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** | 2025/5/21 |
| **評量形式** | 作業紀錄 | **評量日期** | 2025/6/13 |
| **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：01 姓名：劉家妤

**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>Music Player</title>

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

</head>

<body>

    <div><audio id="myMusic" src="../music/moon.mp3" onended="musicPlay()"></audio></div>

    <div id="container">

        <select id="musicList" onchange="changeMusic(0)">

            <option value="../music/moon.mp3">私奔到月球</option>

            <option value="../music/airball.mp3">告白氣球</option>

            <option value="../music/elysees.mp3">Elysees</option>

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

            <option value="../music/rain.mp3">Rain</option>

            <option value="../music/travel.mp3">旅行</option>

            <option value="../music/dancing at night.mp3">我在夜裡跳舞</option>

        </select>

        <div id="img">

            <img src="../img/1.jpg" alt="私奔到月球">

        </div>

        <div id="controlPanel">

            <div id="title">

                <div id="mainTitle">私奔到月球</div>

                <div>

                    <span id="time">00:00 / 00:00</span>

                    <span id="status">循環播放</span>

                </div>

                <input id="titlebar" type="range" oninput="setTime();" min="0" step="1" value="0" >

            </div>

            <div id="btn">

                <span onclick="changeMusic(-1);">9</span>

                <span onclick="backOrForward(-5);">7</span>

                <span id="control1" onclick="play();">4</span> <!-- 播放音樂 -->

                <span onclick="backOrForward(5);">8</span>

                <span onclick="changeMusic(1);">:</span>

                <span onclick="stop();"><</span>

                <span id="mode" onclick="musicMode()">q</span>

            </div>

            <div id="volumeControl">

                <span id="btnSound" onclick="sound();">U</span>

                <span onclick="volumnChange(-1);">-</span>

                <input type="range" oninput="setVolumnRange();" value="60">

                <span onclick="volumnChange(1);">+</span>

                <input type="text" value="60" readonly>

                <img src="../img/logo.png" alt="分屬logo">

            </div>

        </div>

    </div>

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

</body>

</html>

CSS：

:root {

    --thumb-appearance: none;

    --thumb-width: 15px;

    --thumb-height: 15px;

    --thumb-border-radius: 50%;

    --thumb-width-o: 10px;

    --thumb-height-o: 10px;

    /\* 進度條 \*/

    --progress-thumb-background-color: rgb(82, 162, 182);

    --progress-thumb-border: 2px solid rgb(4, 170, 211);

    /\* 音量條 \*/

    --vol-thumb-background-color: rgb(218, 180, 223);

    --vol-thumb-border: 2px solid rgb(228, 124, 210);

}

#container {

    width: 40%;

    height: 700px;

    background-color: lightblue;

    margin: auto;

    box-shadow: 0px 0px 10px gray;

    display: flex;

    justify-content: flex-start;

    flex-direction: column;

    align-items: center;

    min-width: 485px;

}

#musicList {

    width: 80%;

    height: 30px;

    margin-top: 2rem;

    border-radius: 5px;

    background-color: rgb(77, 163, 212);

    color: white;

    font-size: 1.2rem;

    text-align: center;

    border: 0px;

}

#img {

    position: relative;

    width: 80%;

    height: 60%;

}

#img img {

    width: 100%;

    height: 100%;

    object-fit: cover;

    object-position: center;

    box-shadow: 0px 0px 10px gray;

}

#controlPanel {

    width: 90%;

    height: 100px;

    bottom: 2rem;

    padding: 10px;

    align-items: center;

}

#btn {

    margin-top: 3%;

    display: flex;

    justify-content: space-around;

    align-items: center;

}

#btn span {

    border: 1px solid violet;

    padding: 5px;

    background-color: pink;

    border-radius: 50%;

    box-shadow: 0px 0px 10px gray;

    cursor: pointer;

    position: relative;

    color: white;

    font-size: 2rem;

    font-family: webdings;

    text-align: center;

}

#volumeControl {

    display: flex;

    justify-content: space-around;

    align-items: center;

    margin-top: 3%;

}

#title {

    top: 2.5rem;

    margin: auto;

    height: 80px;

    overflow: hidden;

    text-align: center;

}

#mainTitle {

    font-size: 1.5rem;

    font-weight: bold;

    color: white;

    text-shadow: 0px 0px 10px gray;

}

.setMute {

    text-decoration: line-through;

    color: rgb(73, 231, 210);

}

#volumeControl input[type="text"] {

    width: 30px;

    height: 25px;

    background-color: rgb(77, 163, 212);

    color: white;

    font-size: 0.85rem;

    text-align: center;

    border-radius: 5px;

    border: 0px;

    margin-left: 0.5rem;

    display: none;

}

#title input[type="range"],

#volumeControl input[type="range"] {

    width: 400px;

    height: 10px;

    appearance: none;

    border-radius: 5px;

    cursor: pointer;

}

/\* chrom/edge 的進度條&音量調圓球設定 \*/

#title input[type="range"]::-webkit-slider-thumb,

#volumeControl input[type="range"]::-webkit-slider-thumb {

    appearance: var(--thumb-appearance);

    width: var(--thumb-width-o);

    height: var(--thumb-height-o);

}

#title input[type="range"]::-webkit-slider-thumb:hover {

    appearance: var(--thumb-appearance);

    width: var(--thumb-width);

    height: var(--thumb-height);

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

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

    border: var(--progress-thumb-border);

    transition: 0.5s;

}

#volumeControl input[type="range"]::-webkit-slider-thumb:hover {

    appearance: var(--thumb-appearance);

    width: var(--thumb-width);

    height: var(--thumb-height);

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

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

    border: var(--vol-thumb-border);

}

/\* 其他版本瀏覽器 的進度條%音量調圓球設定 \*/

#title input[type="range"]::-moz-range-thumb,

#volumeControl input[type="range"]::-moz-range-thumb {

    appearance: var(--thumb-appearance);

    width: var(--thumb-width-o);

    height: var(--thumb-height-o);

}

#title input[type="range"]::-moz-range-thumb:hover {

    appearance: var(--thumb-appearance);

    width: var(--thumb-width);

    height: var(--thumb-height);

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

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

    border: var(--progress-thumb-border);

}

#volumeControl input[type="range"]::-moz-range-thumb:hover {

    appearance: var(--thumb-appearance);

    width: var(--thumb-width);

    height: var(--thumb-height);

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

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

    border: var(--vol-thumb-border);

}

#volumeControl span {

    border: 1px solid rgb(47, 205, 226);

    background-color: rgb(130, 215, 230);

    cursor: pointer;

    font-size: 0.8rem;

    padding: 3px;

    border-radius: 10%;

}

#btnSound {

    border: 1px solid;

    padding: 5px;

    background-color: rgb(41, 223, 208) !important;

    border-radius: 50% !important;

    box-shadow: 0px 0px 10px gray;

    color: white;

    font-size: 1.5rem !important;

    font-family: webdings;

    margin-right: 0.5rem;

}

#volumeControl img {

    width: 50px;

    height: 50px;

    margin-left: 0.5rem;

}

JavaScript：

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

        //播放&暫停

        function play() {

            if (myMusic.paused) {

                myMusic.play();

                document.getElementById("control1").innerHTML = ";";

                setTimeInitial();

                console.log("播放音樂");

            } else {

                myMusic.pause();

                document.getElementById("control1").innerHTML = "4";

            }

        }

        //停止

        function stop() {

            myMusic.pause();

            //重製播放時間

            myMusic.currentTime = 0;

            myMusic.pause(); //音樂暫停

            document.getElementById("control1").innerHTML = "4"; //修改播放符號

        }

        //快轉&快退

        function backOrForward(v) {

            myMusic.currentTime += v;

        }

 ////////【音樂進度條】//////////////////////////////////////////////////////////////////////////////////////////////////////

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

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

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

        var text = document.getElementById("status");

        //時間格式

        function formatTime(t) {

            var minutes = parseInt(t.toFixed(0) / 60);

            var seconds = parseInt(t.toFixed(0) % 60);

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

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

            return minutes + ":" + seconds;

        }

        //點擊音樂進度條

        function setTime(){

            myMusic.currentTime = titlebar.value / 10000; //初始化音樂當前時間

        }

        //更新音樂進度條

        function setProcessRange(){

            var currentTime = myMusic.currentTime; //目前播放時間

            var duration = myMusic.duration; //音樂總長度

            time.innerHTML = formatTime(currentTime) + " / " + formatTime(duration); //顯示時間

            titlebar.value = currentTime \*10000;

            var percent = (currentTime / duration) \* 100; //計算進度條百分比

            //塗音量條的顏色

            titlebar.style.backgroundImage = `linear-gradient(to right, rgb(157, 109, 219) 0%, rgb(109, 111, 219) ${percent}%,white ${percent}%)`;

        }

        //初始化音樂進度條

        function setTimeInitial(){

                titlebar.max = myMusic.duration\*10000; //音樂的總長度

                setInterval(setProcessRange,10); //更新音樂進度條

        }

////////【歌曲控制(選曲、模式)】////////////////////////////////////////////////////////////////////////////////////////////////////////////////

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

        //選擇歌曲

        function changeMusic(n) {

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

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

            var img = document.getElementById("img").children[0];

            var v = musicList.selectedIndex; //獲取目前選擇的歌曲索引

            var index = v + n; //計算新的歌曲索引

            if(index < 0) {

                index = musicList.length - 1; //如果索引小於0，則選擇最後一首歌

            } else if (index >= musicList.length) {

                index = 0; //如果索引大於歌曲數量，則選擇第一首歌

            }

            myMusic.src = musicList.options[index].value; //更改音樂來源

            mainTitle.innerHTML = musicList.options[index].innerText; //更改音樂標題

            musicList.children[index].selected = true; //更新選擇的歌曲索引

            //更改圖片來源

            if((index + 1) % 7 == 0) {

                img.src = `../img/7.jpg`; //如果是第七首歌，則圖片顯示第七張

            }else{

                img.src = `../img/${(index + 1) % 7}.jpg`;

            }

            console.log("目前選擇的歌曲索引：" + index);

            console.log("v : " + v);

            console.log("n : " + n);

            myMusic.load();

            if(document.getElementById("control1").innerHTML == ";") {

                myMusic.onloadeddata = play; //等歌曲載入完畢後再播放音樂

            }

            myMusic.onloadedmetadata = function() {

                setTimeInitial();

    };

        }

        //音樂模式 (循環、隨機、單曲循環)

        function musicPlay(){

            if(text.innerText == "循環播放" && musicList.length == musicList.selectedIndex + 1){

                changeMusic(0 - musicList.selectedIndex);

            } else if (text.innerText == "隨機播放") {

                var n = Math.floor(Math.random() \* musicList.length); //隨機在musicList中選擇一首歌

                if(n != musicList.selectedIndex) {

                    changeMusic(n - musicList.selectedIndex);

                } else {

                    changeMusic(1); //如果隨機到的歌曲是目前播放的歌曲，則改為下一首

                }

            } else if (text.innerText == "單曲循環") {

                changeMusic(0);

            } else{

                changeMusic(1);

            }

        }

        function musicMode(){

            if (text.innerText == "循環播放") {

                mode.innerHTML = "s";

                text.innerHTML = "隨機播放";

            } else if (text.innerText == "隨機播放") {

                mode.innerHTML = "`";

                text.innerHTML = "單曲循環";

            }else if (text.innerText == "單曲循環") {

                mode.innerHTML = "q";

                text.innerHTML = "循環播放";

            }

        }

///////【音效控制】///////////////////////////////////////////////////////////////////////////////////////////////////////////////

        //靜音

        function sound() {

            myMusic.muted = !myMusic.muted;

            event.target.className = event.target.className == "setMute" ? "" : "setMute";

        }

        //音量控制

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

        var rangeVolumn = volumeControl.children[2];

        var textVolumn = volumeControl.children[4];

        //微調(融合) - 音量降低&加大

        function volumnChange(v) {

            rangeVolumn.value = parseInt(rangeVolumn.value) + v;

            setVolumnRange();

        }

        //音量條

        function setVolumnRange() {

            textVolumn.value = rangeVolumn.value;

            myMusic.volume = textVolumn.value / 100; //真正寫入音量屬性值

            //塗音量條的顏色

            rangeVolumn.style.backgroundImage = `linear-gradient(to right, pink 0%,rgb(218, 180, 223) ${rangeVolumn.value}%,white ${rangeVolumn.value}%)`;

        }

        setVolumnRange(); //初始化音量

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)：







