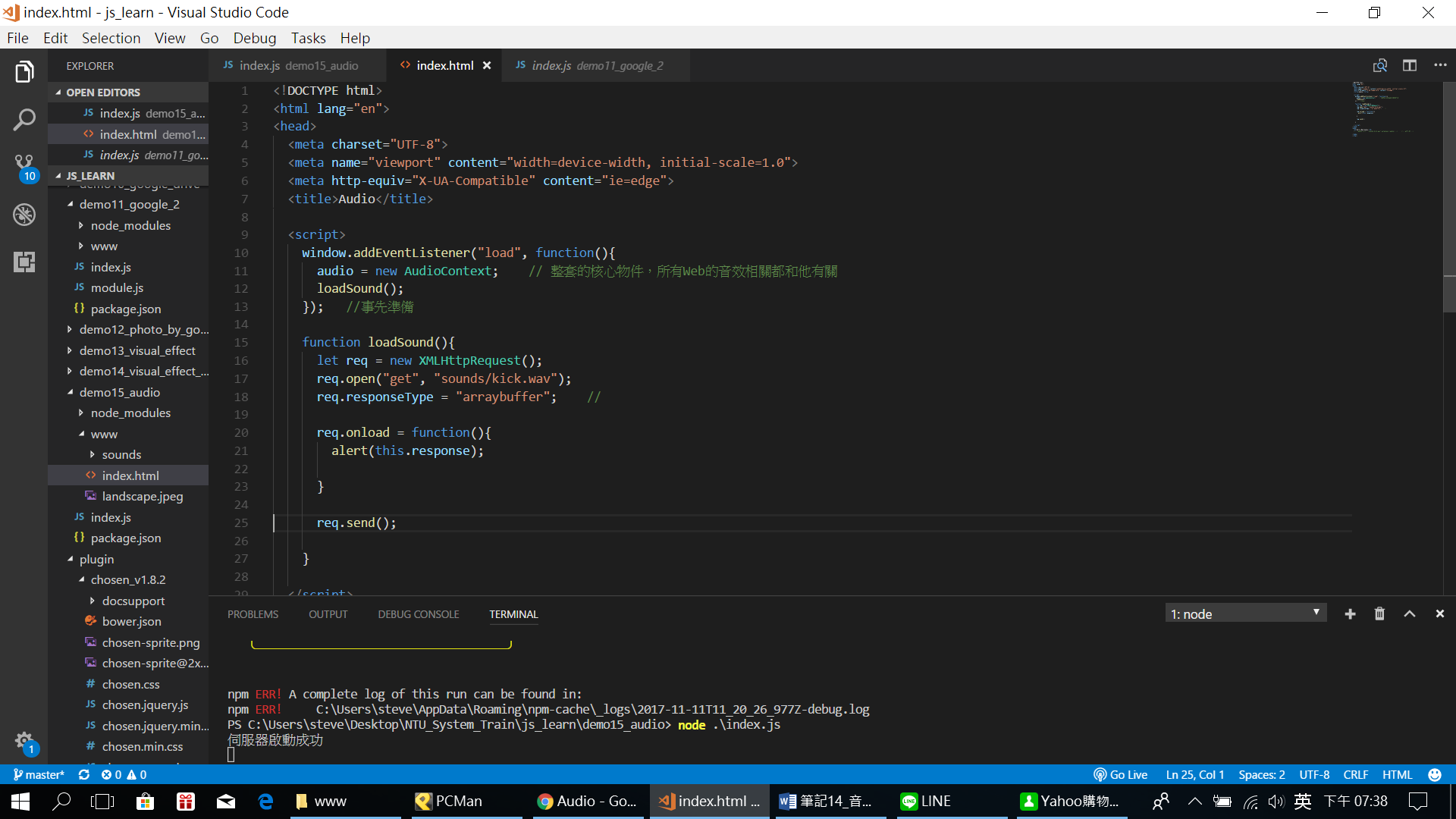


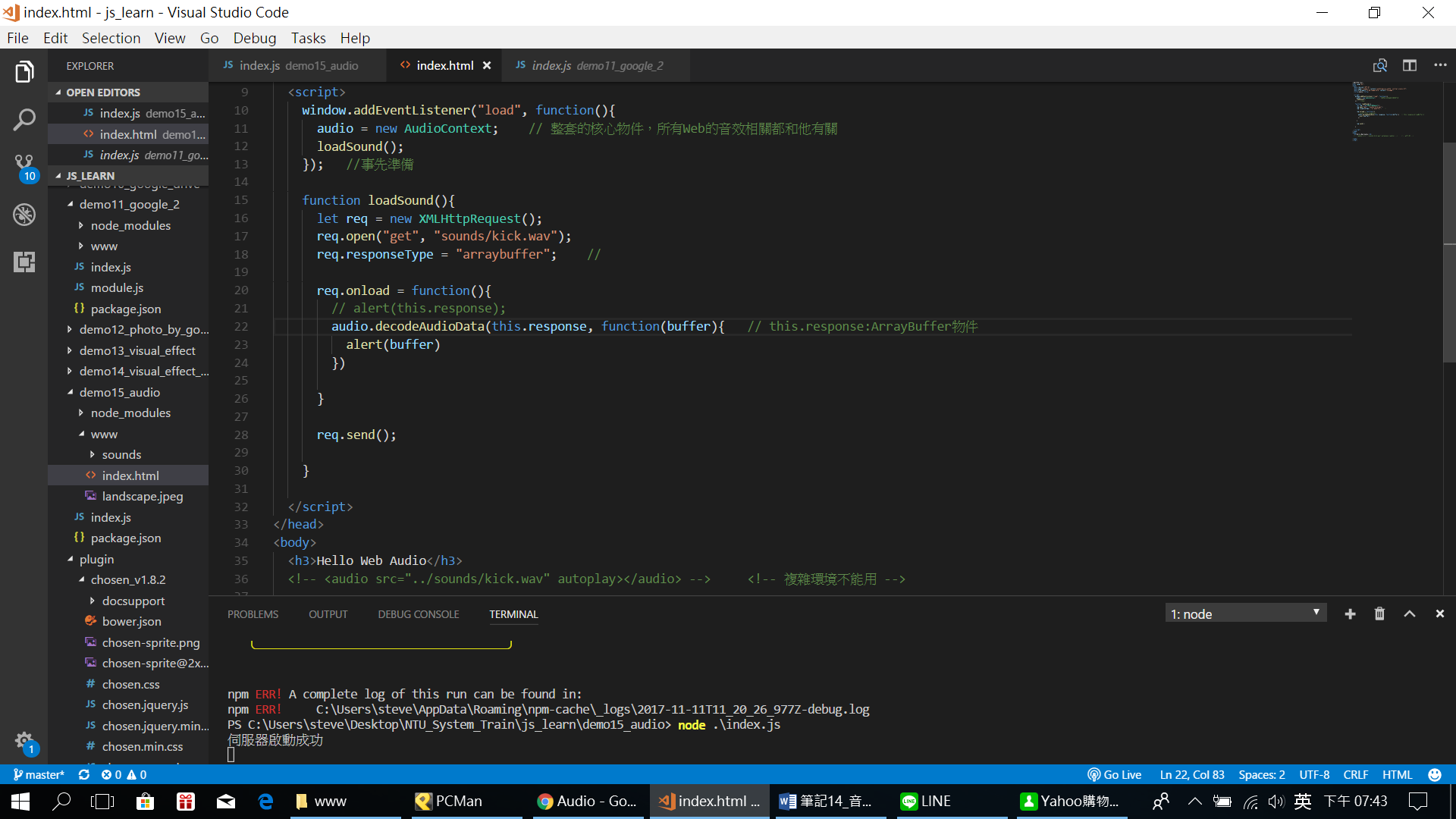
<!-- <audio src="../sounds/kick.wav" autoplay></audio> --> <!-- 複雜環境不能用 -->

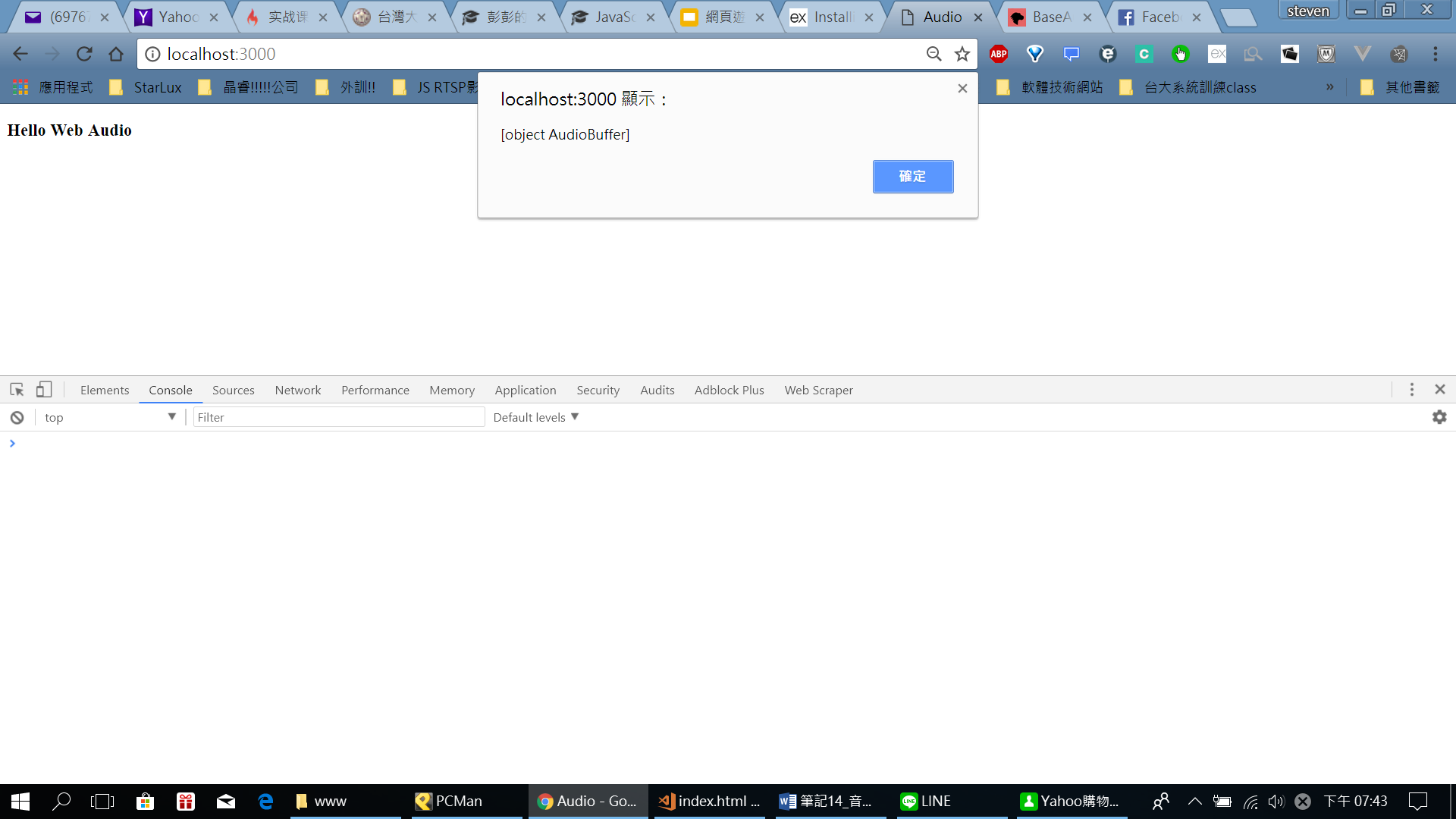
流程:

File > AudioBuffer (此之前是事先處理) > 1. 播放階段(立刻處理): 首先要有個BufferSource物件 可以不調直接輸出

1. 先雛形:



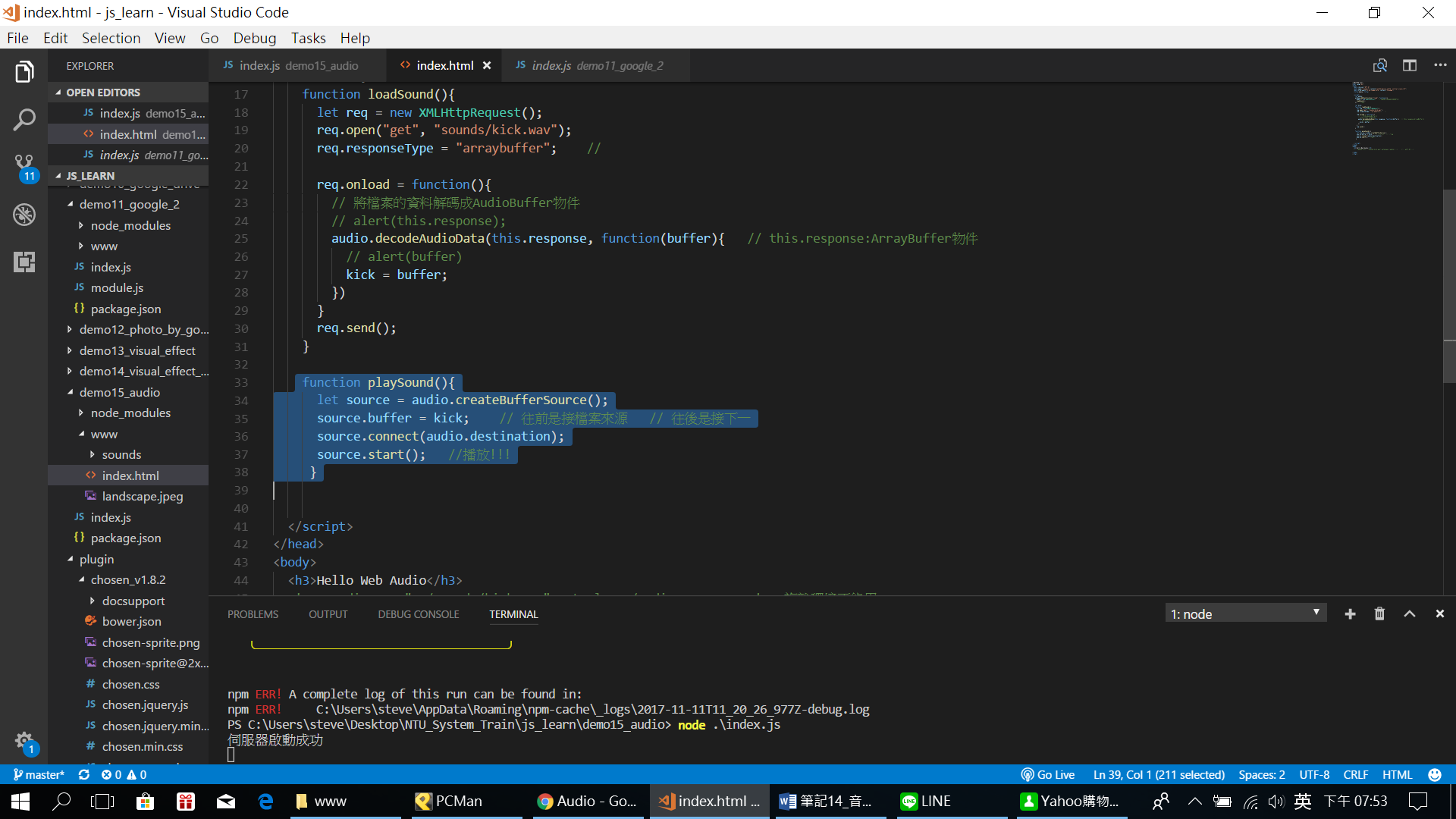




因為:

播放階段(立刻處理): 首先要有個BufferSource物件 可以不調直接輸出

所以:



在browser輸入playsound():

就會聽到聲音囉!

改成想要播什麼就播什麼!!:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Audio</title>

<script>

let audio;

let sounds; // 把所有音效都放進這個物件

window.addEventListener("load", function(){

audio = new AudioContext; // 整套的核心物件，所有Web的音效相關都和他有關

sounds = {}; // 記得初始化!

loadSound("kick");

loadSound("snare");

loadSound("hihat");

}); //事先準備

function loadSound(name){

let req = new XMLHttpRequest();

req.open("get", "sounds/"+ name +".wav");

req.responseType = "arraybuffer";

req.onload = function(){

// 將檔案的資料解碼成AudioBuffer物件

// alert(this.response);

audio.decodeAudioData(this.response, function(buffer){ // this.response:ArrayBuffer物件

// alert(buffer)

// sounds = buffer;

sounds[name] = buffer; // 為了讓sounds裡面的key是name，並可以用name的方式塞值 (變數可以帶入當key)

})

}

req.send();

}

function playSound(buffer /\*要播放的AudioBuffer\*/){

let source = audio.createBufferSource();

source.buffer = buffer; // 往前是接檔案來源 // 往後是接下一

source.connect(audio.destination);

source.start(); //播放!!!

}

function playMusic(){

alert(audio.currentTime) // audio context建立到現在過了多久(從音效環境建立到現在 就算都作時間還是依值往前run! 以秒為單位!!)

}

</script>

</head>

<body>

<h3>Hello Web Audio</h3>

<!-- <audio src="../sounds/kick.wav" autoplay></audio> --> <!-- 複雜環境不能用 -->

<button onclick="playMusic();">Play</button>

</body>

</html>

用browser就可以聽到:

Ex:

playSound(sounds.kick);

undefined

playSound(sounds.snare);

undefined

playSound(sounds.hihat);

undefined

playSound(sounds.hihat);

undefined

playSound(sounds.snare);

undefined 都會有聲音

時間重要 有一個從load到現在的時間:

來改一下:

如果指定的時間是過去時間>> 會立刻撥

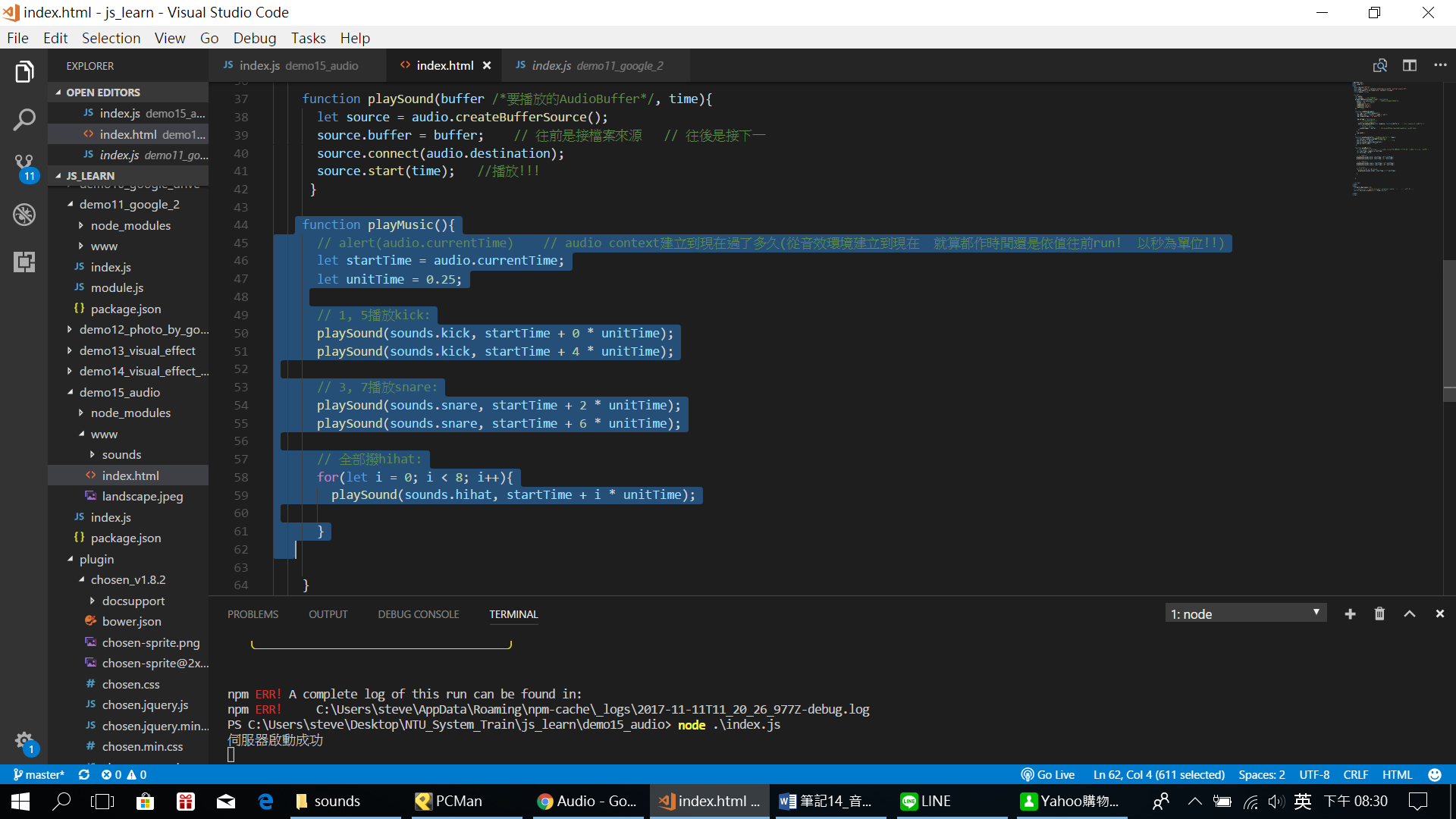
如果指定未來時間才會再該時間點(從load後)才撥

例如想做:

Kick 想發1, 5 秒

Snare > 3, 7秒

Hihat> 每一秒

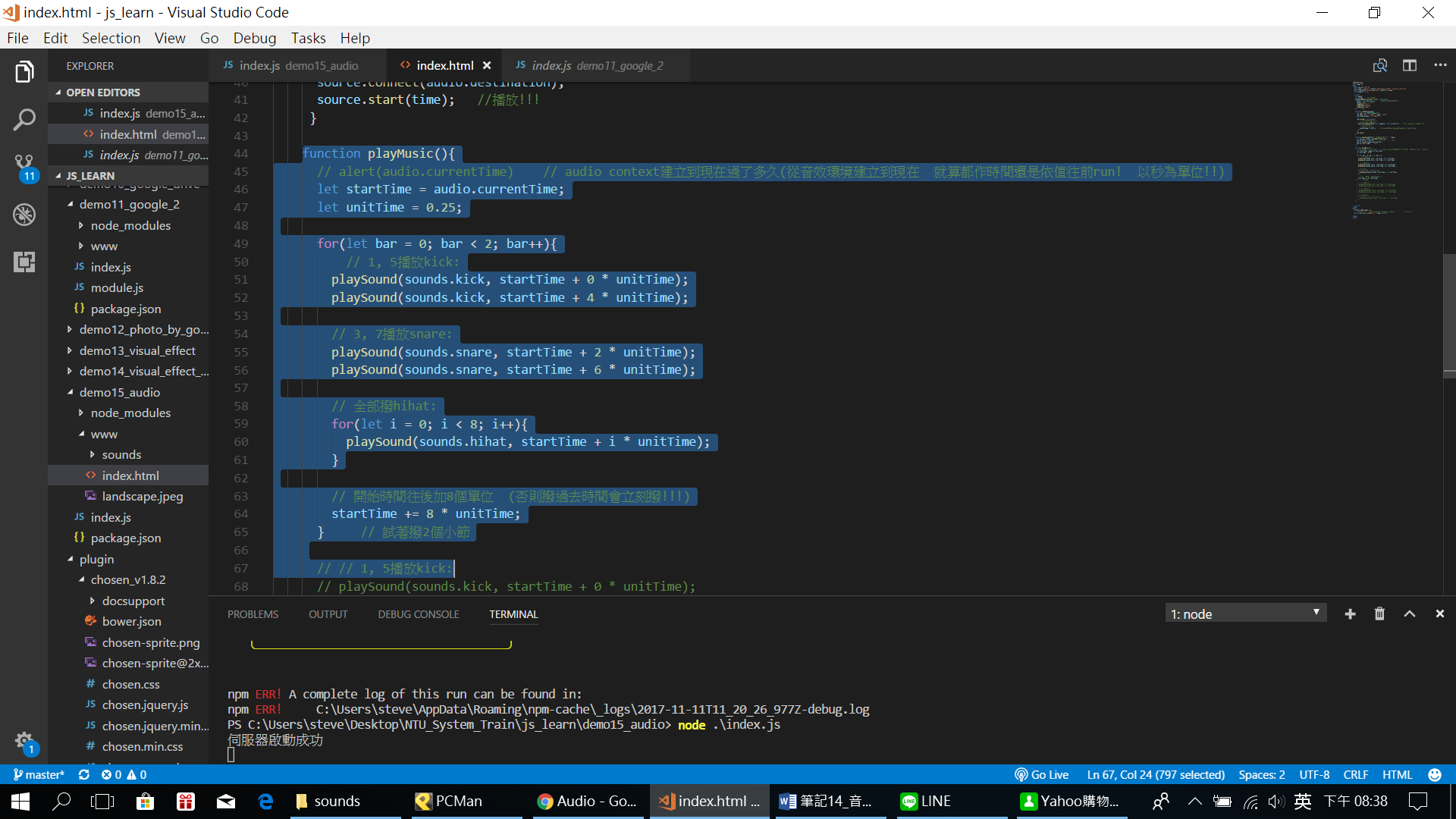


試著去撥放XD

就有很酷的聲音出來啦~~~

接下來我們想撥2個小節:

改為for 迴圈撥2次，然後記得一次迴圈後的starttime要加一節的時間!!!



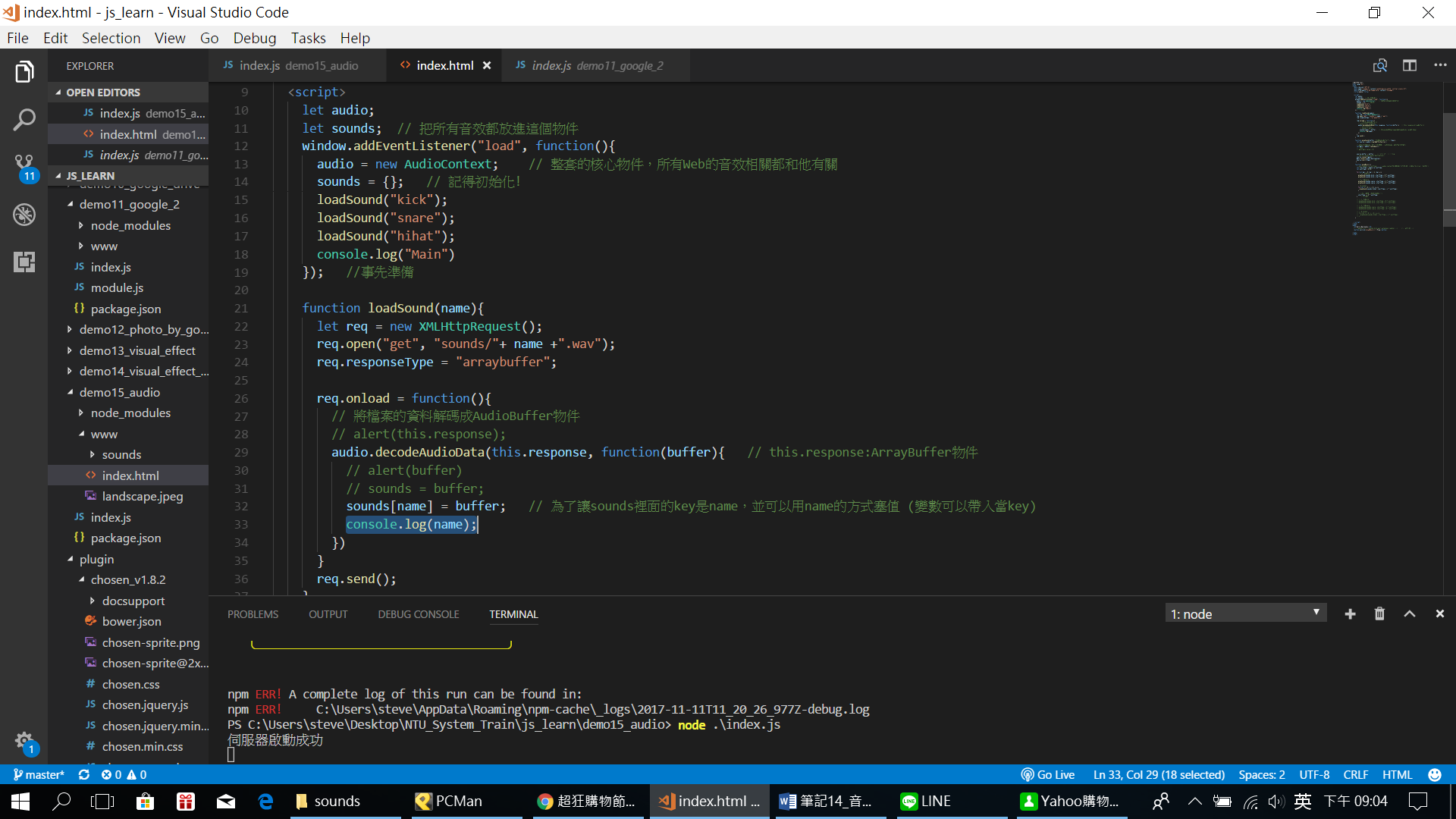
就會發現可以撥2節共16個單位囉

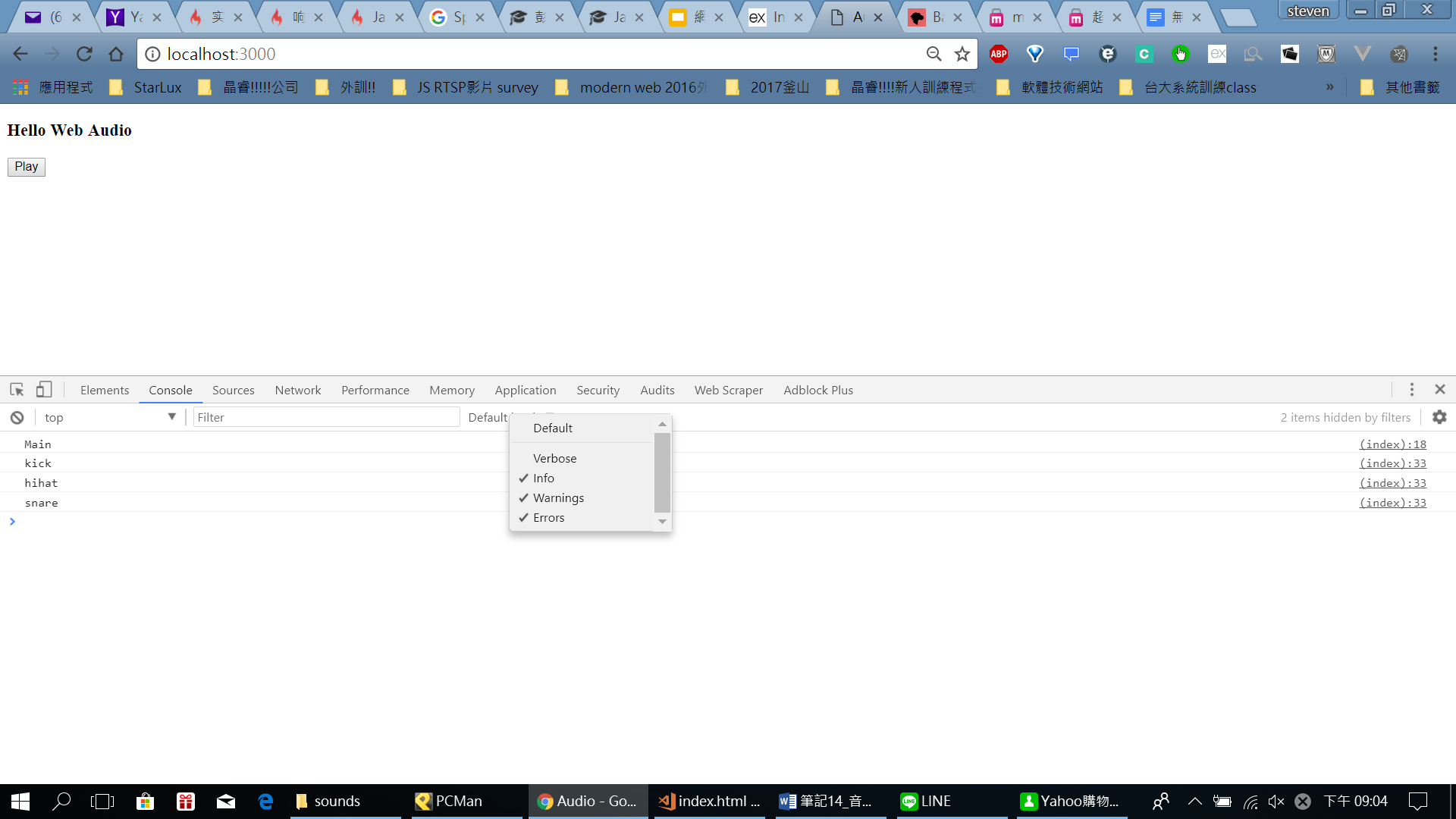
搜尋:

MDN 有更多的是可以調! 大小聲拉 bla

<https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API>

一定要載入完成!!

所以來看一下



結果發現Main竟然在最前面!!!!!!!!!

>>>>>>>>>>>>

寫進階一點點的程式 觀念要有!!

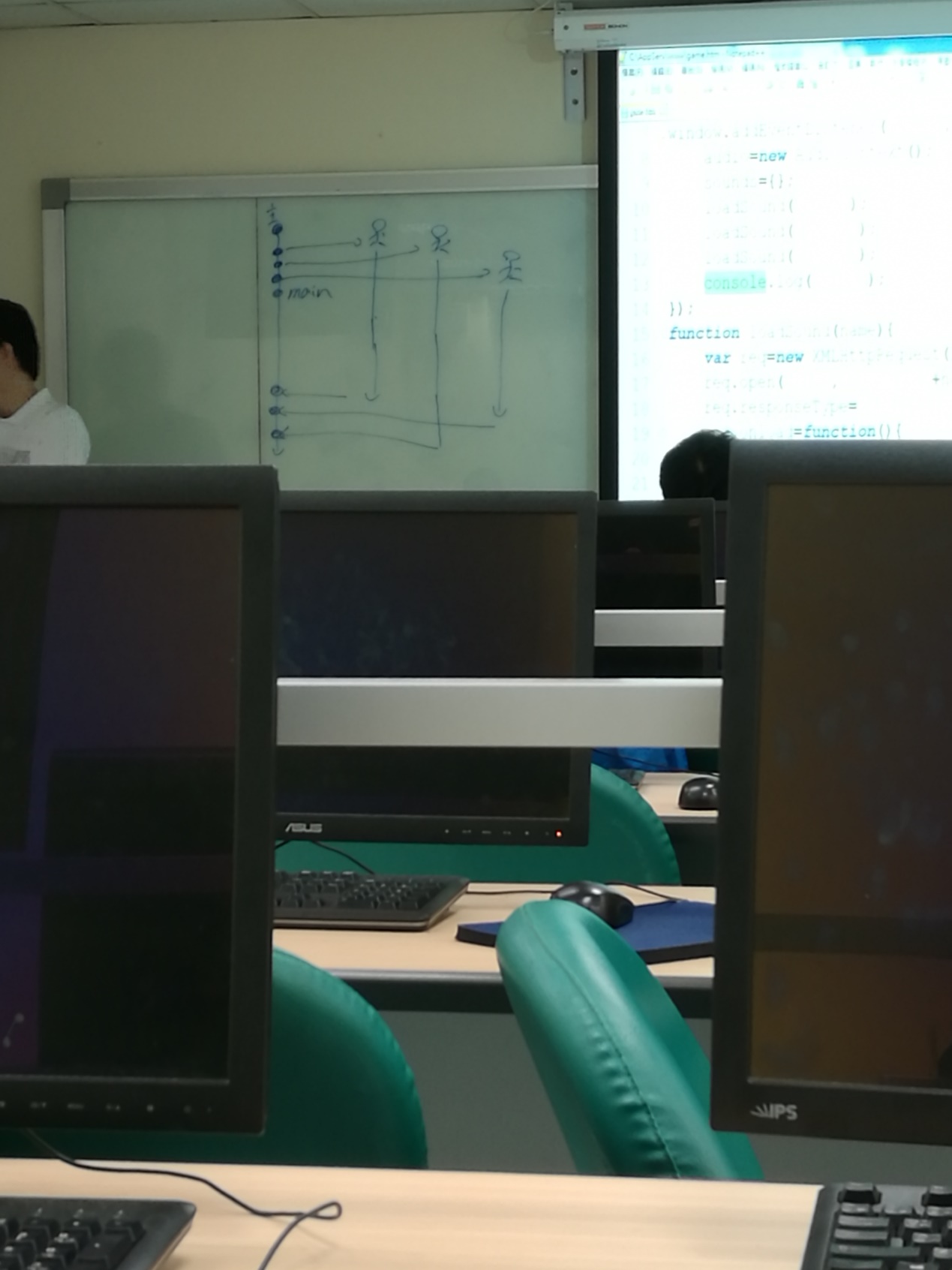
剛的code:

***是一個非同步的動作!!!***

讀取音效、解碼是派出小弟去做

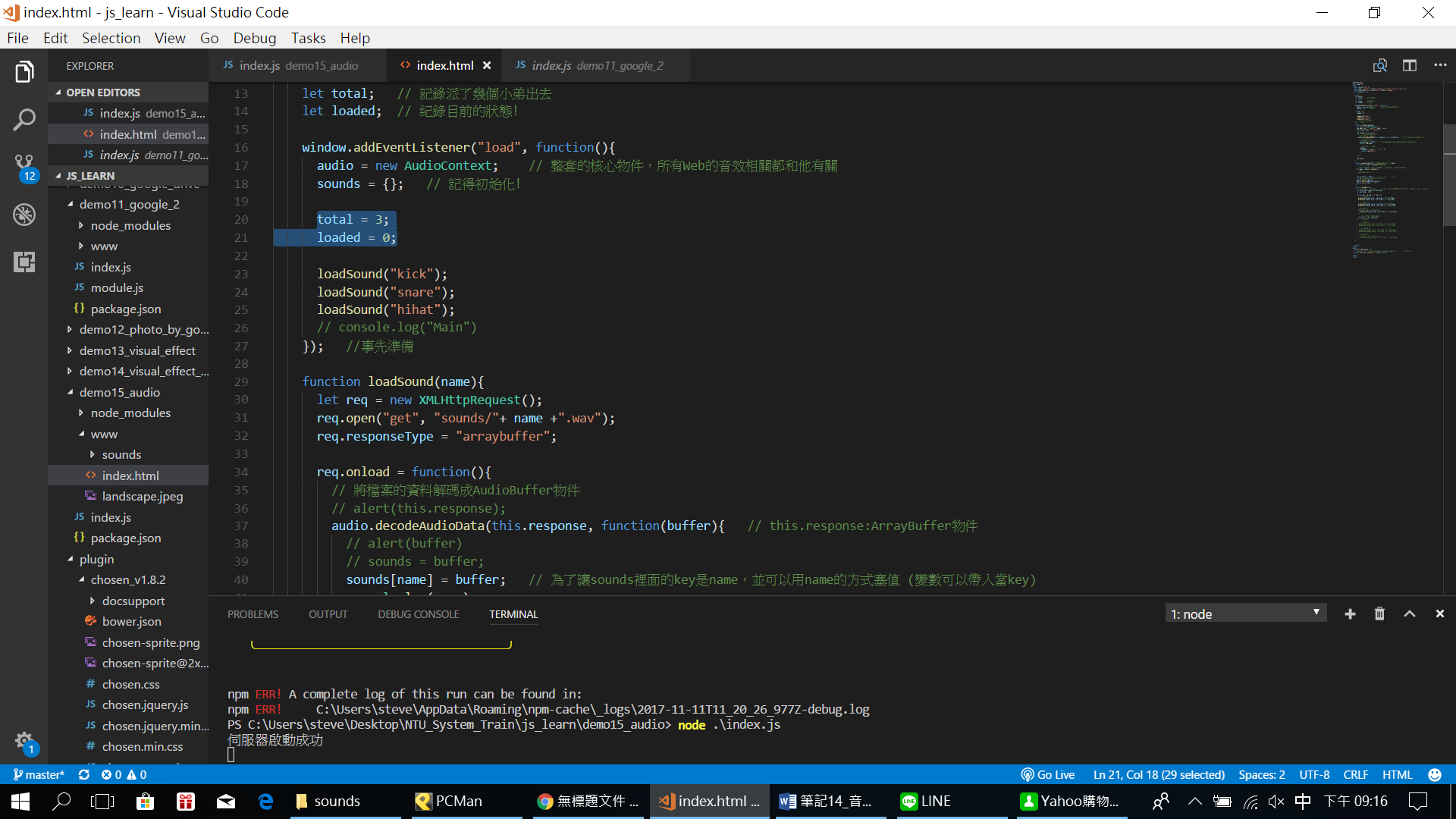
所以主程式只是命令小弟工作，第2個loadSound是派出第2個小弟去處理，以此類推!

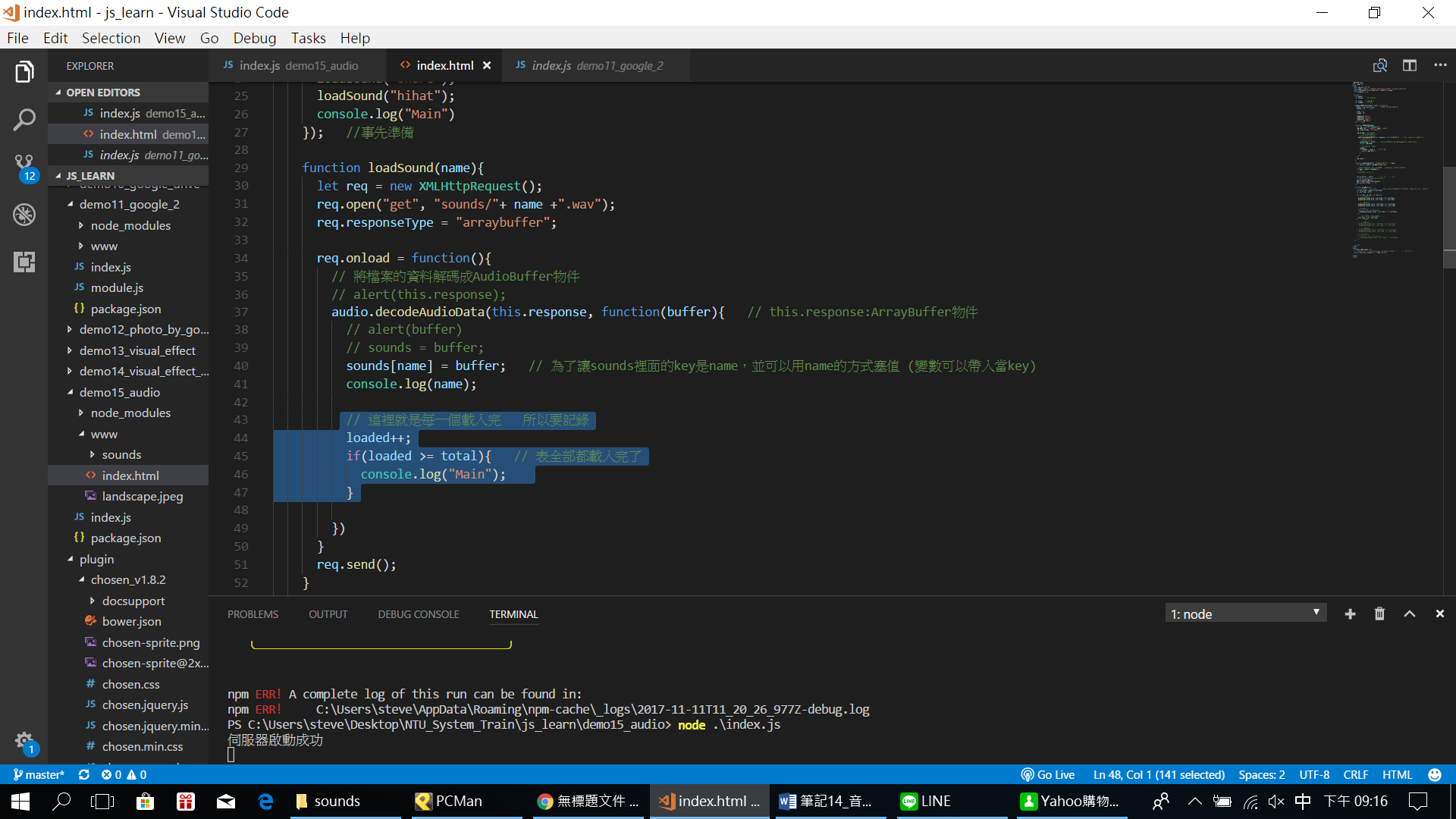
先去的不一定先回來喔!!!!!!!!!!!!!!!!!!!!!!!!!!



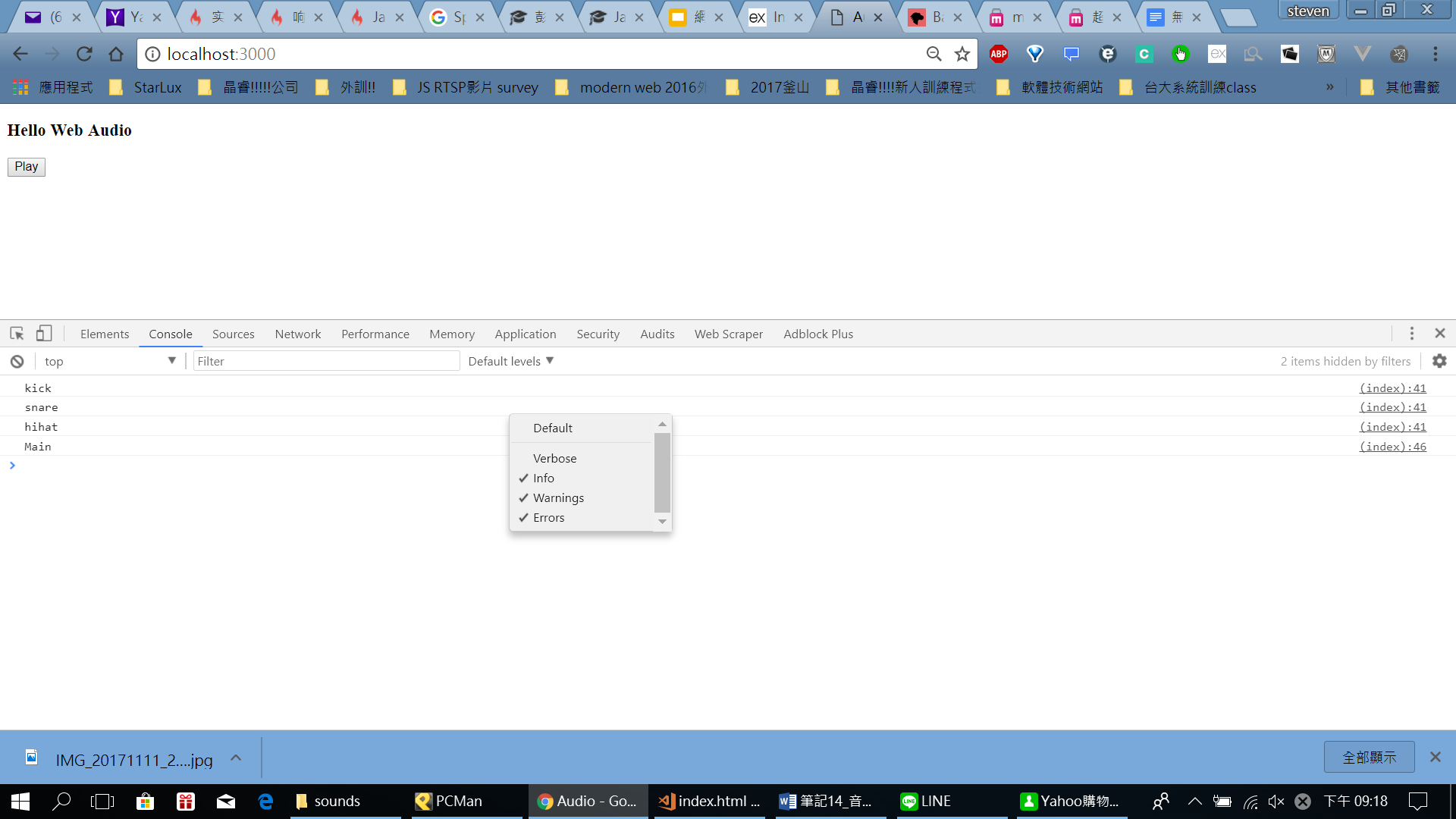
>>> 剛的console.log(“Main”)放的位置不對!

所以我們放個total和loaded來記錄這一切:





這樣重新印的Main時間點就是已經確定全部都載入完了!!!



記得:

***非同步相關的議題***都要非常小心!!!!