座號：19 姓名：柯昱廷

階段性作業三

JavaScript綜合應用

**請利用JavaScript撰寫一音樂播放器，須包含以下功能：**

1. 須有可按鈕執行播放、停止與暫停的功能。
2. 須有可按鈕執行上一首的下一首的功能。
3. 須有直接選擇曲目的功能。
4. 須有可按鈕執行快轉及倒轉的功能。
5. 須有可按鈕執行單曲循環、全部循環及隨機選曲播放的功能。
6. 須有音量控制的功能。
7. 須有資訊看版可顯示目前播放狀態及目前播放曲目的功能。
8. 須有播放進度條，並可由進度條調整曲目快轉或倒轉的功能。
9. 須有我的歌本功能，可供編輯想聽的歌曲。
10. 介面不限，請自行設計。

直接選曲功能



音量控制

歌曲播放長度

進度條

各功能按鈕

資訊看版

我的歌本

**繳交作業時，請以FTP上傳程式原始檔與本文件檔各一份，原始程式碼請另放置在「完成結果檔」資料夾內。**

※以上各項要求皆須完成始得通過※

原始碼請貼於此處下方：

1.37HW3.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

<link href="37HW3.css" rel="stylesheet" />

</head>

<body>

<div id="ControlPanel">

<select id="song">

</select>

<hr />

<span id="btnPlay" onclick="Play()" title="這是播放鈕">4</span>

<span onclick="Stop()"><</span>

<span onclick="changeTime(-5)">7</span> <!--倒轉-->

<span onclick="changeTime(5)">8</span> <!--快轉-->

<span onclick="changeSong(-1)">9</span> <!--上一首-->

<span onclick="changeSong(1)">:</span> <!--下一首-->

<span onclick="setMuted()">V</span>

<span onclick="setLoop()">q</span>

<span onclick="setRandom()">s</span>

<span onclick="setAllLoop()">`</span>

<span onclick="hide()">@</span>

<hr />

<input type="range" id="progress" step="0.001" min="0" max="1" value="0" oninput="setTimeBar()" />

<hr />

<input oninput="RangeVolume()" id="rangeVolume" value=0.5 type="range" max=1 min=0 step=0.01 /> <!--音量調整條-->

<input onclick="btnVolume(-0.01)" id="btnVolumeUp" type="button" value="-" />

<input onclick="btnVolume(0.01)" id="btnVolumeDown" type="button" value="+" />

<input type="text" id="textVolume" value=0.5 readonly /> <!--音量顯示-->

<hr />

<div id="info">

<marquee>請按下播放鈕</marquee>

<div>00:00 / 00:00</div>

<div></div>

</div>

<div id="book" class="hide">

<!--全部歌曲-->

<div ondrop="drop(event)" ondragover="allowDrop(event)">

<div title="music/elysees.mp3">Champs-Elyees</div>

<div title="music/airball.mp3">告白氣球</div>

<div title="music/travel.mp3">帶你去旅行</div>

<div title="music/moon.mp3">月亮</div>

<div title="music/nothing.mp3">nothing</div>

</div>

<!--我的歌本-->

<div ondrop="drop(event)" ondragover="allowDrop(event)">

</div>

<button onclick="UpdateMusic()">更新歌單</button>

</div>

</div>

<audio id="music">

<source src="music/elysees.mp3" type="audio/mpeg" title="Champs-Elyees" />

</audio>

<script src="37HW3.js"></script>

</body>

</html>

2.37HW3.css:

@font-face {

font-family: 'webdings';

src: url('fonts/webdings.eot') format('embedded-opentype');

}

@font-face {

font-family: 'webdings';

src: url('fonts/webdings.svg') format('svg');

}

@font-face {

font-family: 'webdings';

src: url('fonts/webdings.ttf') format('truetype');

}

@font-face {

font-family: 'webdings';

src: url('fonts/webdings.woff') format('woff');

}

@font-face {

font-family: 'webdings';

src: url('fonts/webdings.woff2') format('woff2');

}

#ControlPanel{

margin:auto;

width:600px;

text-align:center;

background-color:#a14b0a;

}

#ControlPanel > span {

border-radius:15%;

font-family: Webdings;

font-size: 3em;

border: 1px solid black;

box-shadow: 1px 1px 1px 1px;

}

#rangeVolume {

width: 494px;

}

#progress {

width: 600px;

}

#textVolume{

width:30px;

}

#book > div {

float: left;

border: 1px groove;

width: 298px;

height: 200px;

}

#book > div > div {

border: 1px groove;

background-color: silver;

}

#book > button {

}

#song, #info {

border-radius: 5px;

width: inherit;

font-size: 1.5rem;

background-color: #fcb388;

color: #a14b0a;

text-align: center;

}

.hide {

display: none;

}

3.37HW3.js:

let audio = document.getElementById("music");

let btnPlay = document.getElementById("btnPlay");

let btnRange = document.getElementById("btnRange");

let divTest = document.getElementById("divTest");

let rangeVolume = document.getElementById("rangeVolume");

let textVolume = document.getElementById("textVolume");

let info = document.getElementById("info");

let song = document.getElementById("song");

let progress = document.getElementById("progress");

let book = document.getElementById("book");

song.addEventListener('change', function () {

changeMusic(song.options.selectedIndex);

})

function hide() { //顯示歌單

book.className = book.className === "" ? "hide" : "";

}

function UpdateMusic() { //更新歌單

let option = Object;

for (j = song.children.length - 1; j >= 0; j--) {

song.removeChild(song.children[j]);

}

for (let i = 0; i < book.children[1].children.length; i++) {

option = document.createElement("option");

option.value = book.children[1].children[i].title;

option.innerText = book.children[1].children[i].innerText;

song.appendChild(option);

}

changeSong(0);

}

function allowDrop(ev) { //停在某物件上面時

ev.preventDefault();//停止物件預設行為

}

function drag(ev) { //拉起來

ev.dataTransfer.setData("text", ev.target.id);

}

function drop(ev) { //放下來

ev.preventDefault();

var data = ev.dataTransfer.getData("text");

if (ev.target.id === "") ev.target.appendChild(document.getElementById(data));

else ev.target.parentNode.appendChild(document.getElementById(data));

}

//我的歌本(把歌本Load進去Select中)

//移動物件部分可以參考==>W3C Drag and Drop API

function load() {

let option = Object;

for (let i = 0; i < book.children[0].children.length; i++) {

book.children[0].children[i].addEventListener('dragstart', function () {

drag(event);

})

book.children[0].children[i].id = "song" + (i + 1);

book.children[0].children[i].draggable = "true";

option = document.createElement("option");

option.value = book.children[0].children[i].title;

option.innerText = book.children[0].children[i].innerText;

song.appendChild(option);

}

}

load();

//全曲循環

function setAllLoop() {

info.children[2].innerHTML === "" ? info.children[2].innerHTML = "全曲循環" : info.children[2].innerHTML = "";

}

//隨機播放

function setRandom() {

//第一種

info.children[2].innerHTML === "" ? info.children[2].innerHTML = "隨機播放" : info.children[2].innerHTML = "";

}

//單曲循環

function setLoop() {

info.children[2].innerHTML === "" ? info.children[2].innerHTML = "單曲循環" : info.children[2].innerHTML = "";

}

//靜音

function setMuted() {

audio.muted = !audio.muted;

//能不要用判斷式就不要用，盡量使用傳參數的方法

}

//\*進度條事件

function setTimeBar() {

audio.currentTime = progress.value \* audio.duration;

}

//\*歌曲切換

function changeMusic(i) {

if (i < 0) i = song.options.length - 1;

else if (i === song.options.length) i = 0;

song.options[i].selected = true; //按下一首後，調整select

audio.children[0].src = song.options[i].value;

audio.children[0].title = song.options[i].innerText;

audio.load();

if (btnPlay.innerText === ";") Play();

}

//\*上一首與下一首

function changeSong(n) {

let index = song.options.selectedIndex + n;

changeMusic(index);

}

//時間格式轉換

function getTimeFormat(timeSec) {

let min = 0; let sec = 0;

min = parseInt(timeSec / 60);

sec = parseInt(timeSec % 60);

min = min < 10 ? "0" + min : min;

sec = sec < 10 ? "0" + sec : sec;

return min + ":" + sec;

}

//取得歌曲播放時間

function getDuration() {

progress.value = audio.currentTime / audio.duration;

info.children[1].innerText = getTimeFormat(audio.currentTime) + " / " + getTimeFormat(audio.duration);

//console.log("audio.currentTime=" + audio.currentTime, "audio.duration=" + audio.duration)

setTimeout(getDuration, 100);

//判斷狀態

if (audio.currentTime === audio.duration) {

switch (info.children[2].innerHTML) {

case "單曲循環":

console.log("單曲循環");

changeSong(0);

break;

case "隨機播放":

console.log("隨機播放");

let n = Math.floor(Math.random() \* song.options.length); //亂數(0~1，不包含1)\*歌曲數目

console.log(Math.floor(n, Math.random(), song.options.length));

changeMusic(n);

break;

case "全曲循環":

if (song.selectedIndex === song.options.length - 1) {

console.log("全曲循環");

changeMusic(0);

}

else changeSong(1);

break;

case "": //如果歌曲播完而且是最後一首，又沒有狀態的話

changeSong(1);

break;

}

}

}

//播放與暫停

function Play() {

if (audio.paused) {

audio.play();

btnPlay.innerText = ";";

info.children[0].innerText = "目前播放：" + audio.children[0].title;

getDuration();

}

else {

audio.pause();

btnPlay.innerText = "4";

info.children[0].innerText = "目前暫停中";

}

}

//停止播放

function Stop() {

audio.pause();

audio.currentTime = 0;

info.children[0].innerText = "目前停止中";

}

//快轉與倒轉

function changeTime(t) {

audio.currentTime += t;

}

//音量調整按鈕

function btnVolume(n) {

rangeVolume.value = Number(rangeVolume.value) + n;

RangeVolume();

}

//音量調整條

function RangeVolume() {

audio.volume = rangeVolume.value;

textVolume.value = audio.volume;

}

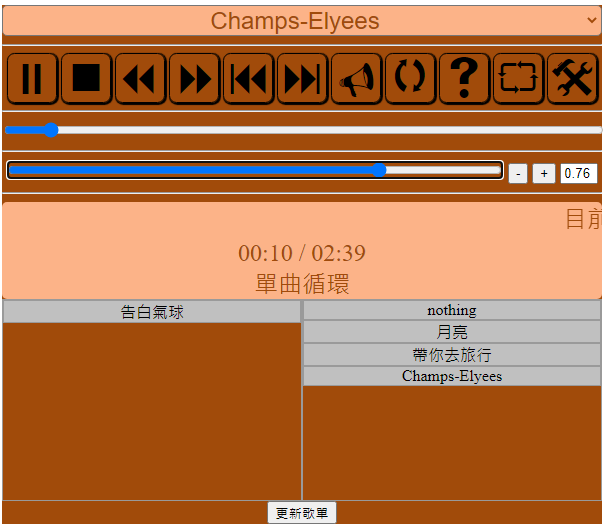
}

畫面截圖請貼於此處下方(請截取執行不同功能時的圖至少3張)：

1.更新歌單:

2.隨機播放:



3.調整音量