

Document: Mandatory Assignment 2, JSON (json\_starter.html).

Student: Brian Munksgaard

## Exercise 1

The code from json\_starter.html is shown below:

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Video Watcher</title>
  <link rel="stylesheet" type="text/css" href="http://teaching-
materials.org/ajax/lib/bootstrap.css">
  <style type="text/css">
    /* This puts titles always on a separate line from thumbnails */
    #videos-list img {
      display: block;
    }
  </style>
</head>
<body>

<div class="container">

  <div class="row">
    <div class="span12">
      <h2 class="page-header">Best Videos Ever ♥</h2>
    </div>
  </div>

  <div class="row">
    <div class="span3">
      <ul id="videos-list">
      </ul>
    </div>
    <div class="span9" id="video-watcher">
    </div>
  </div>
</div>

  <script type="text/javascript" src="http://teaching-materials.org/ajax/lib/youtube.js"></script>
  <script type="text/javascript" src="http://teaching-
materials.org/ajax/lib/jquery.min.js"></script>
  <script>

var videos = [
  {"youtubeId": "TddFnTB_7IM",
   "title": "Trip through the 80s"
  },
  {"youtubeId": "epUk3T2Kfno",
   "title": "Otters Holding Hands"
  },
  {"youtubeId": "il2IrgFHfsg",
   "title": "The Oooh Cat"
  }
];

var addVideoToList = function(video) {

  var videoLink = $('<a>');
  videoLink.append(video.title);
  var thumbnailUrl = youtube.generateThumbnailUrl(video.youtubeId);
  var thumbnailImg = $('<img>');
  thumbnailImg.attr('src', thumbnailUrl);
  videoLink.append(thumbnailImg);

  videoLink.on('click', function(e) {
    e.preventDefault();

    var videoEmbed = $('<iframe></iframe>');
    videoEmbed.attr('src', youtube.generateEmbedUrl(video.youtubeId));
    videoEmbed.attr('width', 560);
    videoEmbed.attr('height', 315);
```

```

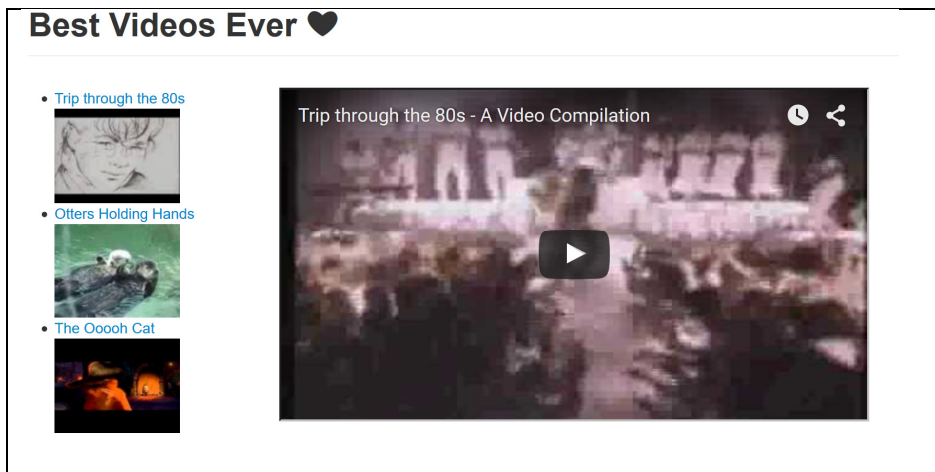
        var videoWatcher = $('#video-watcher');
        videoWatcher.hide();
        videoWatcher.html(videoEmbed);
        videoWatcher.fadeIn();
    });

    var videoItem = $('<li>');
    videoItem.append(videoLink);
    $('#videos-list').append(videoItem);
}

for (var i = 0; i < videos.length; i++) {
    addVideoToList(videos[i]);
}
</script>
</body>
</html>

```

The code lists 3 videos. When a video is clicked, the video is played in an iframe next to the list of videos.



### 1.1 Add an additional video to the JSON, so that you have 4 videos listed

An additional video is added to the page by adding:

```
{ "youtubeId": "2G5rfPISiwo", "title": "The Trooper" }
```

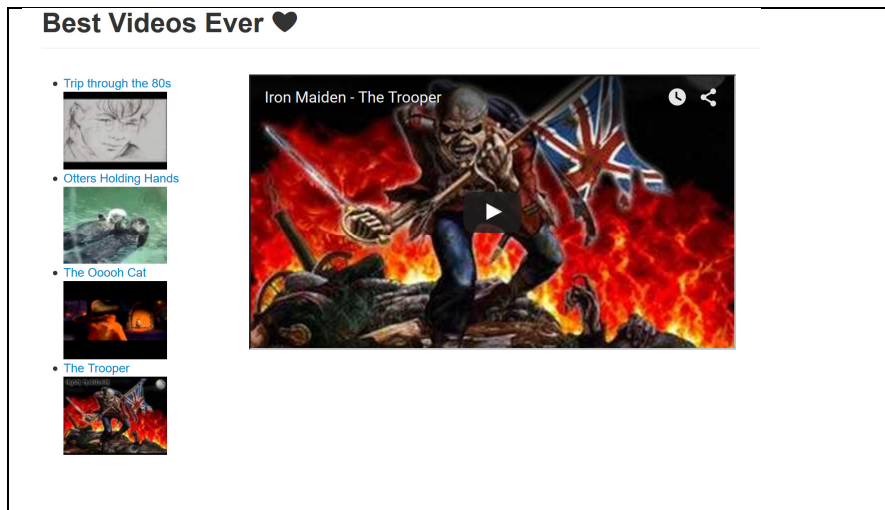
We then have:

```

var videos = [
  { "youtubeId": "TddFnTB_7IM",
    "title": "Trip through the 80s"
  },
  { "youtubeId": "epUk3T2Kfno",
    "title": "Otters Holding Hands"
  },
  { "youtubeId": "il2IrgFHfsg",
    "title": "The Ooooh Cat"
  },
  { "youtubeId": "2G5rfPISiwo",
    "title": "The Trooper"
  }
];

```

And on the page we have 4 videos:



## 1.2 Adding Video Title Header

A video title header is displayed above the video player by changing the following block of code:

```
var videoWatcher = $('#video-watcher');
videoWatcher.hide();
videoWatcher.html(videoEmbed);
videoWatcher.fadeIn();
```

To:

```
var videoWatcher = $('#video-watcher');
videoWatcher.hide();
videoWatcher.html("<H2>" + video.title + "</H2>")
videoWatcher.append(videoEmbed);
videoWatcher.fadeIn();
```

## 1.3 Add an “author” property to each video in the JSON and display that next to the title

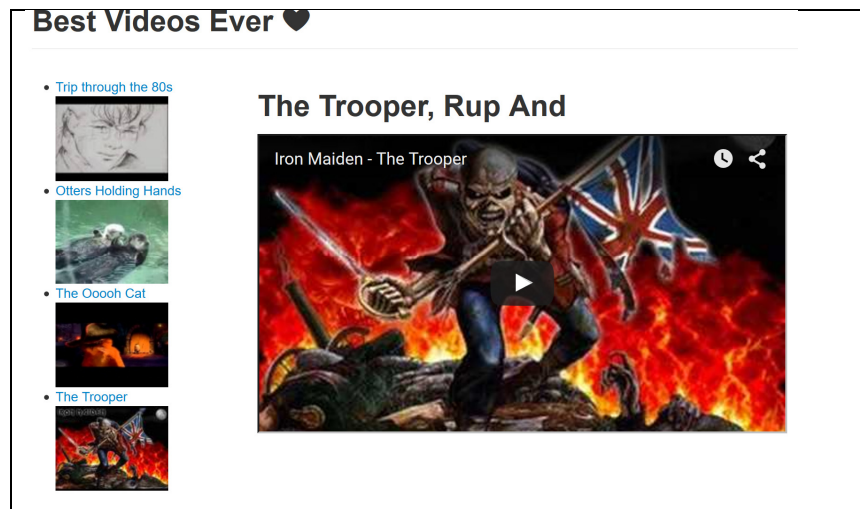
In order to hold an author property the JSON code is changed:

```
var videos = [
  { "youtubeId": "TddFnTB_7IM", "title": "Trip through the 80s", "author": "Anders And" },
  { "youtubeId": "epUk3T2Kfno", "title": "Otters Holding Hands", "author": "Rip And" },
  { "youtubeId": "il2IrgFHfsg", "title": "The Ooooh Cat", "author": "Rap And" },
  { "youtubeId": "2G5rfPISiwo", "title": "The Trooper", "author": "Rup And" }
];
```

And the script code is changed to:

```
var videoWatcher = $('#video-watcher');
videoWatcher.hide();
videoWatcher.html("<H2>" + video.title + ", " + video.author + "</H2>")
videoWatcher.append(videoEmbed);
videoWatcher.fadeIn();
```

Title and author is now displayed above the video player:



## Exercise 2

### 2.1 Add Movie Trailers

The video list is replaced by the following movie trailers:

```
var videos = [
  {"youtubeId": "_comGBmnYew", "title": "The Patriot"},
  {"youtubeId": "2TWYDogv4WQ", "title": "Cast Away"},
  {"youtubeId": "eUdM9vrCbow", "title": "Django Unchained"},
  {"youtubeId": "jNuIn4T-CLk", "title": "From Dusk Till Dawn"}
];
```

### 2.2 Add Ratings

Using the OMDb api (<http://www.omdbapi.com/>) we can add additional information about the current video. Below we will add the IMDB rating and the Rotten Tomatoes Rating. Also, we will use the title from the OMDb service.

First we modify the page layout to display title and other OMDb information:

```
<div class="span9" id="video-watcher"> </div>
```

Is changed to:

```
<div class="span9">
  <div id="video-title"></div>
  <div id="video-watcher"></div>
  <div id="omdbInfo"></div>
  <br/><br/>
</div>
```

In the videoLink click function we change the code to invoke the OMDb API and add the data to the page. Below, in red, we can see the extra code inserted.

```
videoLink.on('click', function(e) {
    e.preventDefault();

    var videoEmbed = $('<iframe></iframe>');
    videoEmbed.attr('src', youtube.generateEmbedUrl(video.youtubeId));
    videoEmbed.attr('width', 560);
    videoEmbed.attr('height', 315);

    var videoWatcher = $('#video-watcher');
    videoWatcher.hide();
    videoWatcher.html(videoEmbed);
    videoWatcher.append();
    videoWatcher.fadeIn();

    var titleInfo = $('#video-title');
    var movieInfo = $('#omdbInfo');

    $.getJSON('http://www.omdbapi.com/?t=' + video.title + '&tomatoes=true', '&callback=?' ,
        function(data) {
            titleInfo.html("<H2>" + data.Title + "</H2>");
            titleInfo.append("<br/>Director: " + data.Director);
            titleInfo.append("<br/>Writer: " + data.Writer);
            titleInfo.append("<br/>Actor(s): " + data.actors)
            titleInfo.append("<br/><br/>");


            movieInfo.html("IMDB Rating: " + data.imdbRating + ", Rotten Tomatoes Rating: " +
data.tomatoRating);
            movieInfo.append("<br/>");
            movieInfo.append("<br/>Genre: " + data.Genre);
            movieInfo.append("<br/>Plot: " + data.Plot);
            movieInfo.append();
        }
    );
});
```

Our page now shows the following:

- [The Patriot](#)
- [Cast Away](#)
- [Django Unchained](#)
- [From Dusk Till Dawn](#)

## Cast Away

Director: Robert Zemeckis  
Writer: William Broyles Jr.  
Actor(s): Paul Sanchez, Lari White, Leonid Citer, David Allen Brooks



IMDB Rating: 7.7, Rotten Tomatoes Rating: 7.4

Genre: Adventure, Drama  
Plot: A FedEx executive must transform himself physically and emotionally to survive a crash landing on a deserted island.

## 2.3 Personalize site

As you can see in 2.2 the plot is a very short text. Changing the API call we can retrieve the full plot:





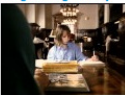

```
$.getJSON('http://www.omdbapi.com/?t=' + video.title + '&tomatoes=true&plot=full',
  '&callback=?' ,
  function(data) {
    titleInfo.html("<H2>" + data.Title + "</H2>");
    titleInfo.append("<br/><b>Director:</b>" + data.Director);
    titleInfo.append("<br/><b>Writer:</b>" + data.Writer);
    titleInfo.append("<br/><b>Actor(s):</b>" + data.actors)
    titleInfo.append("<br/><br/>");

    movieInfo.html("<b>IMDB Rating:</b>" + data.imdbRating + ", <b>Rotten Tomatoes Rating:</b>"
  + data.tomatoRating);
    movieInfo.append("<br/>");
    movieInfo.append("<br/><b>Genre:</b>" + data.Genre);
    movieInfo.append("<br/><b>Plot:</b>" + data.Plot);
    movieInfo.append();
  }
);
```

And the legend is now displayed in bold, making the text easier to read. Also, to further improve readability of the page, a break is added between each video in the list.


```
videoLink.append(video.title);
var thumbnailUrl = youtube.generateThumbnailUrl(video.youtubeId);
var thumbnailImg = $('<img>');
thumbnailImg.attr('src', thumbnailUrl);
videoLink.append(thumbnailImg);
videoLink.append('<br/>');
```

Which, with a few more videos added, gives us:

- [The Patriot](#)  

- [Cast Away](#)  

- [Django Unchained](#)  

- [From Dusk Till Dawn](#)  

- [Regarding Henry](#)  

- [The Last Of The Mohicans](#)  


## The Last of the Mohicans

**Director:** Michael Mann  
**Writer:** James Fenimore Cooper (novel), John L. Balderston (adaptation), Paul Perez (adaptation), Daniel Moore (adaptation), Philip Dunne, Michael Mann (screenplay), Christopher Crowe (screenplay)  
**Actor(s):** Daniel Day-Lewis, Madeleine Stowe, Russell Means, Eric Schweig



**IMDB Rating:** 7.8, **Rotten Tomatoes Rating:** 7.9

**Genre:** Action, Adventure, Drama  
**Plot:** British and French troops do battle in colonial America, with aid from various native American war parties. The British troops enlist the help of local colonial militia men, who are reluctant to leave their homes undefended. A budding romance between a British officer's daughter and an independent man who was reared as a Mohican complicates things for the British officer, as the adopted Mohican pursues his own agenda despite the wrath of different people on both sides of the conflict.

### Exercise 3

We can use the OMDb API to add even more information about the selected movie to the database. Let us say we want to display the video runtime in the format hh:mm. The API returns the runtime in minutes so we need to do some conversion.

First we add a function that can parse a minutes number into the correct format:

```
function parseMinutes(n) {
    var hours = Math.floor(n/60);
    var minutes = n%60;
    var time = (hours + ":") + (minutes < 10 ? "0" + minutes : minutes);
    return time;
}
```

Then we add the information to the page:

```
movieInfo.html("<b>IMDB Rating:</b> " + data.imdbRating + ", <b>Rotten Tomatoes Rating:</b> "
+ data.tomatoRating);
movieInfo.append("<br/>");
movieInfo.append("<br/><b>Runtime: </b>" + parseMinutes(parseInt(data.Runtime)));
movieInfo.append("<br/><b>Genre:</b> " + data.Genre);
movieInfo.append("<br/><b>Plot:</b> " + data.Plot);
movieInfo.append();
```

Since the API actually returns the runtime as “xxx min” we have to first parse that string into an int using the parseInt function.

The complete json\_starter.html now looks like this:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <!--<meta charset="utf-8"> -->
    <title>Video Watcher</title>
    <link rel="stylesheet" type="text/css" href="json_starter-
filer/bootstrap.css">

    <style type="text/css">
      /* This puts titles always on a separate line from thumbnails */
      #videos-list img {
        display: block;
      }
    </style>
  </head>

  <body>

    <div class="container">
      <div class="row">
        <div class="span18">
          <h2 class="page-header">Best Videos Ever ♥</h2>
        </div>
      </div>

      <div class="row">
        <div class="span3">
          <ul id="videos-list"></ul>
        </div>
        <div class="span9">
          <div id="video-title"></div>
          <div id="video-watcher"></div>
          <div id="omdbInfo"></div>
          <br/><br/>
        </div>
      </div>
    </div>
```



```

</div>

<script type="text/javascript">
    function parseMinutes(n) {
        var hours = Math.floor(n/60);
        var minutes = n%60;
        var time = (hours + ":") + (minutes < 10 ? "0" + minutes : minutes);
        return time;
    }
</script>
<script type="text/javascript" src="json_starter-filer/youtube.js"></script>
<script type="text/javascript" src="json_starter-filer/jquery.js"></script>
<script>

    var videos = [
        {"youtubeId": "_comGBmnYew", "title": "The Patriot"},
        {"youtubeId": "2TWYDogv4WQ", "title": "Cast Away"},
        {"youtubeId": "eUdM9vrCbow", "title": "Django Unchained"},
        {"youtubeId": "jNuIn4T-CLk", "title": "From Dusk Till Dawn"},
        {"youtubeId": "PtipW0DoQB", "title": "Regarding Henry"},
        /* {"youtubeId": "uwTKRz-WmHU", "title": "Gladiator"}, */
        /* {"youtubeId": "JmjXiAqyEfo", "title": "A Good Year"}, */
        /* {"youtubeId": "bZJdhq0_Moo", "title": "Dances With Wolves"}, */
        {"youtubeId": "yaQeVnN6pUc", "title": "The Last Of The Mohicans"}
    ];

    var addVideoToList = function(video) {
        var videoLink = $('<a>');
        videoLink.append(video.title);
        var thumbnailUrl = youtube.generateThumbnailUrl(video.youtubeId);
        var thumbnailImg = $('<img>');
        thumbnailImg.attr('src', thumbnailUrl);
        videoLink.append(thumbnailImg);
        videoLink.append('<br/>');

        videoLink.on('click', function(e) {
            e.preventDefault();

            var videoEmbed = $('<iframe></iframe>');
            videoEmbed.attr('src', youtube.generateEmbedUrl(video.youtubeId));
            videoEmbed.attr('width', 560);
            videoEmbed.attr('height', 315);

            var videoWatcher = $('#video-watcher');
            videoWatcher.hide();
            videoWatcher.html(videoEmbed);
            videoWatcher.append();
            videoWatcher.fadeIn();

            var titleInfo = $('#video-title');
            var movieInfo = $('#omdbInfo');

            $.getJSON('http://www.omdbapi.com/?t=' + video.title +
                '&tomatoes=true&plot=full', '&callback=?',
                function(data){
                    titleInfo.html("<H2>" + data.Title + "</H2>");
                    titleInfo.append("<br/><b>Director:</b> " +
data.Director);

                    titleInfo.append("<br/><b>Writer:</b> " + data.Writer);
                    titleInfo.append("<br/><b>Actor(s):</b> " + data.actors);
                    titleInfo.append("<br/><br/>");

                    movieInfo.html("<b>IMDB Rating:</b> " + data.imdbRating +
", <b>Rotten Tomatoes Rating:</b> " + data.tomatoRating);
                    movieInfo.append("<br/>");
                    movieInfo.append("<br/><b>Runtime: </b>" +
parseMinutes(parseInt(data.Runtime)));
                    movieInfo.append("<br/><b>Genre:</b> " + data.Genre);
                    movieInfo.append("<br/><b>Plot:</b> " + data.Plot);
                    movieInfo.append();
                }
            );
        });
    };
</script>

```

```
        var videoItem = $('<li>');
        videoItem.append(videoLink);
        $('#videos-list').append(videoItem);
    }

    for (var i = 0; i < videos.length; i++) {
        addVideoToList(videos[i]);
    }
</script>
</body>
</html>
```