JS 16 MP3 Player Cont.

Real estate has **A LOT** to do with web development. Customers need a lot of information because more information helps to understand what is in front of them. Unfortunately, web developers only have a single page to display that information.

How can we give customers all the information that they need and doing so with only a single web page? The answer is to use real estate. The solution is to go up by stacking information on top of each other and making information show up **ONLY** when the user wants it.

In this session, we create a **pop up banner** that **ONLY** shows up when the user hovers the mouse over an event block. **We have used the *hover* before and we did so when writing CSS ➔** `p#p1:hover`.

However, this session will focus on *hover in JS*, which is very different.

1. Add event listener 2. Absolute positioning 3. Classes and objects

==

0. Introduction

If we put extra information onto the website, it will make the website look busy and loaded with too much information. So, we will instead put the extra information inside a pop up banner. Next, we will hide the pop up banner.

We only display the extra information when the user asks for it. For this project, the user "**asks**" for something by **hovering** the mouse cursor over a song.

Until then, the pop up banner is hidden. This is good because it turns our website into an album cover that is similar to a table of content. If the user wants more information, they simply ask for it by "**hovering**" their mouse cursor over a song and the pop up banner will appear.

Continue to the next page

1. <u>DO NOT</u> WRITE THE CODE BELOW IN `<script> </script>.` Instead, look at the code in BLUE and change it to have your favorite music. The explanation will show you how to customize the extra information.

```
function init_system()
{
      alert( "inside init_system()" );

      /* ==================
      bName, sTitle, sCaption, sGenre, sFN



      */
      var song_0 = new SongProfile
      (
      "Magic!",
      "Rude",
      "#1",
      "Pop",
https://vuongducnguyen.com/audio/Magic!_Rude_(Violin_Cover_by_Robert_Mendoza).mp3
      );

      playlist.push( song_0 );

      var song_1 = new SongProfile
      (
      "BSB",
      "As Long As You Love Me",
      "#1",
      "Pop",
https://vuongducnguyen.com/audio/Magic!_Rude_(Violin_Cover_by_Robert_Mendoza).mp3
      );

      playlist.push( song_1 );

      var song_2 = new SongProfile
      (
      "Jackson 5",
      "I Want You Back",
      "#1",
      "Pop",
https://vuongducnguyen.com/audio/Magic!_Rude_(Violin_Cover_by_Robert_Mendoza).mp3
      );

      playlist.push( song_2 );

      var song_3 = new SongProfile
      (
      "1D",
      "What's Makes You Beautfiul",
      "#1",
      "Pop",
https://vuongducnguyen.com/audio/Magic!_Rude_(Violin_Cover_by_Robert_Mendoza).mp3
      );

      playlist.push( song_3 );
```

```
        //song_1.update_setting(   );

        // -- make clone of table rows

         var i = 0;
        for( ; i < playlist.length; i++ )
        {
         make_a_clone( i, playlist[i] );

         }
        alert( "done making clone" );
        trackNum = 0;

        change_songs( 0, playlist );

        // -- extra info pop up
        song_0.set_listener( 0, "Good Song" );
        song_1.set_listener( 1, "Best BB of all time!!!" );
        song_2.set_listener( 2, "Good beat" );
        song_3.set_listener( 3, "2 Directions");
        extraInfoDom = document.getElementById( "extraInfo" );

        // -- hide the base part
        document.getElementById( pUpDOM ).style.display = "none";


}
```

Explanation

The code in **green** will go inside the pop up banner and serve as extra information. The pop up banner will be stacked on top of the website. You can change the code in **green** to be anything you like.

Continue to the next page

2. Write The HTML Code In Between `<body> </body>`. **ONLY** write the code in **RED and BLACK** and position it under the code in **Purple**

```
<button id = "minusOne" onclick = "dec()"> Back
</button>

<button id = "plusOne" onclick = "incr()"> Next
</button>



<div
id = "extraInfo"
>

</div>
```

Explanation

This is our pop up banner. Notice that we are using a `<div> </div>` to create our pop up banner and we give it an id of “**extraInfo**”. Right now, the pop up banner is empty but we will put data into it later on.

Remember that we want to create an effect where the extra information looks like it is stacked on top of the website. The CSS and JS code to create this effect will be the next couple of chapters.

3. Write The CSS Code In Between <style> </style>. Style it any way you like.

```
#extraInfo
{
       position  : absolute;

      padding   : 0px 0px 0px 0px;
      margin    : 0px 0px 0px 0px;

      width      : 250px;
      height     : 150px;

      background-color: #00ffbb;
      box-shadow: 7px 5px 2px #f0f0f0;
      border    : 1px solid #a0a0a0;

      text-shadow: 0px 1px 1px #a0a0a0;
      font-weight: bold;
      color      : #ffa500;
      font-family: Arial, Tahoma, Serif;

}
```

Continue to the next page

Explanation

Here we are styling our pop up banner. Remember that CSS is different from HTML. In CSS, we replace the word "**id**" with the hash tag ➔ `#extraInfo`

Look at the code in **green**. We are making the pop up banner large because we might want to add more information inside of it.

4. Write The JS Code In Between `<script> </script>`. Write ALL OF IT!!! The code below is a private function of SongProfile … So, where do we put it? Color code is for explanation.

```
set_listener( divNum, eInfo )
{
       console.log( "id: " + pUpDOM + divNum );
       this.objId = divNum;

       this.extraInfo = eInfo;

       pUpObj = document.getElementById( pUpDOM + divNum );
       //pUpObj = document.getElementById( pUpDOM );
       pUpObj.addEventListener( "mouseover", this.show_pop_up.bind(this) );

       pUpObj = document.getElementById( pUpDOM );
       pUpObj.addEventListener( "mouseleave", this.hide_pop_up.bind(this) );

}
```

Explanation

We created our pop up banner using a `<div> </div>` and then styled it using **CSS**. **By default, the pop up banner cannot detect anything**. So, we must write code to give our pop up banner the **ability to detect** if the user has positioned the mouse cursor over a song.

We give the ability by setting up the event listener. Once the event listener has been set up, an event listener is **ALWAYS** active. When the event occurs, an action (ie. a JS function) will be called.

In the code above, we are setting two event listeners. The first one is the code in **red** and the second is the code in **green**.

In the code in **red**, the event is "**mouseover**". ***This means that the user has put the mouse cursor over an HTML element***. When this happens, the action is "`this.show_pop_up`".

1. The action is a **private function** that belongs to the **class SongProfile**.
2. We know that it is a **private function** because of the word "`this.`" that can be seen in front of the function name.

In the code in green, the event is "**mouseleave**". This means that the user has moved the mouse cursor away from an HTML element. When this happen, the action is `"this.hide_pop_up"`

In our situation, the HTML element is the horizontal row that displays the songs of our MP3 player.

1. If the user puts **the mouse cursor over a song**, the action is `"this.show_pop_up"` and the pop up banner becomes visible
2. If the user **moves the mouse cursor away**, the action is `"this.hide_pop_up"` and the pop up banner becomes hidden

The actions will be **defined** below.

5. Write The JS Code In Between `<script> </script>`. Write ALL OF IT!!! The code below is a private function of SongProfile. So, where do we put it? Color code is for explanation.

```
show_pop_up()
{
       console.log( "id: " + pUpDOM + this.objId );

       //alert( "inside show_pop_up()" );
       extraInfoDom.innerHTML     = this.extraInfo;
       extraInfoDom.style.border = "1px solid #a0a0a0";
       extraInfoDom.style.left    = "395px";
       // extraInfoDom.style.top     = parseInt(200) * parseInt(this.objId) + "px";
       extraInfoDom.style.top     = document.getElementById( pUpDOM + this.objId ).offsetTop + 160 + "px";

      console.log( "top: " + document.getElementById( pUpDOM + this.objId ).offsetTop );

       console.log( "" );
       //extraInfoDom.style.top     = document.getElementById( pUpDOM + this.objId ).style.top;

       console.log( "extraInfoDom.style.left:" + extraInfoDom.style.left);
       console.log( "extraInfoDom.style.top:" + extraInfoDom.style.top);
       extraInfoDom.style.display= "block";

}

hide_pop_up()
{
       //alert( "inside show_pop_up()" );
       extraInfoDom.innerHTML     = " ";
       extraInfoDom.style.border = "5px solid #ffd700";

       extraInfoDom.style.display= "none";

}
```

Continue to the next page for Explanation

Explanation

The code in **blue** updates the banner using **.innerHTML** to have information that we want the user to see. This can be any information we want.

The code in **gold** is the code that positions the pop up banner next to the mouse cursor. The goal is to position the banner right next to the song that is selected.

In our case, putting the mouse cursor over a row (ie. hovering over a song) indicates the song selected. This will cause the pop up banner to be right next to the mouse cursor.

1. This way, the banner matches the song that the user has hovered the mouse over.

The code in grey and black will look familiar. If you go back to JS 9 and 10, we use the `.style.display` to make an HTML element visible or invisible. Remember that

`.style.display = “none” makes an HTML element invisible`

`.style.display = “block” makes an HTML element visible.`

In our code, **when we put the mouse cursor over a song**, the pop up banner becomes visible.

However, **when we move the mouse cursor away**, the pop up banner becomes hidden.

CSS Challenge

1. Pick 3 colors
2. Style your MP3 player using the 3 colors that you selected

JS Challenge

Inside the private function `this.show_pop_up`, check the genre of the song. **Remember that genre is a private attribute**

a. If the genre of the song is “**Pop**”, change the backgroundColor of the pop up banner to “`#ffd700`”
b. If the genre of the song is “**EDM**”, change the backgroundColor of the pop up banner to “`#00BFFF`”
c. Else, change the backgroundColor of the pop up banner to “`#FFA500`”