113年智慧行動商務系統開發班第1期

網頁前端程式設計與開發學習評量

|  |  |  |  |
| --- | --- | --- | --- |
| **評量編號** | 2-4 | | |
| **評量主題** | 前端程式設計綜合應用 | **實施時間** | 2024/11/15 |
| **評量形式** | 作業紀錄 | **評量日期** | 2024/12/16 |
| **評量目的** | 評定學員是否具備網頁前端程式設計能力，包含撰寫HTML、CSS與JavaScript程式碼並使其相互搭配運用，並可以Function架構進行程式基本解耦之實作。 | | |
| **評量說明** | **任務一**：請利用前端(Front-end)程式設計方法包含HTML、CSS與JavaScript或其他您熟知的方法，撰寫題目所指定的網頁軟體，具備題目所示的功能及適切的介面，並可於直接瀏覽器執行。  **任務二**：程式碼必須有基本的結構化設計，包括將HTML、CSS與JavaScript分開、函數化、參數化等降低耦合力的撰寫技巧。 | | |
| **提交要求** | 繳交時請以FTP上傳**相關程式檔案**與**作業文件檔**各一份，**相關程式檔案**請另放置在「完成結果檔」資料夾內，最外層資料夾名稱請用**兩碼學號與姓名**，例如1號王小明，資料夾請命名為「01王小明」。 | | |

座號：15 姓名：陳延銓

**請利用前端(Front-end)程式設計方法包含HTML、CSS與JavaScript，撰寫一網頁音樂播放器軟體，並可於直接瀏覽器執行，須達到以下要求：**

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。

10. 播放器介面上須放置本分署Logo，位置不限。

11. 介面不限，可自行設計，但須符合一般操作原則(UI/UX)。

12. CSS及JavaScript檔必須獨立在HTML檔之外，再引入使用。

13. 程式碼必須有基本的結構化設計，包括函數化、參數化等技巧。

進度條

介面上要有Logo

直接選曲功能

各功能按鈕





音量控制

音量控制

資訊看版

播放狀態呈現

**完成參考範例**

程式原始碼請貼於此處下方(若有好幾個檔案請註明檔名)：

HTML：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>音樂播放器</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">

    <link rel="stylesheet" href="musicplayer.css">

</head>

<body>

    <audio id="audioPlayer" title="曉之車" src="audio/retracks-fictionjunction-yuuka-曉之車-retracks.mp3" preload="auto"

        onended="musicState()"></audio>

    <hr>

    <div id="controlPanel" class="bg-info-subtle">

        <div id="selectUI" class=".bg-secondary-subtle">

            <select id="songSelect" onclick="changeSong(0)" class="bg-body-secondary">

                <option value="audio/retracks-fictionjunction-yuuka-曉之車-retracks.mp3">曉之車</option>

                <option value="audio/中島美嘉-雪の華-music-video.mp3">雪之華</option>

                <option value="audio/焰之扉.mp3">焰之扉</option>

                <option value="audio/水之証.mp3">水之証</option>

            </select>

            <img src="images/logo.png"/>

        </div>

        <hr>

        <span class="bg-warning-subtle" onclick="changeSong(-1)">9</span>

        <span class="bg-warning-subtle" onclick="funcChangeTime(-10)">7</span>

        <span class="bg-warning-subtle" onclick="stopMusic()">&lt;</span>

        <span class="bg-warning-subtle" id="btnFunc" onclick="playMusic()">4</span>

        <span class="bg-warning-subtle" onclick="funcChangeTime(10)">8</span>

        <span class="bg-warning-subtle" onclick="changeSong(1)">:</span>

        <span class="bi bi-volume-mute-fill bg-warning-subtle" id="btnMute" onclick="muteVolume()"></span>

        <span class="bg-warning-subtle" onclick="setCircle()">q</span>

        <span class="bg-warning-subtle" onclick="setRandom()">s</span>

        <span class="bg-warning-subtle" onclick="setAllLoop()">`</span>

        <hr>

        <input id="volumeRange" type="range" value="70" oninput="changeVolumeValue()">

        <input type="button" value="小" onclick="setVolume(-1)">

        <input type="button" value="大" onclick="setVolume(1)">

        <input class="bg-info-subtle" type="text" oninput="changeVolumeValue()" readonly>

        <hr>

        <input type="range" value="0" min="0" oninput="setProgress()">

        <hr>

        <div>00:00 / 00:00</div>

        <div>歡迎使用我的播放器</div>

        <div>正常</div>

    </div>

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

</body>

</html>

CSS：

:root{

    --vol-thumb-appearance: none;

    --vol-thumb-background-color: black;

    --vol-thumb-width: 25px;

    --vol-thumb-height: 20px;

    --vol-thumb-border-radius: 15px;

    --currenttime-thumb-background-color: rgb(111, 18, 148);

    --currenttime-thumb-border-radius: 25px;

}

#controlPanel {

    margin: auto;

    width: 610px;

    border: 5px solid rgb(90, 46, 192);

}

#controlPanel span {

    margin-left: 6px;

    font-size: 3rem;

    cursor: pointer;

    font-family: Webdings;

    border: 1px solid black;

}

#controlPanel select {

    margin-left: 6px;

    margin-top: 6px;

    /\* padding-top: 10px; \*/

    font-size: 2.2rem;

    width: 85%;

    height: 3rem;

    cursor: pointer;

    font-family: Webdings;

    border: 1px solid black;

}

#controlPanel img {

    width: 10%;

    margin-top: 6px;

    margin-right: 3px;

    margin-left: 3px;

    border: 1px solid black;

}

#controlPanel input:nth-of-type(1) {

    appearance: none;

    width: 80%;

    height: 8px;

    border: 2px solid rgb(54, 18, 138);

    background-color: transparent;

}

#controlPanel input:nth-of-type(1):focus::-webkit-slider-thumb {

    box-shadow: 0 0 0 1px #fff, 0 0 0 0.25rem rgba(13, 110, 253, 0.25);

}

#controlPanel input:nth-of-type(1):focus::-moz-range-thumb {

    box-shadow: 0 0 0 1px #fff, 0 0 0 0.25rem rgba(13, 110, 253, 0.25);

  }

#controlPanel input:nth-of-type(1)::-webkit-slider-thumb {

    appearance: var(--vol-thumb-appearance);

    background-color: var(--vol-thumb-background-color);

    width: var(--vol-thumb-width);

    height: var(--vol-thumb-height);

    border-radius: var(--vol-thumb-border-radius);

}

#controlPanel input:nth-of-type(1)::-moz-range-thumb {

    appearance: var(--vol-thumb-appearance);

    background-color: var(--vol-thumb-background-color);

    width: var(--vol-thumb-width);

    height: var(--vol-thumb-height);

    border-radius: var(--vol-thumb-border-radius);

}

#controlPanel input:nth-of-type(2) {

    border: 2px solid rgb(14, 81, 126);

    border-radius: 15px;

    height: 1.8rem;

    background-color: green;

    color: wheat;

}

#controlPanel input:nth-of-type(3) {

    border: 2px solid rgb(14, 81, 126);

    border-radius: 15px;

    height: 1.8rem;

    background-color: red;

    color: wheat;

}

#controlPanel input:nth-of-type(4) {

    width: 35px;

    height: 1.8rem;

}

#controlPanel input:nth-of-type(5) {

    appearance: none;

    width: 97%;

    height: 8px;

    border: 2px solid rgb(71, 10, 10);

}

#controlPanel input:nth-of-type(5)::-webkit-slider-thumb {

    appearance: var(--vol-thumb-appearance);

    background-color: var(--currenttime-thumb-background-color);

    width: var(--vol-thumb-width);

    height: var(--vol-thumb-height);

    border-radius: var(--currenttime-thumb-border-radius);

}

#controlPanel input:nth-of-type(5)::-moz-range-thumb {

    appearance: var(--vol-thumb-appearance);

    background-color: var(--currenttime-thumb-background-color);

    width: var(--vol-thumb-width);

    height: var(--vol-thumb-height);

    border-radius: var(--currenttime-thumb-border-radius);

}

#controlPanel div:nth-of-type(2),

div:nth-of-type(3),

div:nth-of-type(4){

    margin: auto;

    text-align: center;

    font-size: 2rem;

    font-family:標楷體;

}

JavaScript：

const player = document.getElementById("audioPlayer");

const controlPanel = document.getElementById("controlPanel");

const select = document.getElementById("songSelect");

const playBtn = controlPanel.getElementsByTagName("span")[3];

const muteBtn = controlPanel.getElementsByTagName("span")[6];

const circleBtn = controlPanel.getElementsByTagName("span")[7];

const randomBtn = controlPanel.getElementsByTagName("span")[8];

const allLoopBtn = controlPanel.getElementsByTagName("span")[9];

const input = controlPanel.getElementsByTagName("input");

const volumeRange = input[0];

const txtVol = input[3];

const progressBar = input[4];

const infoPanel = controlPanel.getElementsByTagName("div");

const songTimeText = infoPanel[1];

const songNameText = infoPanel[2];

const statusText = infoPanel[3];

var funcIndex = 0;                                             // 0是沒有打開功能，1是單曲循環，2是隨機播放，3是全部循環

function changeFunc(btn, text, clickFunc) {

    btn.innerText = text;

    btn.onclick = clickFunc;

}

function getCurrentTime() {

    songTimeText.innerText = getTimeFormat(player.currentTime) + "/" + getTimeFormat(player.duration);

    progressBar.value = player.currentTime \* 10000;

}

function changeSong(n) {

    let index = select.selectedIndex;

    player.src = select.children[index + n].value;

    player.title = select.children[index + n].innerText;

    select.selectedIndex = index + n;

    select.children.selected = true;

    if (playBtn.innerText == ";")

        player.onloadeddata = playMusic;

}

function funcChangeTime(sec) {

    player.currentTime += sec;

}

function getTimeFormat(t) {

    let m = parseInt(t / 60);

    m = m < 10 ? "0" + parseInt(t / 60) : parseInt(t / 60);

    let s = parseInt(t % 60);

    s = s < 10 ? "0" + parseInt(t % 60) : parseInt(t % 60);

    return m + ":" + s;

}

function stopMusic() {

    pauseMusic();

    player.currentTime = 0;

    songNameText.innerText = "停止播放" + player.title + ".......";

}

function playMusic() {

    player.play();

    changeFunc(playBtn, ";", pauseMusic);

    setInterval(()=> getCurrentTime(), 0.1)

    songNameText.innerText = "正在播放" + player.title + "中.......";

    progressBar.max = player.duration \* 10000;

    changeVolumeValue();

}

function pauseMusic() {

    player.pause();

    changeFunc(playBtn, "4", playMusic);

    songNameText.innerText = player.title + "暫停播放中.......";

}

function muteVolume() {

    controlmute("bi bi-volume-up-fill bg-warning-subtle", unmuteVolume)

}

function unmuteVolume() {

    controlmute("bi bi-volume-mute-fill bg-warning-subtle", muteVolume)

}

function controlmute(text, click) {

    muteBtn.className = text;

    player.muted = !player.muted;

    muteBtn.onclick = click;

}

//先重設單曲循環/隨機播放/全部循環的按鈕背景色

function resetAllFuncBtnColor() {

    circleBtn.className = "bg-warning-subtle";

    randomBtn.className = "bg-warning-subtle";

    allLoopBtn.className = "bg-warning-subtle";

    statusText.innerText = "正常"

}

function setCircle() {

    resetAllFuncBtnColor();

    if (funcIndex != 1) {

        circleBtn.className = "bg-info";

        funcIndex = 1;

        statusText.innerText = "單曲循環播放中"

    }

    else

        funcIndex = 0;

}

function setRandom() {

    resetAllFuncBtnColor();

    if (funcIndex != 2) {

        randomBtn.className = "bg-info";

        funcIndex = 2;

        statusText.innerText = "隨機播放中"

    }

    else

        funcIndex = 0;

}

function setAllLoop() {

    resetAllFuncBtnColor();

    if (funcIndex != 3) {

        allLoopBtn.className = "bg-info";

        funcIndex = 3;

        statusText.innerText = "全部循環播放中"

    }

    else

        funcIndex = 0;

}

function musicState() {

    if (funcIndex == 1) {

        //單體循環

        changeSong(0);

    }

    else if (funcIndex == 2) {

        //隨機播放

        let n = Math.floor(Math.random() \* select.children.length);

        n -= select.selectedIndex;

        changeSong(n);

    }

    else if (funcIndex == 3 && select.length == select.selectedIndex + 1) {

        //全部循環

        changeSong(0 - select.selectedIndex);

    }

    else if (select.length == select.selectedIndex + 1) {

        stopMusic();

    }

    else {

        changeSong(1);

    }

}

function setProgress() {

    player.currentTime = parseInt(progressBar.value / 10000);

}

function changeVolumeValue() {

    txtVol.value = volumeRange.value;

    player.volume = volumeRange.value / 100;

}

function setVolume(v) {

    let val = player.volume \* 100 + v;

    if (val > 100)

        val = 100;

    else if (val < 0)

        val = 0;

    volumeRange.value = val;

    changeVolumeValue();

}

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)：







 

 