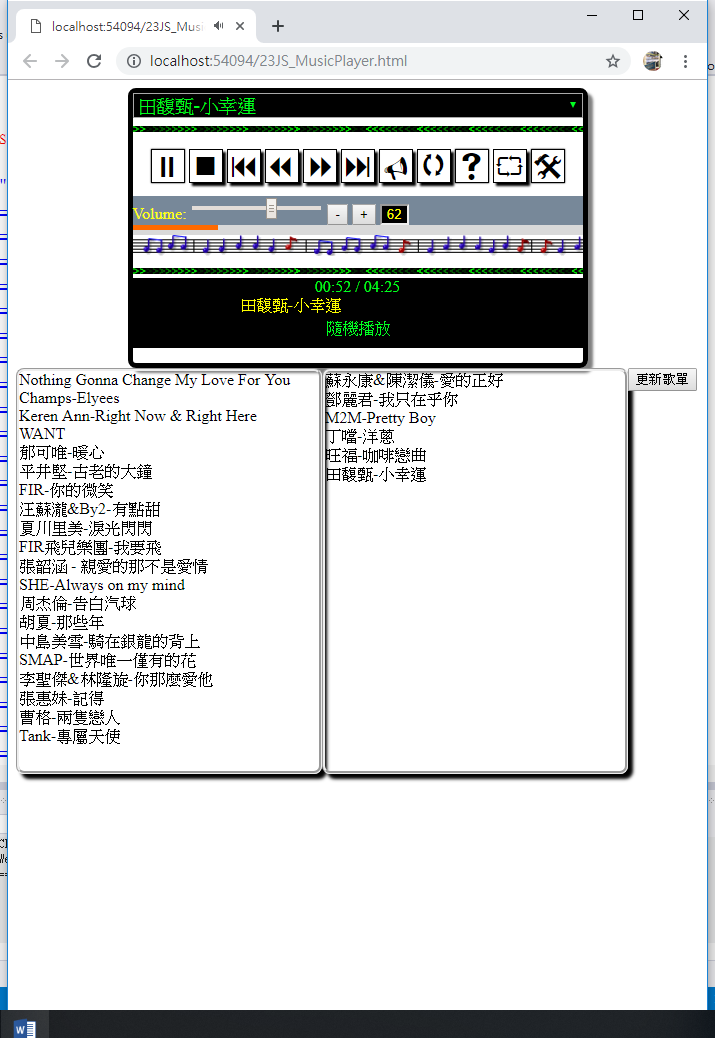
座號：01 姓名：林秋輝

階段性作業四

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. 須有顯示目前曲目總長度與已播放長度的功能。
9. 須有可按鈕執行快轉及倒轉的功能。
10. 須有我的歌本功能，可供編輯想聽的歌曲。
11. 介面不限，請自行設計。



歌曲播放長度

我的歌本

資訊看版

音量控制

各功能按鈕

直接選曲功能

**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：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>21JS\_Player</title>

<link href="PlayerStyle.css" rel="stylesheet" />

</head>

<body>

<audio id="audio">

<source id="song" type="audio/mpeg" />

<!--<source src="music/nothing.mp3" type="audio/mpeg" />-->

</audio>

<div id="ControlPanel" style="width:600px;">

<!--歌曲播放清單-->

<select id="music"></select>

<div id="FuncBtn">

<span id="play">4</span>

<!--<span id="pause">;</span>-->

<span id="stop"><</span>

<span id="prevsong">9</span>

<span id="prevtime">7</span>

<span id="nexttime">8</span>

<span id="nextsong">:</span>

<span id="muted">V</span>

<span id="loop">q</span>

<span id="random">s</span>

<span id="allloop">`</span>

<span id="songbook">@</span>

</div>

<div id="volume">

<input id="vol" type="range" min="0" max="100" value="80" />

<input id="volM" type="button" value="-" />

<input id="volP" type="button" value="+" />

<input id="volValue" type="text" value="80" readonly="readonly" />

</div>

<div id="settime">

<div id="progress"></div>

</div>

<div id="InfoPanel">

<div id="duration">00:00 / 00:00</div>

<marquee id="info">請按撥放按鈕~~~~~</marquee>

<div id="info2"></div>

</div>

<hr />

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

<div id="Sbook">

<div title="music/忘憂大提琴.mp3">忘憂大提琴</div>

<div title="music/nothing.mp3">Nothing Gonna Change My Love For You</div>

<div title="music/以後別做朋友.mp3">以後別做朋友</div>

<div title="music/聽見下雨的聲音.mp3">聽見下雨的聲音</div>

<div title="music/tearshining.mp3">夏川里美-淚光閃閃</div>

<div title="music/sweety.mp3">汪蘇瀧&By2-有點甜</div>

<div title="music/careyou.mp3">鄧麗君-我只在乎你</div>

<div title="music/airball.mp3">周杰倫-告白汽球</div>

<div title="music/little.mp3">田馥甄-小幸運</div>

<div title="music/MV.mp3">旺福-咖啡戀曲</div>

<div title="music\moon.mp3">五月天-私奔到月球</div>

<div title="music\PartyAnimal.mp3">五月天-派對動物</div>

</div>

<div id="Tbook">

</div>

<div id="btnUpdateMusic">

更新歌單

</div>

</div>

</div>

<script src="Player.js"></script>

</body>

</html>

var audio = document.getElementById("audio");

var play = document.getElementById("play");

//var pause = document.getElementById("pause");

var stop = document.getElementById("stop");

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

var music = document.getElementById("music");

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

var prevtime = document.getElementById("prevtime");

var nexttime = document.getElementById("nexttime");

var prevsong = document.getElementById("prevsong");

var nextsong = document.getElementById("nextsong");

var vol = document.getElementById("vol");

var volM = document.getElementById("volM");

var volP = document.getElementById("volP");

var volValue = document.getElementById("volValue");

var muted = document.getElementById("muted");

var loop = document.getElementById("loop");

var random = document.getElementById("random");

var allloop = document.getElementById("allloop");

var info2 = document.getElementById("info2");

var duration = document.getElementById("duration");

var settime = document.getElementById("settime");

var Sbook = document.getElementById("Sbook");

var Tbook = document.getElementById("Tbook");

var btnUpdateMusic = document.getElementById("btnUpdateMusic");

var songbook = document.getElementById("songbook");

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

play.addEventListener("click", PlayMusic);

//pause.addEventListener("click", PauseMusic);

stop.addEventListener("click", StopMusic);

music.addEventListener("change", function () { ChangeMusic(music.selectedIndex); });

prevtime.addEventListener("click", function () { TimeChange(false); });

nexttime.addEventListener("click", function () { TimeChange(true); });

prevsong.addEventListener("click", function () { SongChange("p"); });

nextsong.addEventListener("click", function () { SongChange("n"); });

vol.addEventListener(BrowserTest(), function () { VolumeChange("c"); });

volM.addEventListener("click", function () { VolumeChange("m"); });

volP.addEventListener("click", function () { VolumeChange("p"); });

muted.addEventListener("click", SetMuted);

loop.addEventListener("click", loopSong);

random.addEventListener("click", randomSong);

allloop.addEventListener("click", allloopSong);

settime.addEventListener("click", function () { setTimeUseBar(event); });

Sbook.addEventListener("dragstart", function () { drag(event); });

Tbook.addEventListener("dragover", function () { allowDrop(event); });

Tbook.addEventListener("drop", function () { drop(event); });

Tbook.addEventListener("dragstart", function () { drag(event); });

Sbook.addEventListener("dragover", function () { allowDrop(event); });

Sbook.addEventListener("drop", function () { drop(event); });

btnUpdateMusic.addEventListener("click", UpdateMusic);

songbook.addEventListener("click", function () { displayBook(book.title) });

VolumeChange("c"); //設定初始音量

getDefaultSong(); //設定初始歌單

//瀏覽器偵測

function BrowserTest() {

if (navigator.userAgent.search("Chrome") != -1)

return "input";

else if (navigator.userAgent.search("Opera") != -1)

return "input";

else if (navigator.userAgent.search("Firefox") != -1)

return "input";

else

return "change";

}

function displayBook(displayFlag) {

if (displayFlag == "hide") {

book.title = "show";

}

else {

book.title = "hide";

}

book.className = book.title;

}

//更新歌單

function UpdateMusic() {

//清掉所有歌

for (i = music.children.length - 1; i >= 0; i--) {

music.removeChild(music.children[i]);

}

for (i = 0; i < Tbook.children.length; i++) {

var option = document.createElement("option");

var Snode = Tbook.children[i];

option.value = Snode.title;

option.innerText = Snode.innerText;

music.appendChild(option);

}

SongChange("a");

}

function drop(evt) {

evt.preventDefault();

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

evt.target.appendChild(document.getElementById(data));

}

function allowDrop(evt) {

evt.preventDefault();

}

function drag(evt) {

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

//alert(evt.target.id);

}

//抓預設歌曲

function getDefaultSong() {

for (i = 0; i < Sbook.children.length; i++) {

var option = document.createElement("option");

var Snode = Sbook.children[i];

option.value = Snode.title;

option.innerText = Snode.innerText;

music.appendChild(option);

Snode.draggable = true;

Snode.id = "song" + (i + 1);

}

SongChange("a");

}

function setTimeUseBar(evt) {

//console.log(ControlPanel.style.width);

audio.currentTime = audio.duration \* (evt.offsetX / parseInt(ControlPanel.style.width));

//console.log(ControlPanel.style.width);

}

function getDuration() {

var durationTime = getTimeFormat(audio.duration);

var currentTime = getTimeFormat(audio.currentTime);

duration.innerText = currentTime + " / " + durationTime;

progress.style.width = Math.floor(audio.currentTime / audio.duration \* 100) + "%";

console.log(progress.style.width);

setTimeout(getDuration, 1000);

if (durationTime == currentTime) {

if (info2.innerText == "單曲循環") {

SongChange("s");

}

else if (info2.innerText == "隨機播放") {

SongChange("r");

}

else if (music.selectedIndex == music.options.length - 1) {

if (info2.innerText == "全曲循環") {

SongChange("a");

}

else {

StopMusic();

}

}

else {

SongChange("c");

}

}

}

function getTimeFormat(TimeSec) {

var min = Math.floor(TimeSec / 60);

var sec = Math.floor(TimeSec) - min \* 60;

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

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

return min + ":" + sec;

}

//全曲循環

function allloopSong() {

if (info2.innerText == "全曲循環") {

info2.innerText = "";

}

else {

info2.innerText = "全曲循環";

}

}

//隨機播放

function randomSong() {

if (info2.innerText == "隨機播放") {

info2.innerText = "";

}

else {

info2.innerText = "隨機播放";

}

}

//單曲循環

function loopSong() {

if (info2.innerText == "單曲循環") {

info2.innerText = "";

}

else {

info2.innerText = "單曲循環";

}

}

function SetMuted() {

if (muted.innerText == "V") {

audio.muted = true;

muted.innerText = "U";

}

else {

audio.muted = false;

muted.innerText = "V";

}

}

function VolumeChange(sound) {

switch (sound) {

case "m":

vol.value--;

break;

case "p":

vol.value++;

break;

}

volValue.value = vol.value;

audio.volume = volValue.value / 100;

}

function SongChange(strFlag) {

var index = music.selectedIndex;

var length = music.options.length;

switch (strFlag) {

case "n":

if (index < length - 1)

index++;

else

index = 0;

break;

case "p":

if (index == 0)

index = length - 1;

else

index--;

break;

case "s":

audio.currentTime = 0;

break;

case "r":

index = Math.floor(Math.random() \* length);

break;

case "a":

index = 0;

break;

default:

index++;

break;

}

//song.src = music.options[index].value;

//music.options[index].selected = true;

ChangeMusic(index);

}

function TimeChange(boolFlag) {

if (boolFlag)

audio.currentTime += 10;

else

audio.currentTime -= 10;

}

function PlayMusic() {

audio.play();

getDuration();

ShowStatus(1);

if (play.innerText == "4")

play.innerText = ";";

else {

play.innerText = "4";

PauseMusic();

}

}

function PauseMusic() {

audio.pause();

ShowStatus(2);

}

function StopMusic() {

audio.pause();

audio.load();

play.innerText = "4";

ShowStatus(3);

}

function ChangeMusic(index) {

song.src = music.options[index].value;

song.title = music.options[index].innerText;

console.log(song.title);

music.options[index].selected = true;

audio.load();

if (play.innerText == ";") {

audio.play();

ShowStatus(1);

}

//PlayMusic();

}

function ShowStatus(status) {

var strStatus = "";

switch (status) {

case 1:

strStatus = "現正播放：" + song.title;

break;

case 2:

strStatus = "音樂暫停";

break;

case 3:

strStatus = "音樂停止";

break;

}

info.innerText = strStatus;

}

body {

}

#ControlPanel {

margin: auto;

border: 5px solid

}

#FuncBtn > span {

border: 2px solid;

box-shadow: 3px 3px 3px black;

cursor: pointer;

font-family: Webdings;

font-size: 24pt;

}

#volValue {

width: 25px;

background-color: black;

color: yellow;

text-align: center;

}

#settime {

background-color: #808080;

height: 10px;

cursor: pointer;

}

#progress {

background-color: #ff6a00;

height: 10px;

width: 0;

}

#Sbook, #Tbook {

width: 294px;

border: 3px solid;

float: left;

height: 200px;

overflow: auto;

cursor: copy;

}

#btnUpdateMusic {

border: 2px solid;

text-align: center;

font-size: 18pt;

cursor: pointer;

background-color: pink;

clear: both;

}

.hide {

display: none;

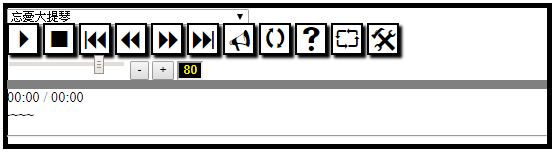
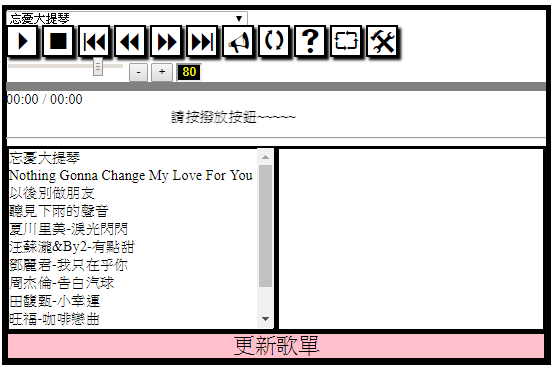
}

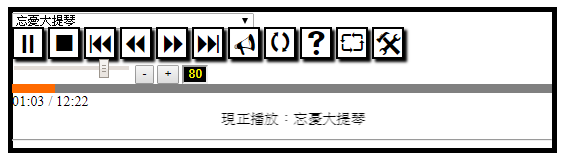
.show {

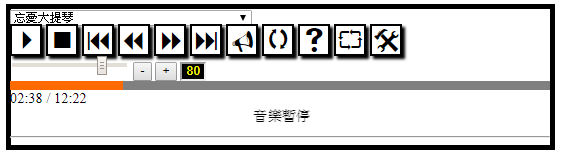
display: block;

}

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5張)：

顯示/隱藏 歌曲清單

播放功能

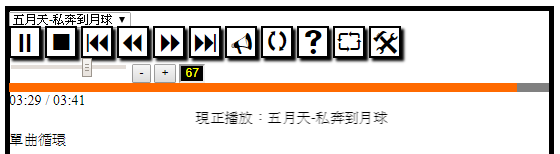
暫停 功能

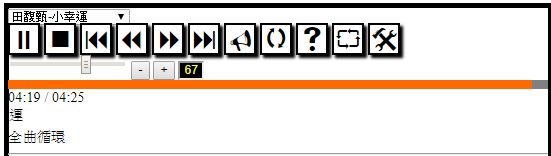
更新歌單 功能

下一首 功能

音量調整 功能



單曲循環 功能

全曲循環 功能