

ITMD 465/565 Assignment 2- IIT News

Steps

1. Use JQuery Library

I have used the JQuery from the Internet instead of downloading it and then access to it so I save some milliseconds as after accessing once, the rest of the accesses are immediate.

```
<!-- acces to jquery from internet -->  
<script src="http://code.jquery.com/jquery-2.1.0.js"></script>
```

2. Ajax file

I have used all the data reading the clips.json file. I have created an object named clip that takes and saves the information of the json file.

```
videoapi.start = function(hookElementSelection, dataurlForJsonFile) {  
    videoapi.clips = [];  
    $.ajax({url:dataurlForJsonFile}).success(function(data) {  
        parseJSONData(data);  
        renderClips(hookElementSelection);  
        renderNonWatchedClipsFirst(hookElementSelection);  
    });  
};  
  
var parseJSONData = function(data) {  
    var clips = data;  
    $.each(clips, function(i, clip) {  
        var aClip = new videoapi.Clip(clip.id, clip.name, clip.description, clip['content-url'], clip['thumb-url']);  
        videoapi.clips.push(aClip);  
    });  
}
```

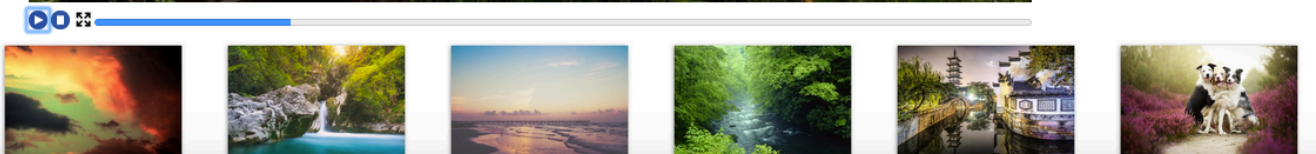
3. Arrange videos thumbnails/ Local storage, watched end of list.

There is a list displayed of the thumbnails and when selected the video will display also the title of the video.



If I select the first thumb (dogs) its video appears and now its thumb will appear at the end of the thumb list, as the videos that have been watched will be stored in the local storage and then displayed at the end. You can see now that also the title of the video is displayed.

War on Drugs continues



4. Video Description 25% / Css3 animation slide from left

Also after the 25% of the video has been played, entering from the left with an animation, the description of the video will appear. I have used the following animation:

```
@keyframes myAnimationOne {
  from { left: 0px ; transform: translate(-200px); }
  to { left: 20px ; transform: translate(200px); }
}

@-webkit-keyframes myAnimationOne {
  from { left: 0px ; transform: translate(-200px); }
  to { left: 20px ; transform: translate(125px); }
}
```

Now the video will look like the following:

War on Drugs continues



5. Custom video player controls.



I have implemented a custom media player bar with a play/pause button, stop button, full screen button and a progress bar.

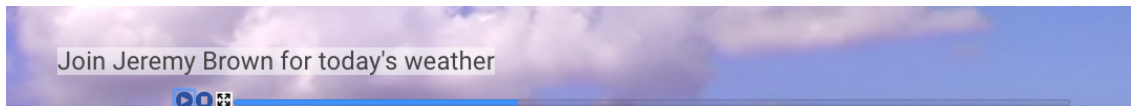
8. Thumbnail clicked.

As explained before when you click on a thumbnail, its video appears and also it starts playing automatically.

Graduate Extension

9. Theatre mode (description bottom, transition)

As you can see in my media player bar I have introduced a full screen button. When you click on it the video takes the full screen and now the description appears at the bottom of the video. Also the thumbnail list disappears. When clicked again you will return to the normal view. A Boolean variable will tell you if you are in full screen mode or not in order to hide or show elements or to add or change css features to the elements.



It also has a transition to appear:

```
.theatre video{  
  width: 100%;  
  height: 100%;  
  transition: width 2s;  
}
```

10. When finished play the next video.

When a video in full screen has ended, the event listener detects it and it automatically plays the next video on the list:

```
videoElement.addEventListener('ended', function() {  
  
  if (theatremode==true){  
    videoapi.playVideo((index+1) % videoapi.clips.length);  
  }  
  
  videoElement.pause();  
  
}, false);
```

11. Store current time when paused.

Also when a video is paused, the current time of the video and the source of the video is stored in local storage. So when you click on a thumb first it checks if it was the one paused in order to start from that time.

```
if($("#videoElement source").attr("src")==localStorage.getItem("video")) {  
  videoElement.currentTime = localStorage.getItem("time");  
}
```