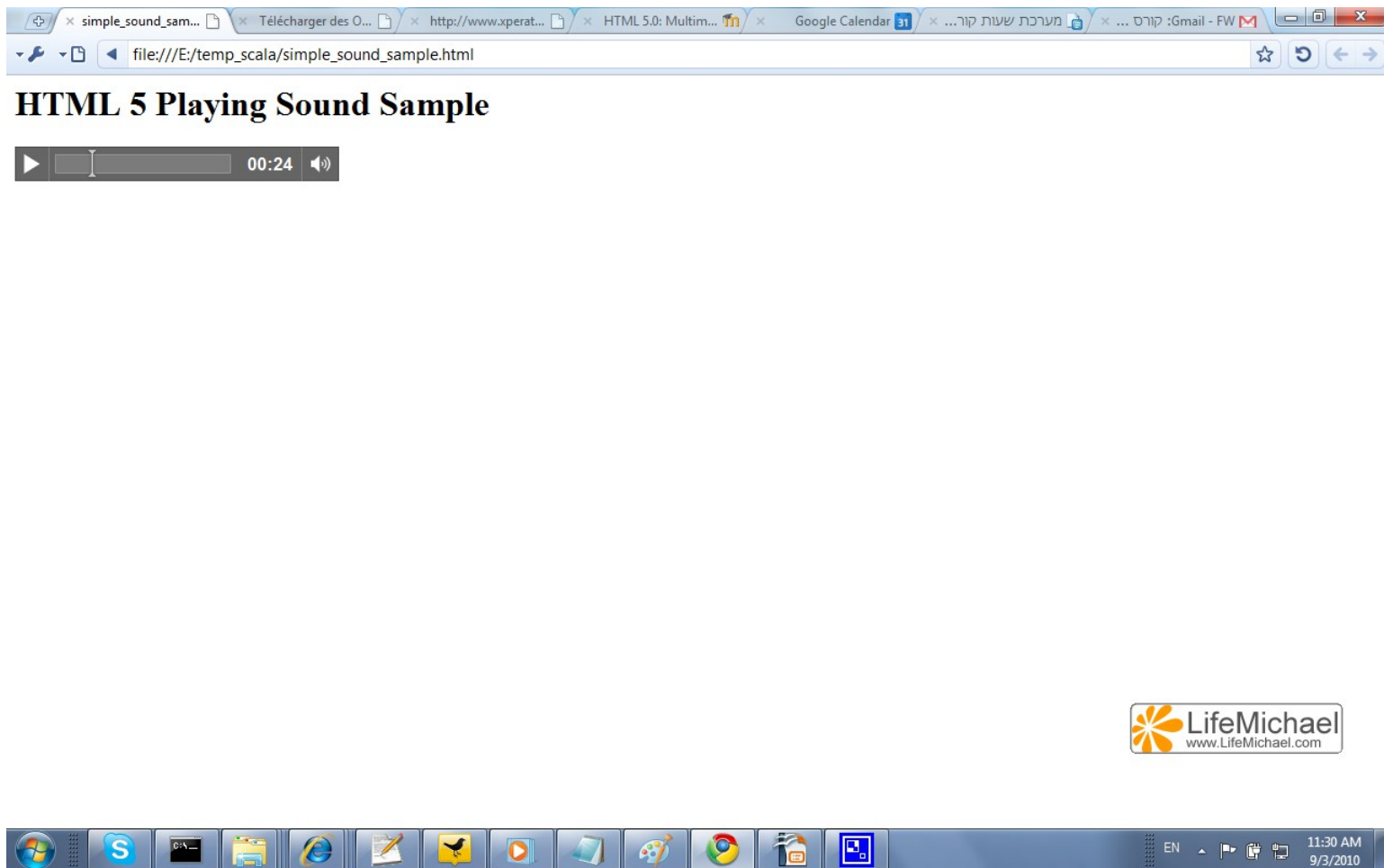
```
<audio controls="controls">  
  <source src="antony_raijekov_views_track_04_jazzabel.ogg"  
    type="audio/ogg">
```

```
  browser does not support html 5.0
```

```
</audio>
```



Audio



Capturing Audio & Video

❖ The `navigator.mediaDevices.getUserMedia()` function allows web applications to access the user camera and microphone.

Capturing Audio & Video

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <video id="vid" autoplay></video>
  <button id="show">Open camera</button>
  <script>
    var constraints = window.constraints = {
      audio: false,
      video: true
    };

    function handleSuccess(stream) {
      var video = document.querySelector('video');
      var videoTracks = stream.getVideoTracks();
      window.stream = stream;
      video.srcObject = stream;
    }
  </script>
</body>
</html>
```


Capturing Audio & Video

```
function handleError(error) {  
    console.log(`error: ${error.name}`, error);  
}  
  
async function init(e) {  
    try {  
        const stream = await navigator.mediaDevices.getUserMedia(constraints);  
        handleSuccess(stream);  
        e.target.disabled = true;  
    } catch (e) {  
        handleError(e);  
    }  
}  
  
document.querySelector('#show').addEventListener('click', e => init(e));  
</script>  
</body>  
</html>
```