**Task 1**

The value of the attribute *data-video-id* in the *video* tag was replaced by the reference ID provided by Brightcove:  
data-video-id="5214098325001" => data-video-id="ref:skillassessment"

An id attribute has been added to the video tag with the value “MyPlayer”:  
<video id="MyPlayer"

An onclick event has been added to each button in order to execute a JavaScript function when a button is clicked:

<button id="playbtn" onclick="playVideo()">PLAY</button>

<button id="pausebtn" onclick="pauseVideo()">PAUSE</button>

<button id="mutebtn" onclick="muteVideo()">MUTE</button>

An extra button has been added for unmute the video:

<button id="unmutebtn" onclick="unmuteVideo()">UNMUTE</button>

Before executing the play(), pause() and muted() methods we have to make sure the video is loaded and ready to play. The video was previously loaded by the ready() method included in the task code

We will reference the player with the variable myPlayer which is a reference to the video object. At this point we can play, pause and mute the video.

function playVideo() {

myPlayer.play();

}

function pauseVideo() {

myPlayer.pause();

}

function muteVideo() {

myPlayer.muted(true);

}

function unmuteVideo() {

myPlayer.muted(false);

}

In order to get the long description and thumbnail url of a video the getVideo() method (from the catalog library of methods) can be used.

The execution of the getVideo() method will produce an event called catalog\_request which contains all the data of our video.

myPlayer.catalog.getVideo('ref:skillassessment', function(error, video){  
/\*console.log('error',error);\*/  
});

myPlayer.on('catalog\_response',function(evt){  
 var api\_response = JSON.parse(evt.response.response);  
 console.log(api\_response.long\_description);  
 console.log(api\_response.thumbnail);  
}  
});

Task 2

When a video is embed in a page it doesn’t resize with the browser window.

That’s because the video player is given a fixed width and height.

In this task the intrinsic ratio  CSS technique is being used to fluidly constrain a child element to a ratio set in their parent element.

First thing I did was identify in the code the child (.BCLvideoWrapper) and parent (#BCLcontainingBlock) elements

   <div id="BCLcontainingBlock">

    <div class="BCLvideoWrapper">

Then I identified their class and id selectors in the CSS code and I cleaned the code of irrelevant classes and id’s for this example

 #BCLcontainingBlock {} (bounding container)

.BCLvideoWrapper {} (parent container with an intrinsic ratio)

These containers have to be declared around the video player but they are not and there is a </div> tag missing

1) The player needs to be wrapped by 2 divs, 1 </div> is missing and in the wrong place

The resize function

       $BCLcontainingBlock.width($BCLbodyContent.width() \* newWidth);

-->container around the video player a