**芸工　情報処理基礎 第１２回　2024年7月5日　　西尾**

**【音声ファイル１】**

**０．前回の内容**

教材をインターネット上のニシオマトリックスから「プログラミングAサンプル10」をダウンロードして、デスクトップに作成したフォルダa2020prog内に入れた。ZIPファイルを展開（解凍、Unzip）して動作を確認した。VSCodeを開き、教材のフォルダをドラッグし、編集（エディット）を行い、JavaScriptのコードを変更し、プログラムを変更した結果をGoogleChromeの画面で確認した。

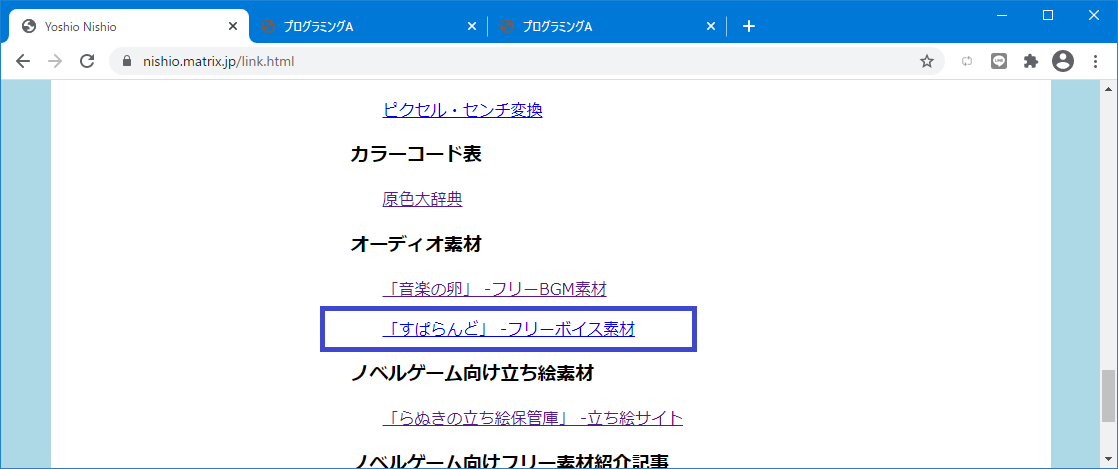
**１．今回の内容**

概要

前回は自分でアニメーションつまりパラパラマンガの画像を描き、利用することによりオリジナルソフトを作ってもらいました。

今回は、音声を扱います。画面を押すと音が出ます。教材「プログラミングAサンプル11」をダウンロードしてVSCodeを使用し、プログラムのJavaScriptのコードを変更します。

　なお今回使わせていただいたサウンドファイルはニシオマトリックスのリンク集の中の「すぱらんど」から参照できます。



**【音声ファイル２】**

**２．「プログラミングAサンプル１１」を表示する**

　まず、ニシオマトリックスに入る

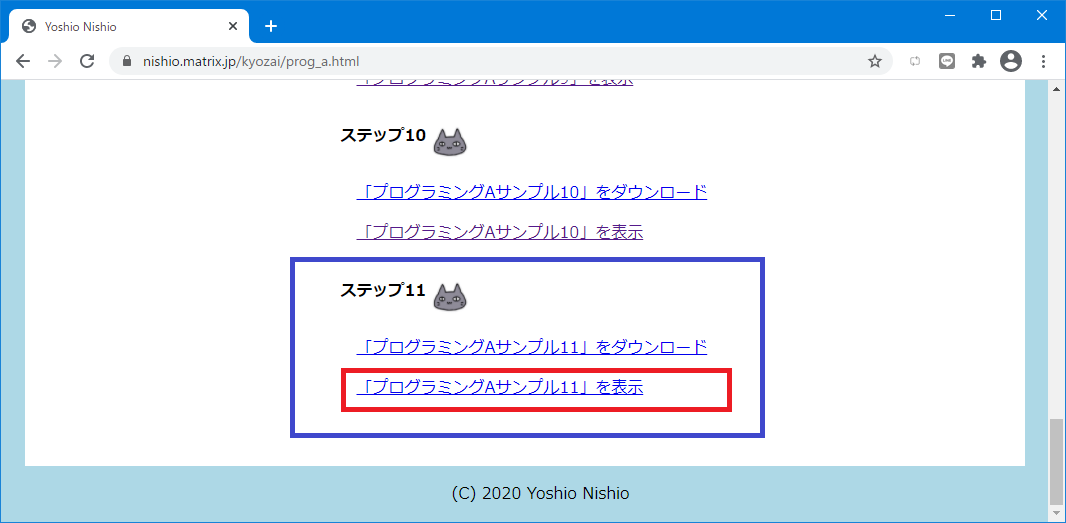
<https://nishio.matrix.jp/>



「教材」を押す



「プログラミングA」を押す



**「プログラミングAサンプル１１」を表示**を押す

左の画面ではマウスをダウンすると音声が出ます。右の画面ではマウスをダウンしたときのｘｙ座標が表示されます。

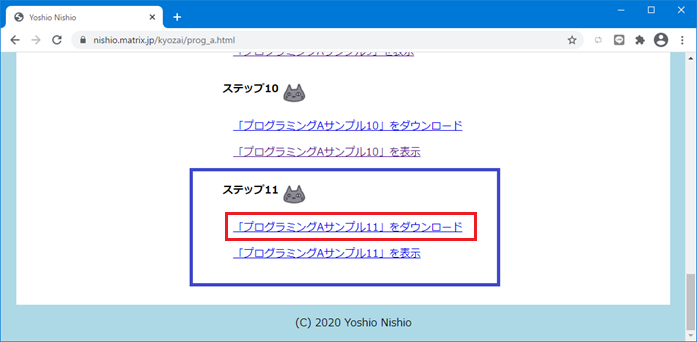


最後に✖で閉じておきましょう。（閉じていないと勘違いの元になります）

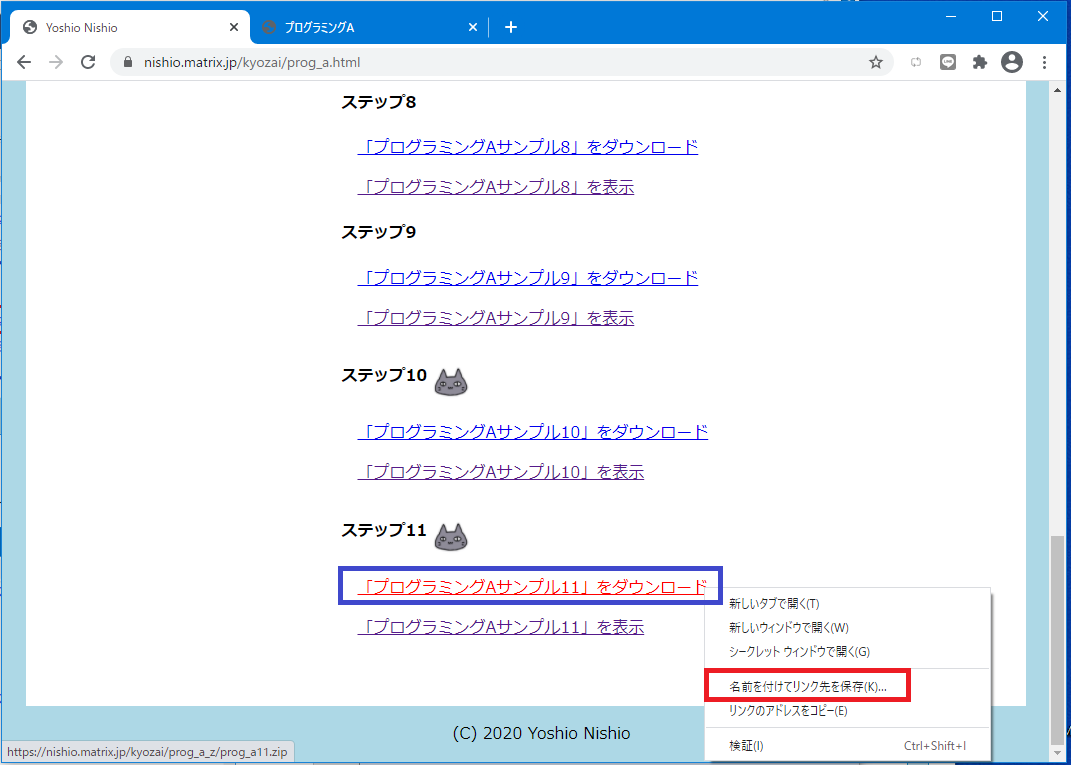
**【音声ファイル３】**

**３．「プログラミングAサンプル１１」のダウンロード**

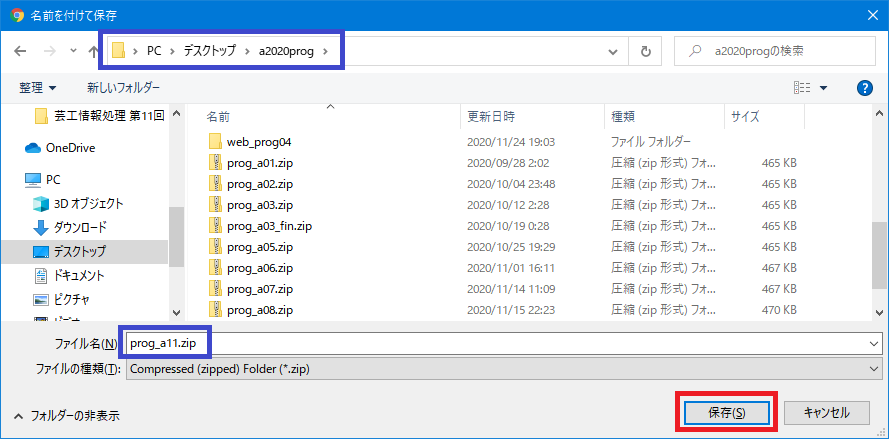
ニシオマトリックスの「教材」の「プログラミングA」に行きます



**「プログラミングAサンプル１１」をダウンロード**を右ボタンクリック／副ボタンクリックすると、プルダウンメニューが出てくるので**「名前を付けてリンクを保存」**をクリックし選択する



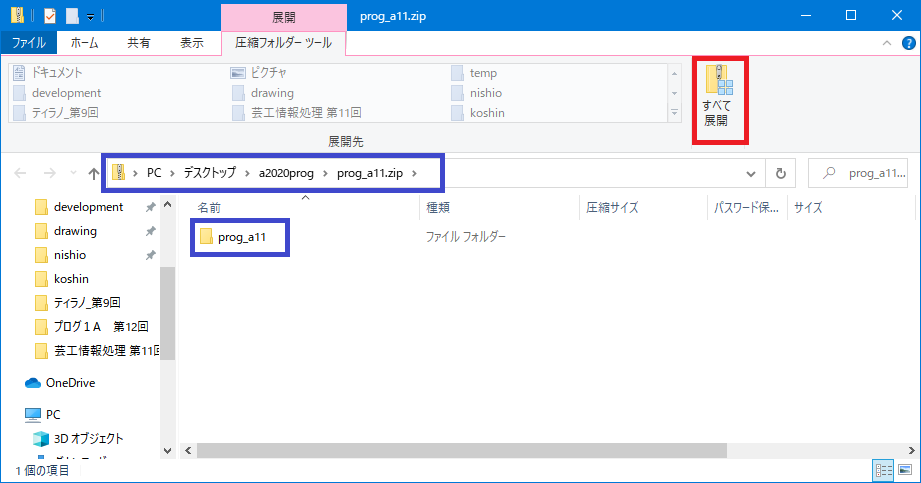
保存先の候補が表示される

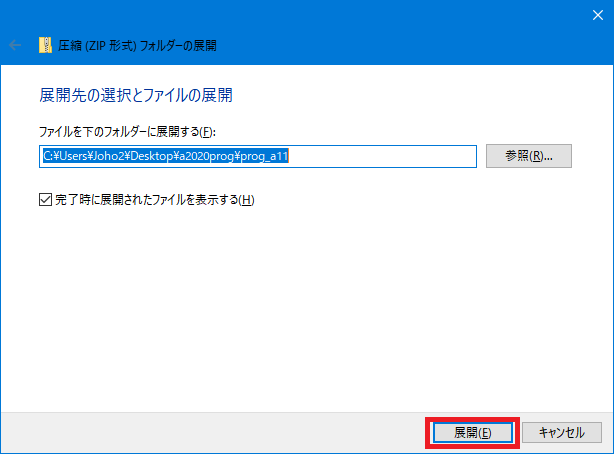


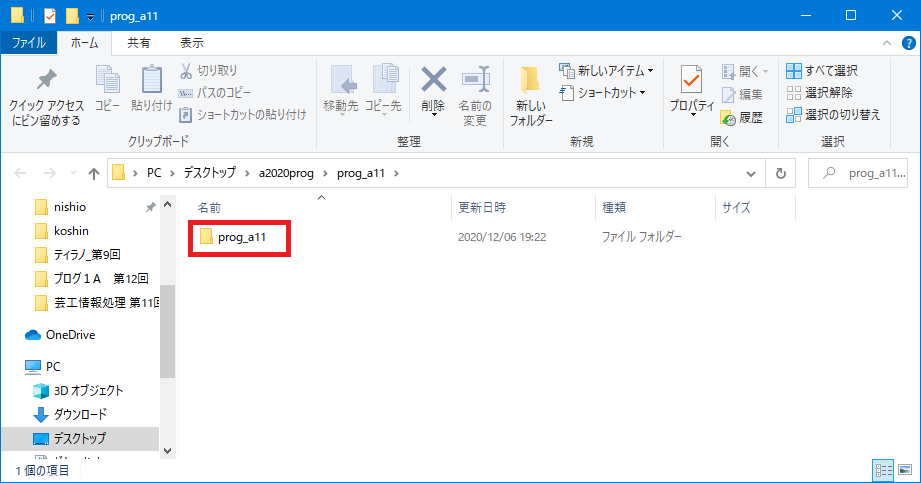
下図のようにウインドウの左下に出てくるのでクリックする



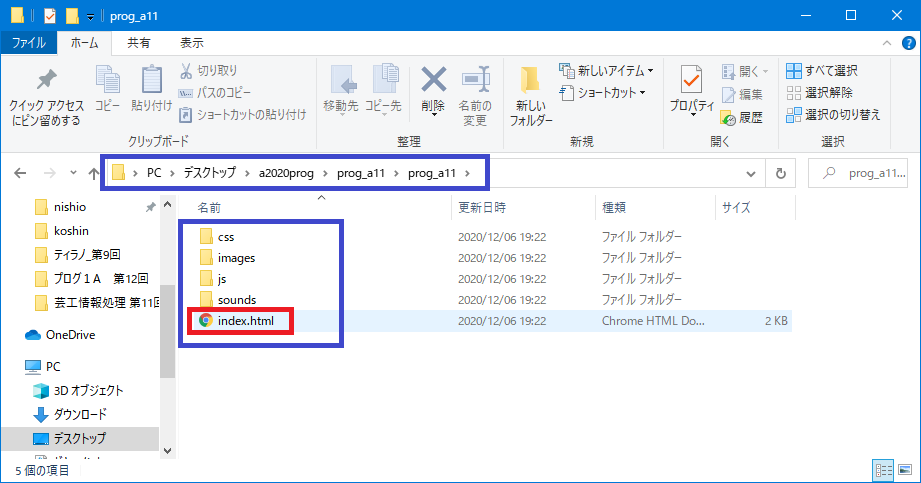
前回と同様にして展開（解凍）します。「すべて展開」を押します。







ちゃんと解凍したものがあるかチェックします

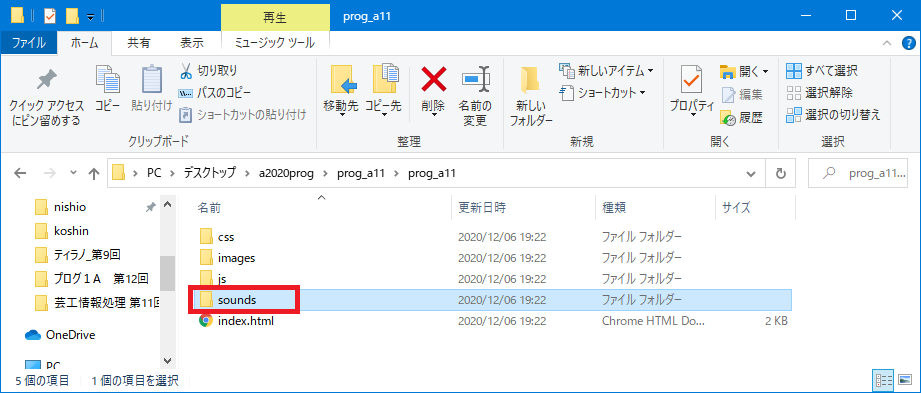


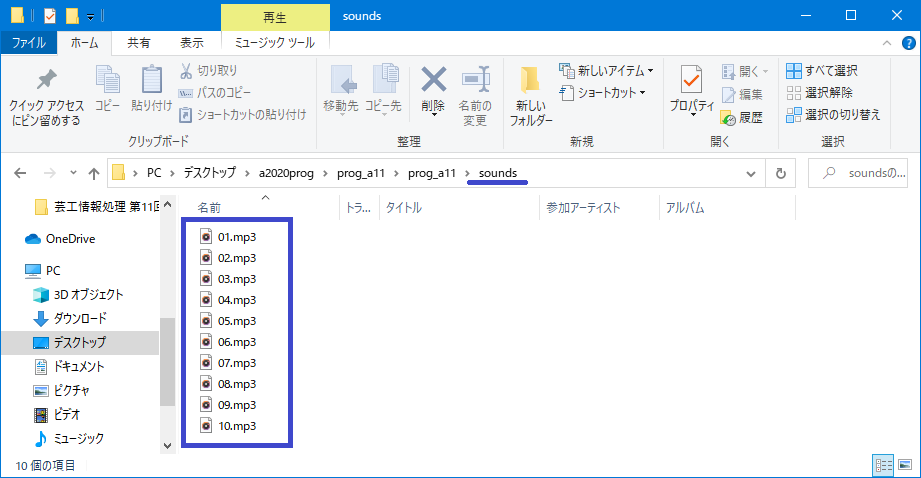
index.htmlをクリックして、ちゃんと入ったかを確認します



今回はここから始めて行きます。最初はどこを押しても同じ音がでます。マウスをダウンする場所により異なる音を出すようにして行きます。動物園の看板と各獣舎の案内音声のようなイメージで作って行きましょう。

また、prog\_a11フォルダの下のsoundsフォルダの中も見てみましょう。





１０個のmp3ファイル（音声ファイル）を用意しました。後で使います。

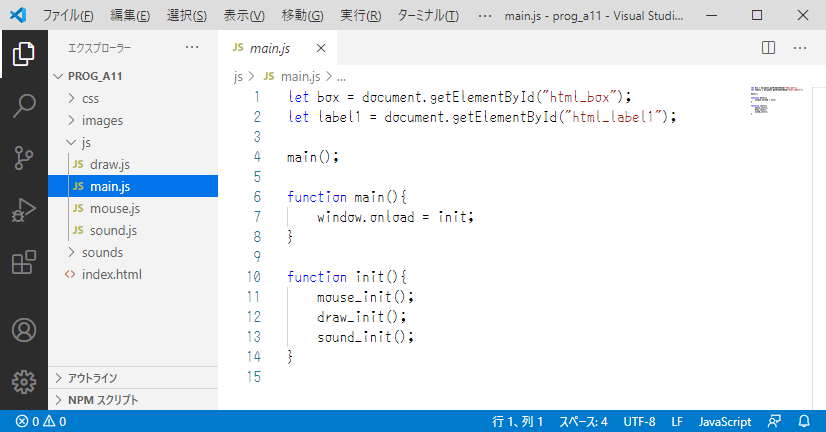
**【音声ファイル４】**

**４．プログラムの解説**

今回のプログラムは、構造が非常に簡単です。今まで雰囲気で動いていた部分を少しだけ細かく解説することにします。きっちり理解しなくてもOKです。雰囲気だけでも味わってください。

今回のプログラムの中心となるJSファイルは４つのファイルからできています。

メイン　main.jsファイル



main.jsの４行目にエントリーポイントがあります。

index.htmlからこのmain.jsを参照していて、実行できそうなのが、この４行目のmain();だけなのでこれが動きます。

６行～８行にこのmain関数の定義があります。

７行目にブラウザのウインドウが呼ばれたら、init関数が動くように登録されています。

10行～14行にinit関数（初期化関数）の定義があります。中身は３つのinit関数からなります。

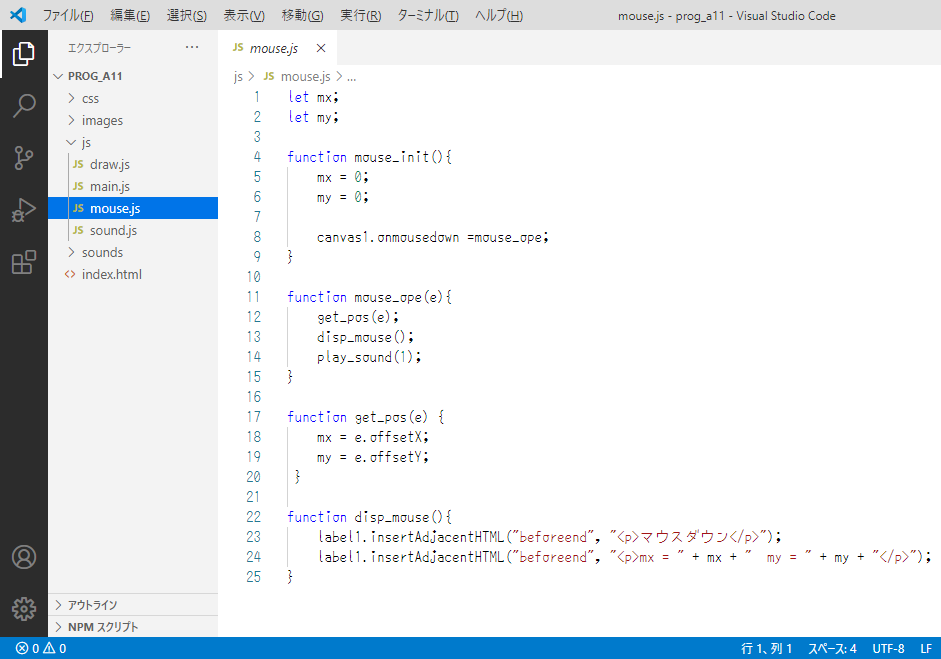
mouse\_init マウスの初期化、

draw\_init 描画の初期化、

sound\_init サウンドの初期化

です

マウス　mouse.js



マウスの関係はこのmouse.jsファイルにまとめてあります。

メインから呼ばれたmouse\_init関数の中身は

マウスの位置（mx, my）の初期値を（0, 0）に置き

キャンバス上でマウスをダウン（押す）したときmouse\_ope関数を発動させるように登録する

です。

mouse\_ope関数（マウスオペレーション関数）は11行～15行で、

get\_pos マウスをダウンしたときのｘｙ座標を得る

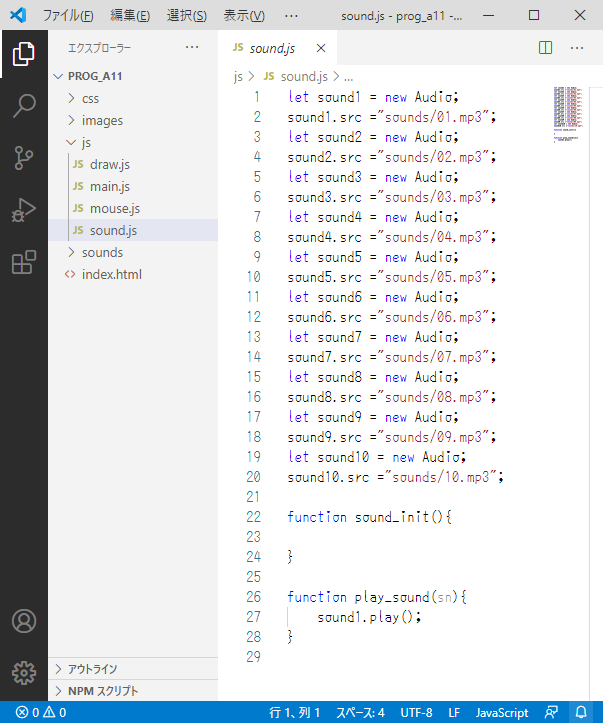
disp\_mouse マウスの情報を表示する

play\_sound 音を再生する

の３つから成ります

14行が　play\_sound(1)　のように｟｠の中に１という引数（ひきすう）があるのは、１曲目（ではなく1番目）の音をかけてくれという意味です。ということは、この１が３に変われば3番目の音をかえてくれという意味になります。

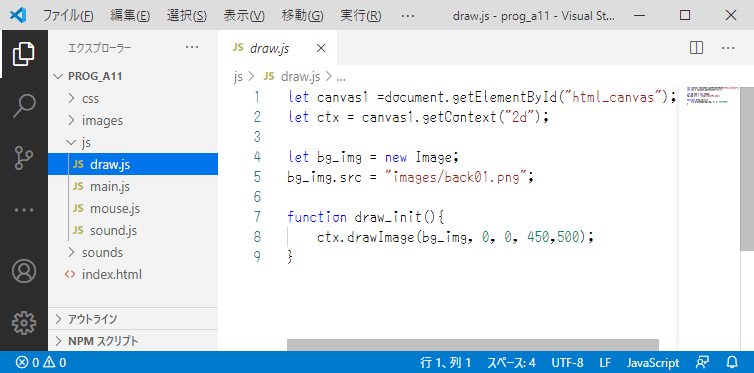
サウンド sound.js



22行～24行を見ると、sound\_init関数（サウンド初期化）がありますが、何もしない関数です。何もないということを明示しています。

26行～28行にplay\_sound関数（サウンド再生）があります。引数でsnがあります。サウンドナンバの意味です。中身の27行を見ると サウンドナンバsnに無関係に,サウンド１を再生しています。後でここを改良します。

描画　 draw.js



7行～9行のdraw\_init関数（描画初期化関数）はバックグラウンドの画像を描画します。

前回までは背景が白色でした。今回は背景に画像を入れています。imagesフォルダの中のback01.pngを使っています。もちろんこの画像を他の画像に入れ替えれば背景が変わります。今回はアニメーションを入れていないため、描画するのは最初の1回で十分なので、初期化時に描いています。

**【音声ファイル５】**

**５．サウンドの変更**

sound.jsファイルのプログラムを変更して行きます。

青の枠内は音の宣言です。

変数sound1～sound10を、soundsフォルダの下のサウンドファイル01.mp3～10.mp3に対応させています。

赤い枠内はplay\_sound関数（サウンド再生関数）です。

今のところ引数snが何であるか無関係に、27行のsound1.play関数が実行されます。

この関数はライブラリで宣言されており、自分で定義する必要がありません。

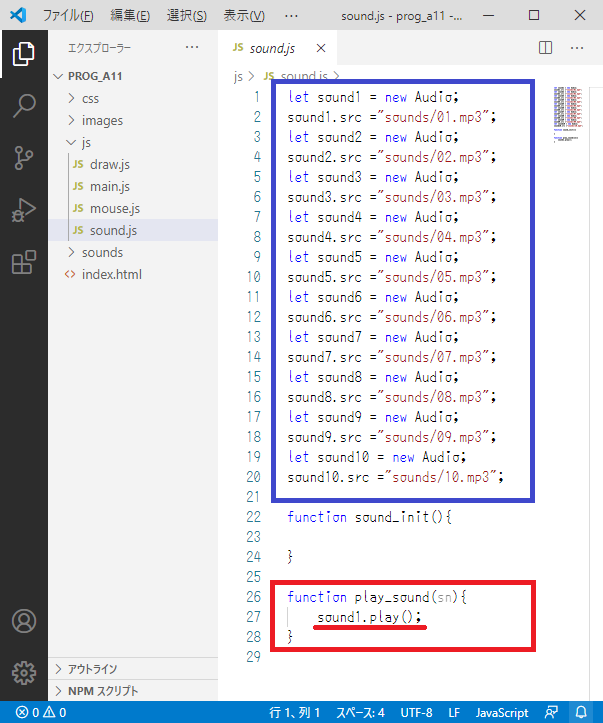
**（１）実験１**

27行目のsound1.playはサウンド1の再生を意味します。変更して実行してみましょう。

27行の変更 **sound1.play ➞ sound2.play**

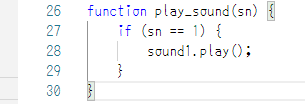
**マウスをダウンすると、出てくる音が変わったことが確認できます。**

sound10まで試してみてください

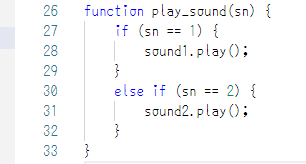


このままだと、音を変えるのが不便なので、引数snの値で音が変わるようにします。

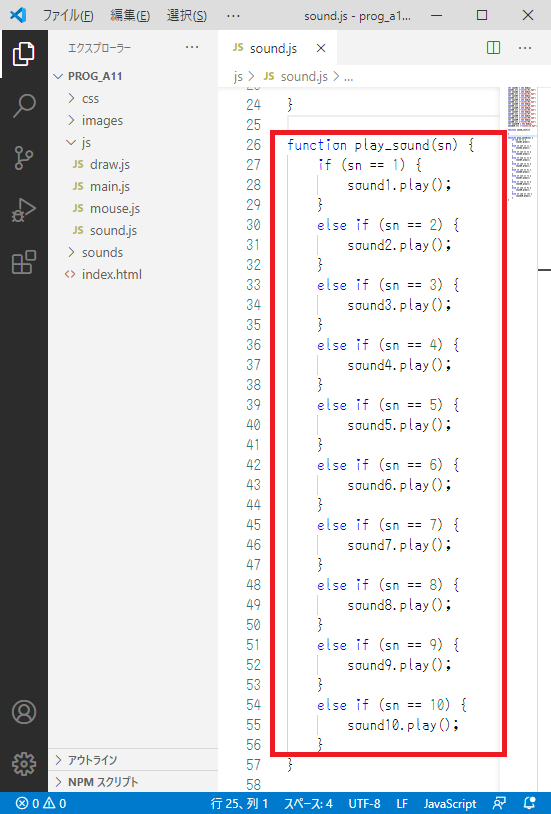
もしもsnが１だったらsound1.playを実行は、プログラムでは



さらに、もしもsnが２だったらsound2.playを実行は、プログラムでは

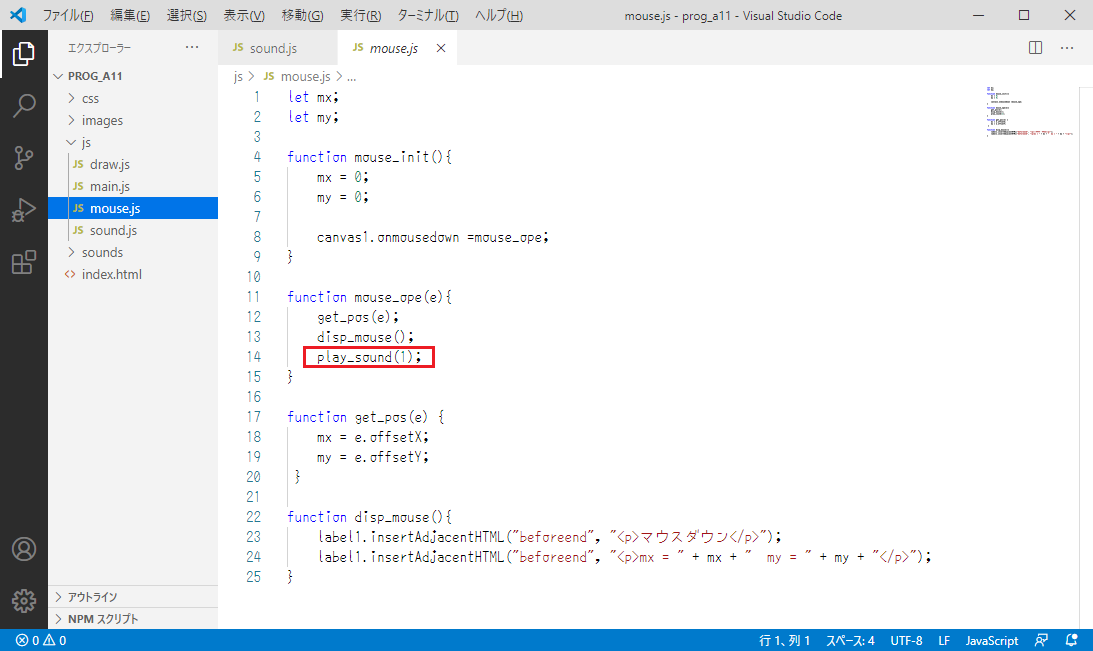


となります。コピー＆ペーストをうまく使いsnが10だった場合までのプログラムにします。

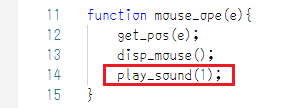


まず実行して確かめましょう。音の変化はありませんね。変化させるには呼ぶ方の引数を変える必要があります。

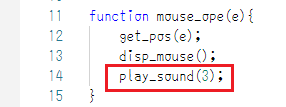
mouse.js ファイルの14行目



拡大すると



14行の引数を変えます。もちろん、変えた後はファイルの保存をいちいちしなくてはなりませんね



**（２）実験２**

この14行目の**play\_sound関数の引数を1～１０まで変化させ実行して音の変化を確認しましょう**

**【音声ファイル６】**

**６．マウスダウンとサウンドの変更**

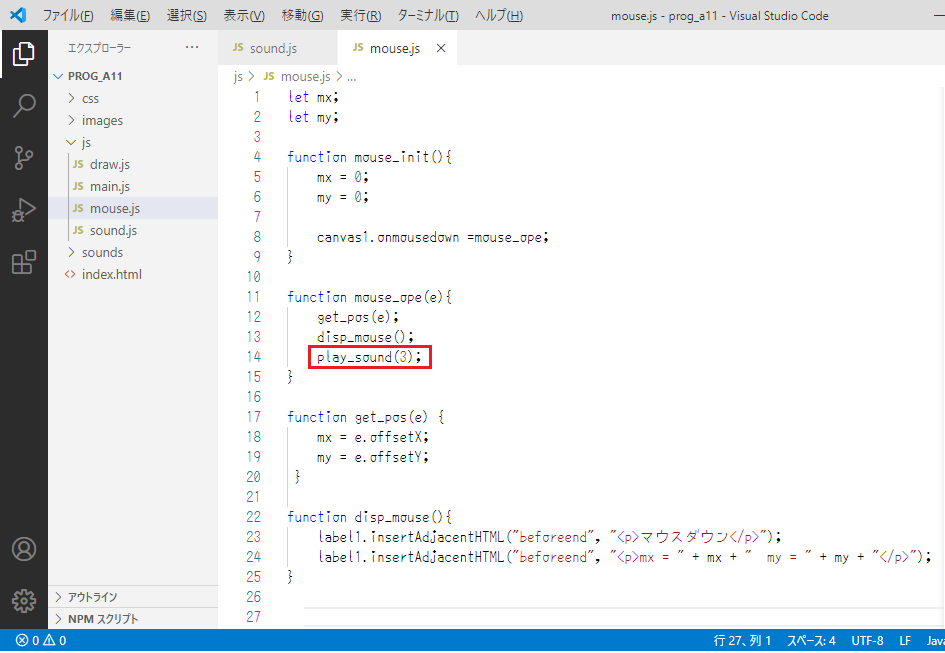
　マウスダウンにより出てくる音が変わりましたが、マウスでどこを押しても同じ音でした。

動物園の案内板にするには、押す場所により解説の音声を変える必要があります。

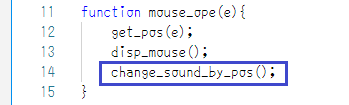
マウスを押す場所によって、違う音を出すようにします。

mouse.jsファイルの変更

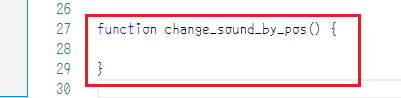
　関数が長くなるのを防ぐため、関数の構成を変えます

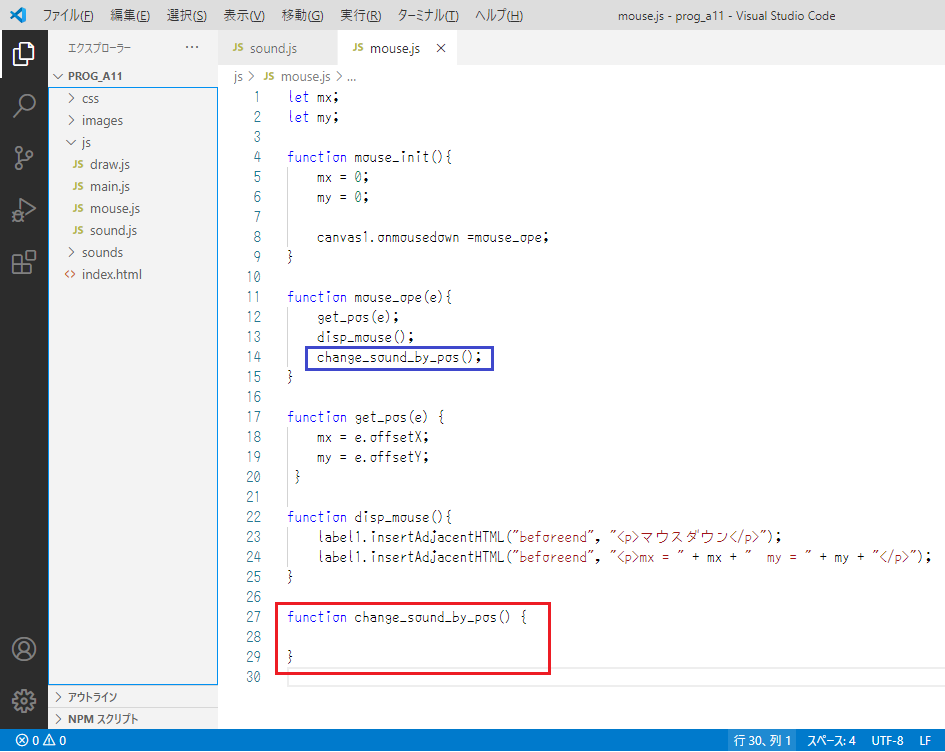


14行を書き換えます



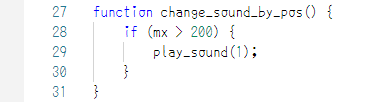
27行目から新しい関数を作ります



次に28行から書き変えていきます

**（１）実験３**

mx > 200 のとき、サウンド１がなる



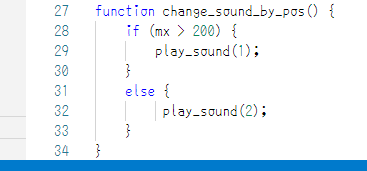
マウスを下図のような範囲でダウンして、音を確認してください



**（２）実験４**

mx > 200 のとき、サウンド１が鳴る

その他のとき、サウンド２が鳴る



マウスを下図のような範囲でダウンして、音を確認してください



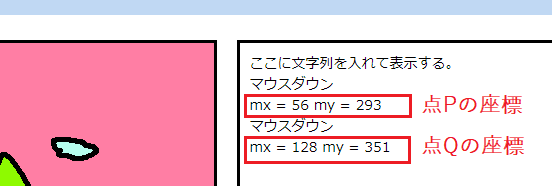
**（３）実験５**

画像の中のある部分でマウスダウンすると音が鳴るようにします

まず、マウスで指定したい範囲を覆う四角の左上の点Pと右下の点Qを押します



右の画面にマウスの座標が表示されます



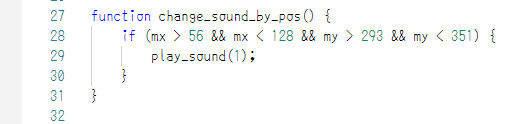


条件は



です

これをプログラムにします



if文の条件式には独特な演算子が使われます

&&　は　Andや「かつ」と呼ばれます

比較演算子を示します



論理演算子

演算子 意味

＆＆ かつ and

｜｜ または or

！ 否定 not

**if (mx > 56 && mx < 128 && my > 293 && my < 351)**

の部分は

**もしも　ｍｘ＞５６　かつ　ｍｘ＜１２８　かつ　ｍｙ＞２９３　かつ　ｍｙ＜３５１　ならば**

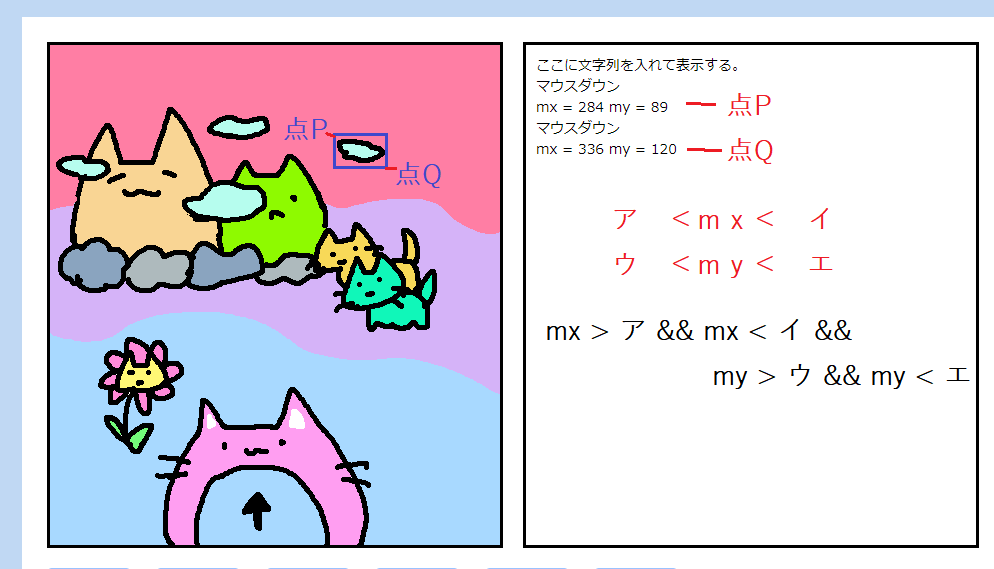
と読みます

実行してみましょう。ネコ花のところでマウスをダウンしたときだけ音がでますね。

確認しましょう。

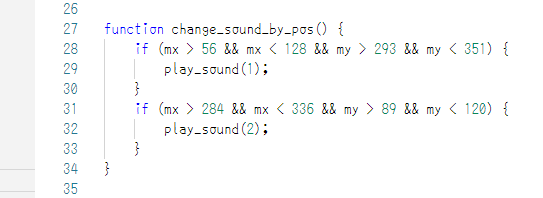
**（４）実験６**

画像の中のある部分でマウスダウンして音を出す



上の図のアイウエに入る数値を考えてください

答えをプログラムにすると下図のようになります



**【音声ファイル７】**

**７．課題**

　前節では2か所の矩形（四角形）の範囲を指定して、その中でマウスのボタンをダウンしたとき、固有の音を出すようにしました。さらに5カ所程度、矩形の範囲を指定し、固有のサウンドが出るようにプログラムを改良しましょう。

自分で矩形の範囲を探り、mouse.jsファイルの33行目の後に命令を挿入して、動くプログラムにしてください。

画像をいつものようにWordファイルに張り付け、さらにmouse.jsファイルも改良した部分が分かるように付けてください。

提出例を参考にして、

情報処理基礎 　日付

学籍番号　名前

実行時の画像

mouse.jsファイルのリスト

感想・コメント（3行程度）

を学情システムに提出してください

提出例

情報処理基礎 2024年7月5日

0123456　西尾吉男



let mx;

let my;

function mouse\_init(){

    mx = 0;

    my = 0;

    canvas1.onmousedown =mouse\_ope;

}

function mouse\_ope(e){

    get\_pos(e);

    disp\_mouse();

    change\_sound\_by\_pos();

}

function get\_pos(e) {

    mx = e.offsetX;

    my = e.offsetY;

 }

function disp\_mouse(){

    label1.insertAdjacentHTML("beforeend", "<p>マウスダウン</p>");

    label1.insertAdjacentHTML("beforeend", "<p>mx = " + mx + "  my = " + my + "</p>");

}

function change\_sound\_by\_pos() {

    if (mx > 56 && mx < 128 && my > 293 && my < 351) {

        play\_sound(1);

    }

    if (mx > 284 && mx < 336 && my > 89 && my < 120) {

        play\_sound(2);

    }

ここにプログラムを追加する

}

感想・コメント　サウンドが・・・

**【音声ファイル8】**