**芸工情報処理基礎　第5回　2024年5月17日　　西尾**

**【音声ファイル１】**

**注意　この授業は第1回目から指示通りに学習・実行をしないと動きません。**

**途中からこの授業をとる人は、第1回目の内容から順番に進めてください。**

**途中で飛ばしたりすると、全く動作しません。**

**（少しは動くが、表示がまともに出なくなる）**

**必ず、第1回から飛ばさず順にやってください。**

**この注意に従わず、やっていない場合、サポートできません。**

**また、プログラミングのコードは文章ではありません。**

**動作を確かめながら、少しずつ積み上げるものです。**

**一気に打ち込んで、動かないのは当然です。気を付けましょう。**

**１．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「プログラミングサンプル３」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。

**２．今回の内容**

概要

　前回のプログラムを理解し、改良を加える。

教材「プログラミングサンプル３からの続き」をダウンロードしてVSCodeを使用し、プログラムのJavaScriptのコードを変更する。

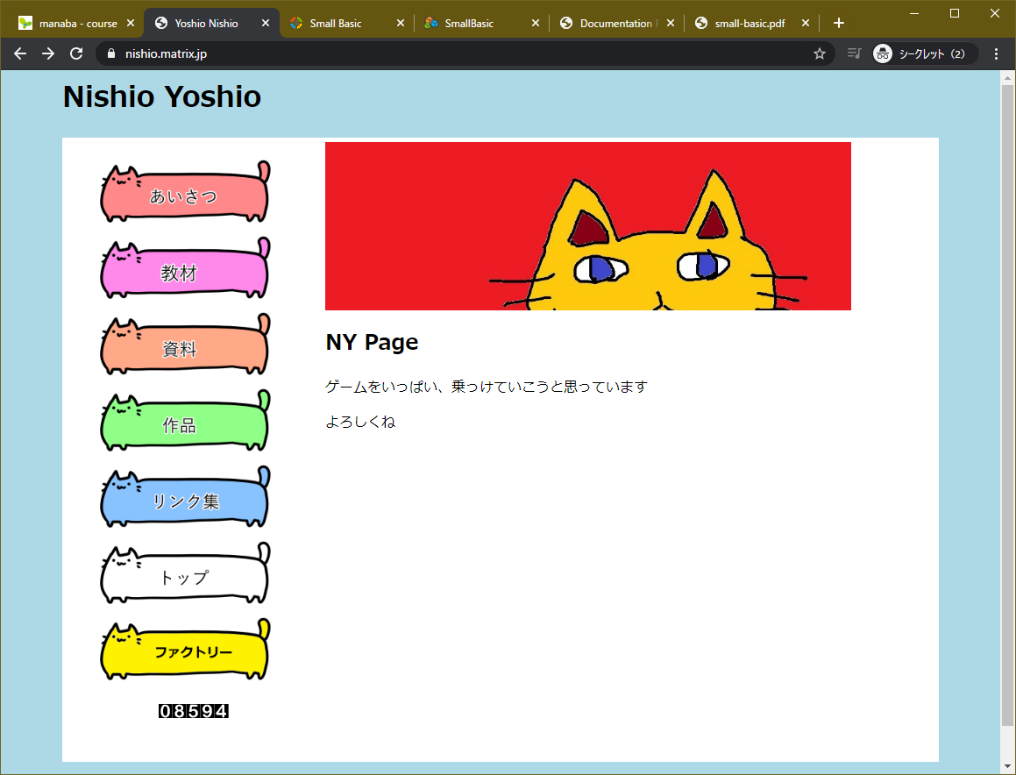
前回のプログラムではマウスを押した時（マウスダウン）のｘｙ座標を取得し表示を行いさらに、そのｘｙ座標のところに図形を描画したが、単色だった。今回は色パレットを表示し、取った色で四角を描画できるように改良を施す。

**【音声ファイル２】**

**３．前回の「プログラミングAサンプル3」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

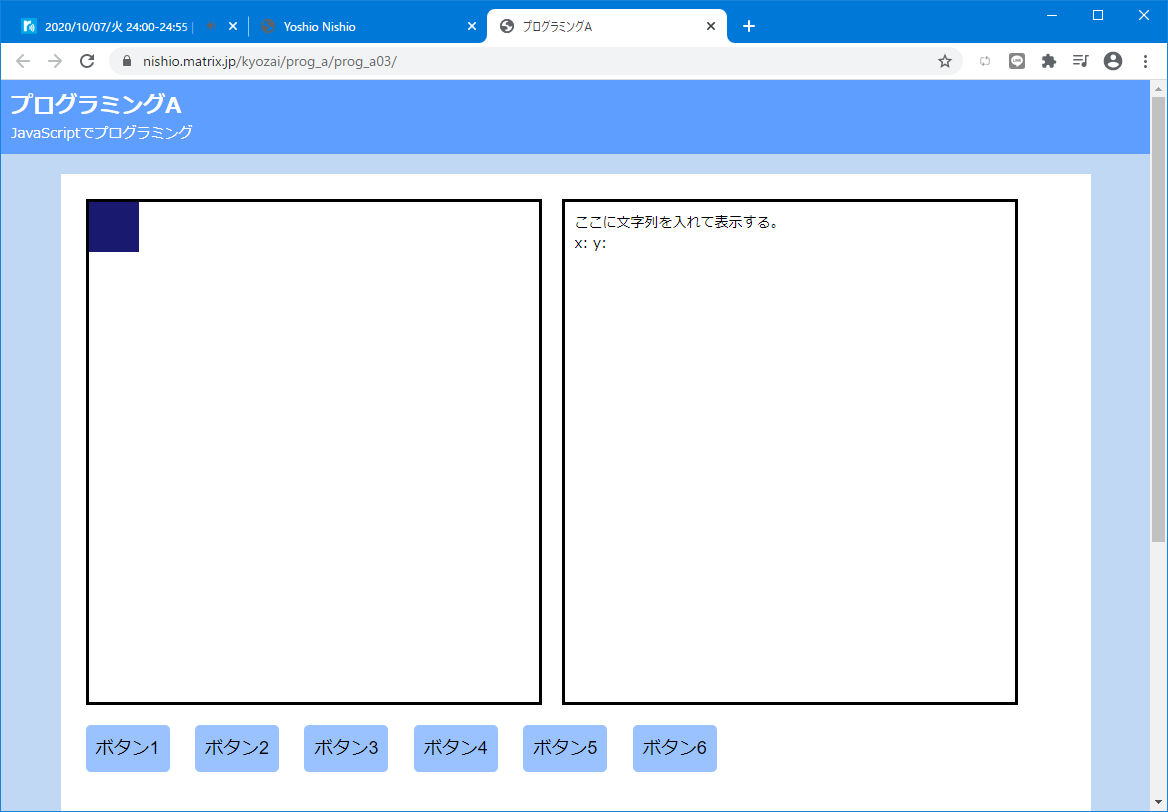


「プログラミングA」を押す



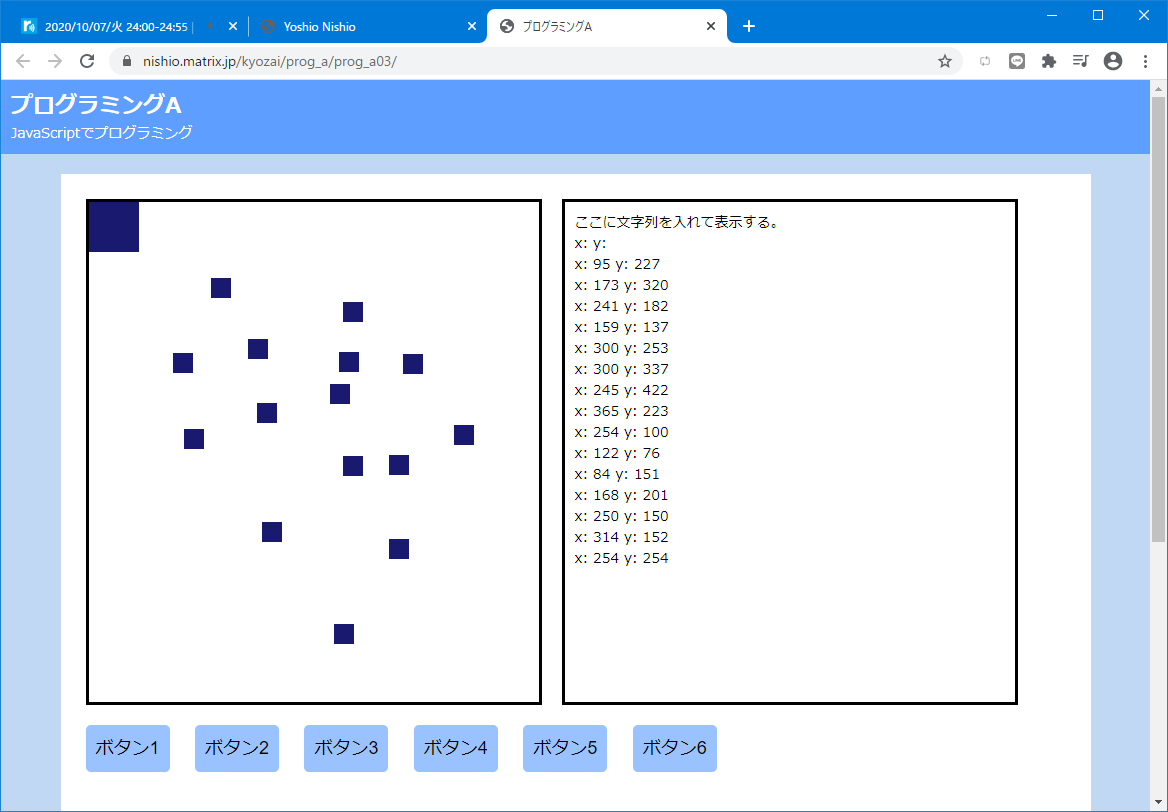
**「プログラミングAサンプル3」を表示**を押す



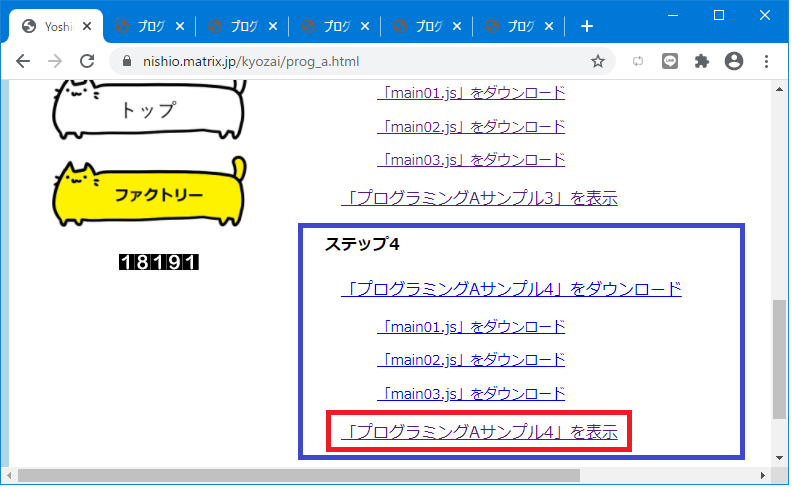


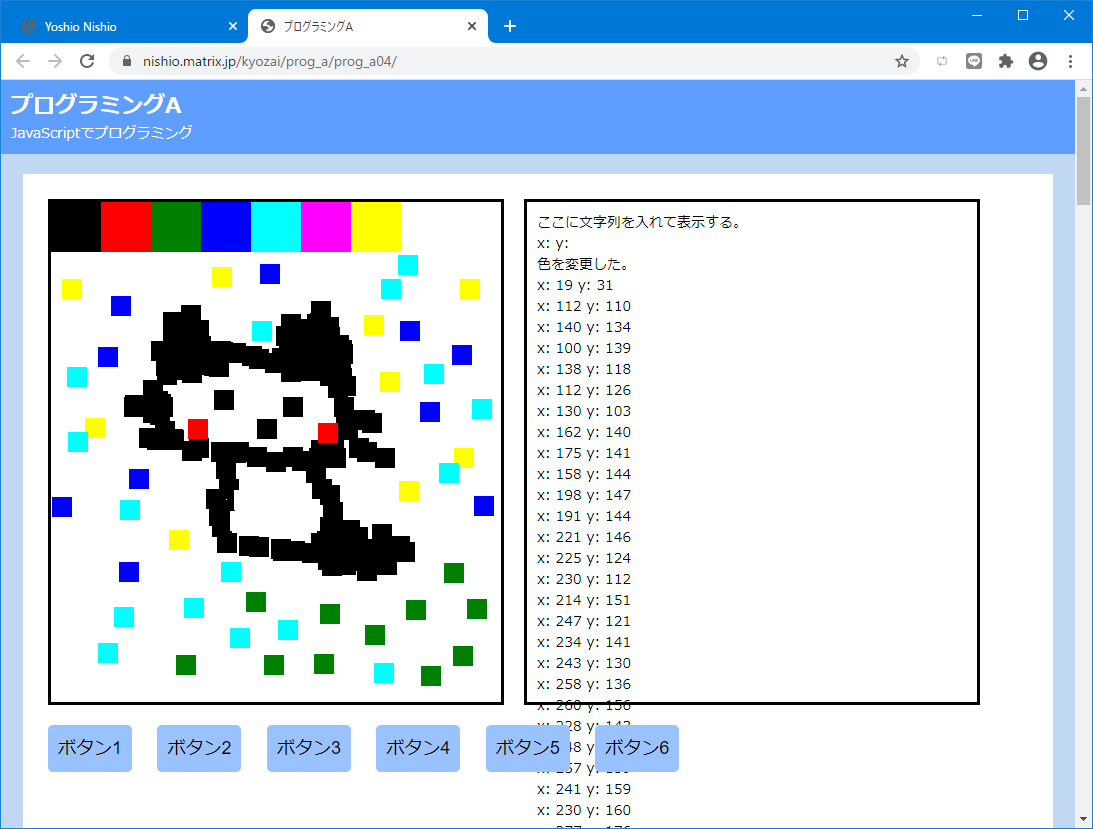
今回は左の四角のキャンバスでマウスをクリックし、右の方では、その時のマウス座標を出していきます。

クリックしてみてください。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）



**「プログラミングAサンプル４」を表示**を押す



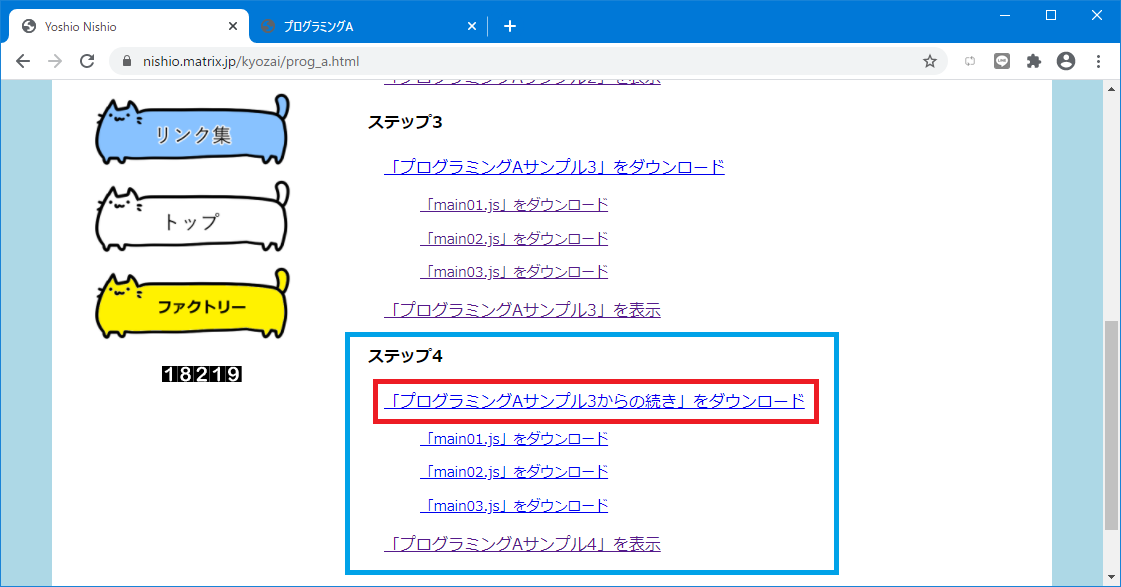


色パレットをクリックして、色をとり、絵の枠内をクリックしてみてください。前の例との違いが理解できたでしょうか。最後に✖で閉じておきましょう。（閉じていないと勘違いの元になります）

**【音声ファイル３】**

**４．「プログラミングAサンプル３からの続き」のダウンロード**

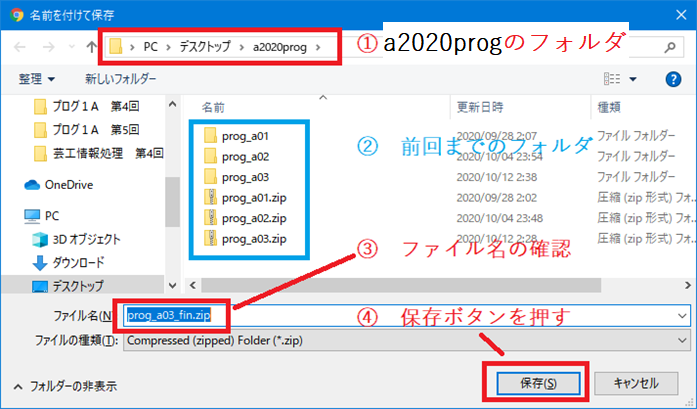
ニシオマトリックスの「教材」の「プログラミングA」に行きます



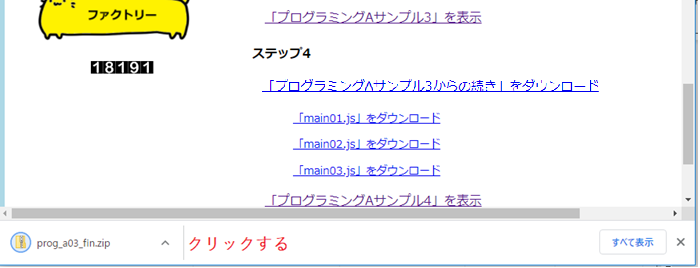
**「プログラミングAサンプル３からの続き」をダウンロード**を右ボタンクリック／副ボタンクリックすると、プルダウンメニューが出てくるので**「名前を付けてリンクを保存」**をクリックし選択する



保存先の候補が表示される

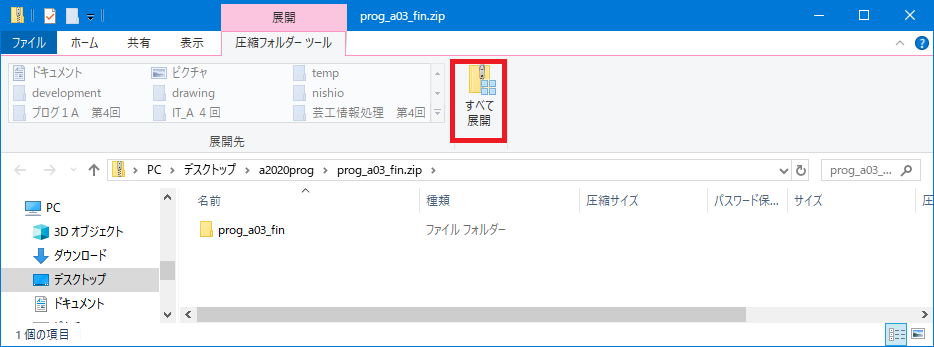


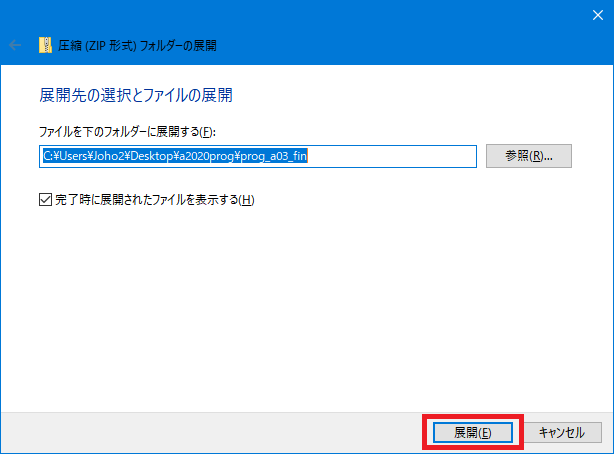
下図のようにウインドウの左下に出てくるのでクリックする

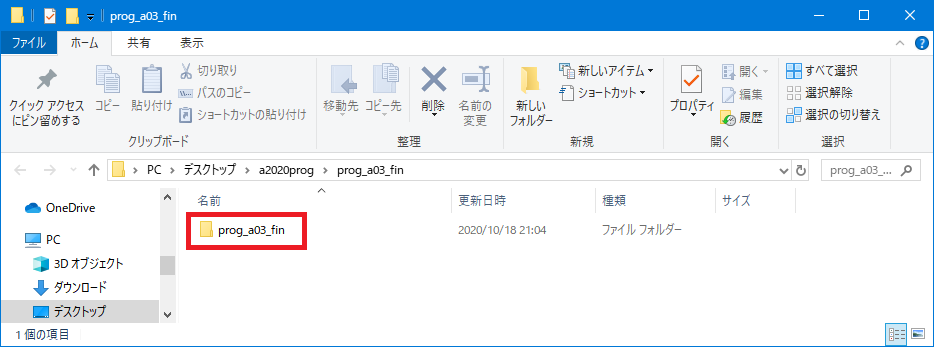


上図のように左下にｚｉｐファイルの表示が出ない場合は、デスクトップのa2020progフォルダからダブルクリックして入り、prog\_a03\_fin.zipを見つけてダブルクリックしても同様なことができます。

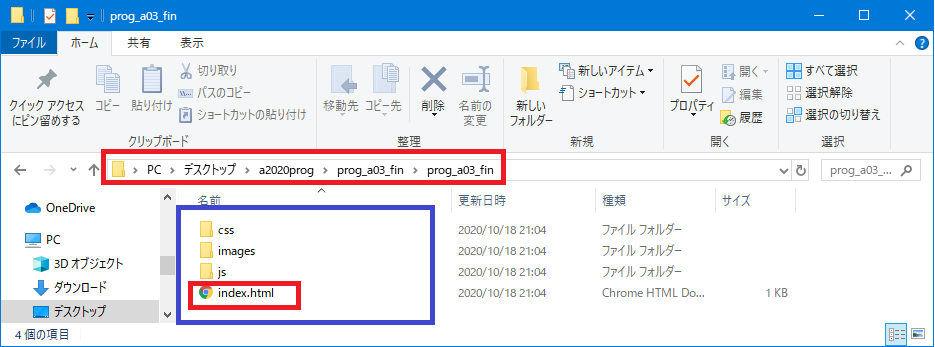
前回と同様にして展開（解凍）します



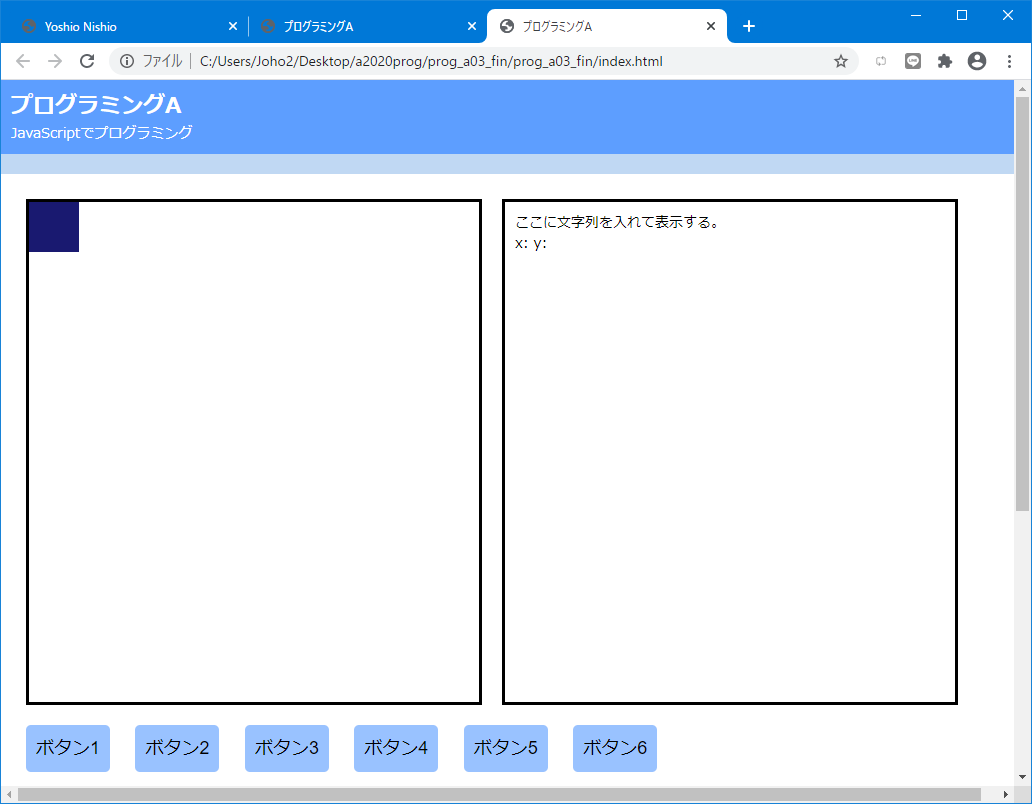




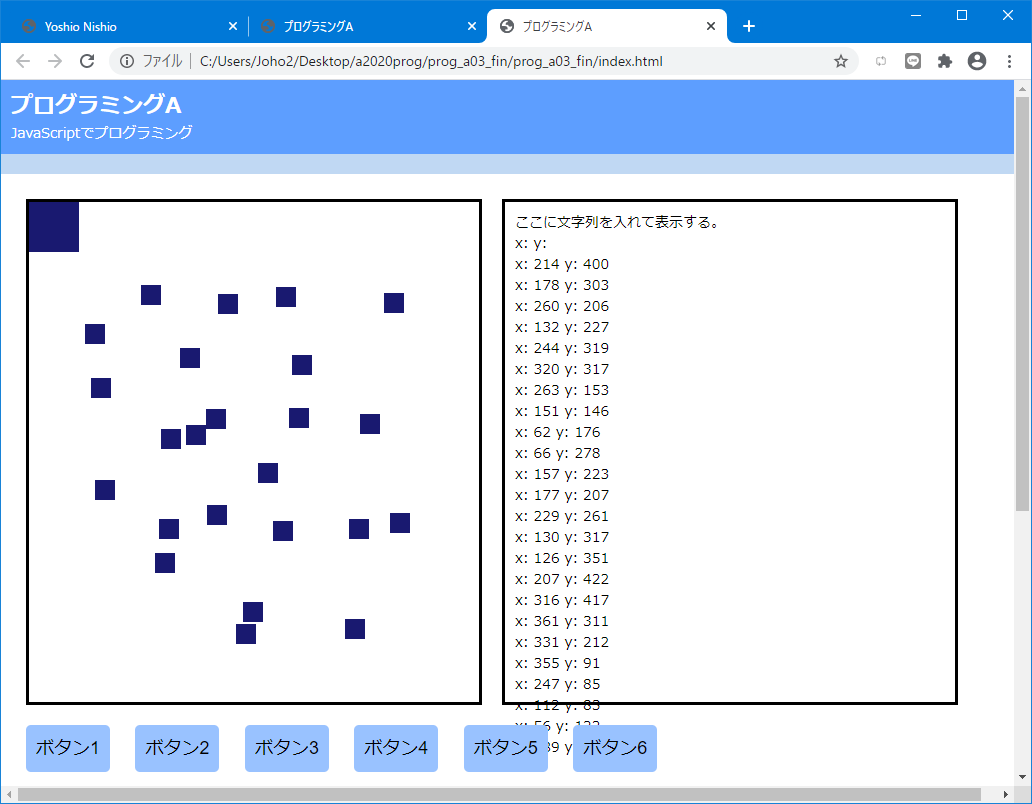
ちゃんと解凍したものがあるかチェックします



index.htmlをクリックして、ちゃんと入ったかを確認します



マウスで押してみると単色しか出ません。前回のプログラムの最終形しか入っていないからです。



今回はここから初めて、

[「プログラミングAサンプル4」を表示](https://nishio.matrix.jp/kyozai/prog_a/prog_a04)

で見たプログラムに発展させていきます。

**【音声ファイル４】**

**５．VSCodeでjsファイルを編集する準備**

　JavaScriptファイルのことを略してjsファイルと言います。

VSCodeを立ち上げて、教材からダウンロードしたjsファイルを編集する準備をしましょう。

（１）VSCodeを立ち上げる

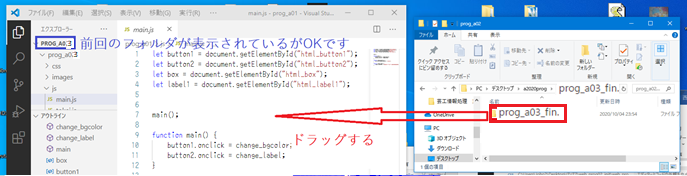
　VSCodeを立ち上げる



VSCodeを立ち上げます



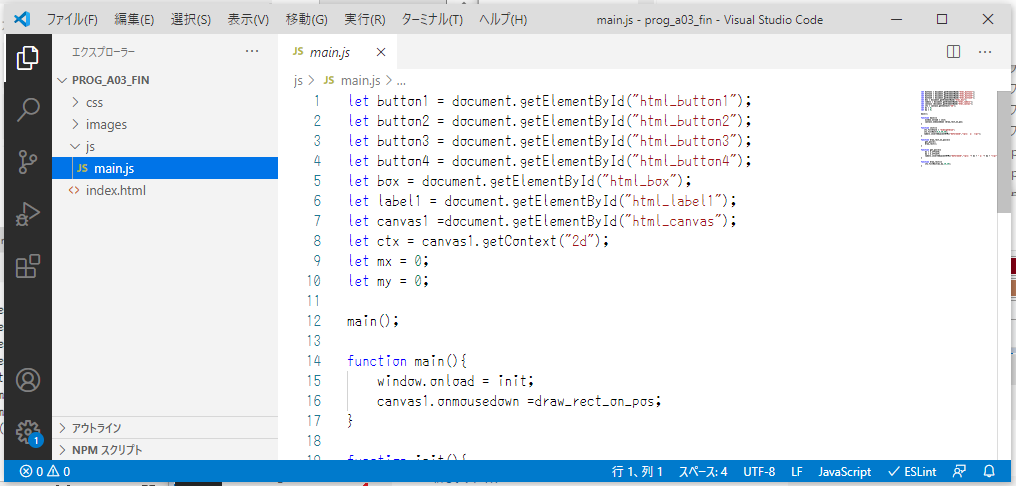
「prog\_a03\_fin」フォルダ（Windowsでは２重になっているので、下の方のフォルダ）をVSCodeのウインドウに向かってドラッグする。



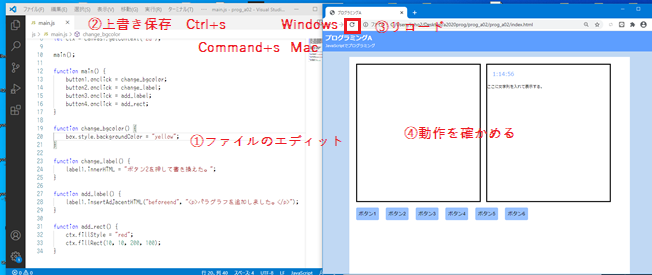
「エクスプローラー」を表示させる。「main.js」をクリックする



main.jsファイルを編集する準備が出来ました。



**画面の配置とプログラミングの流れ　スムーズに作業が進むように配置を整えましょう**



プログラミングとデバッグのながれは次のような①~③を繰り返すことです。

①　VSCodeを編集（エディット）する。Ctrl＋ｓキーでファイルを保存する。

②　Google Chromeでリロードする。

③　ボタンを押すなどして、動作を確かめる。

**【音声ファイル５】**

**６．main.jsファイルのプログラムの詳細**

前回のプログラムのmain.jsを、詳しくみてみましょう

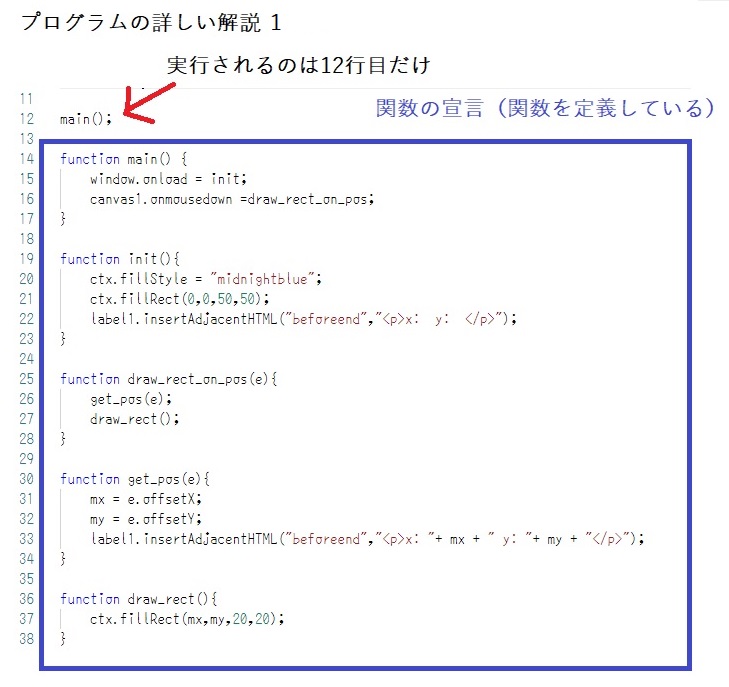
細かなところよりも大まかな概要をつかむようにしましょう。誰も細かなことまで知っている人はいませんから。

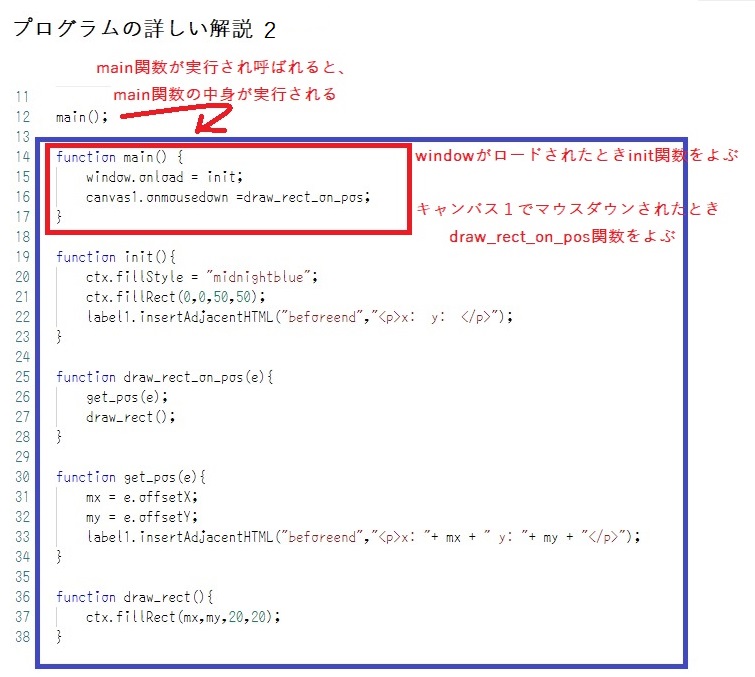
車に乗る人は、エンジンの細かな仕組みを知らなくても乗っていますね。それでいいのです！

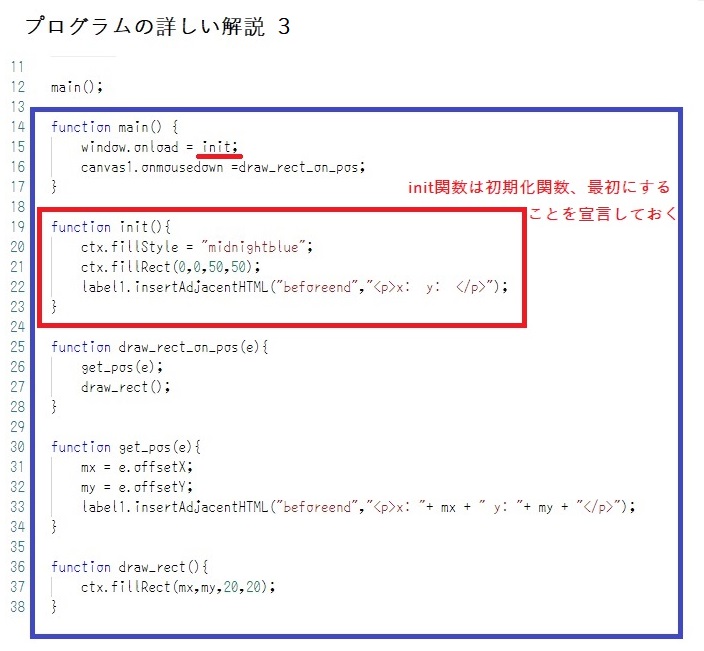
ただ、ガソリンやオイル、タイヤ、そして交通ルールぐらいは知らないとだめですが。

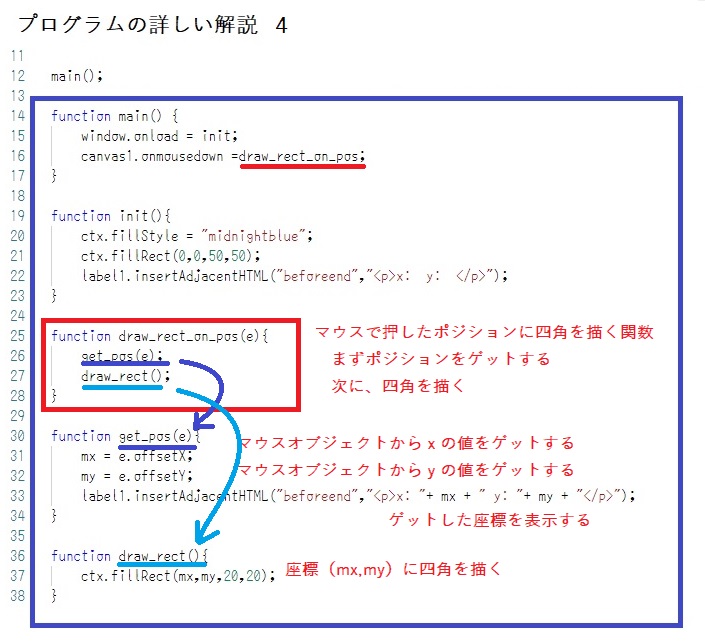
VSCodeでmain.jsファイルの中身を見てみましょう。

1回の説明で分かれば、天才です。









**【音声ファイル６】**

**７．main.jsファイルのプログラムの改良と実行１**

**プログラムの改良を行って行きます。**

プログラミングで大事なことの一つに、効率が良くなる画面の配置があります。

ちゃんと出来ていますか？

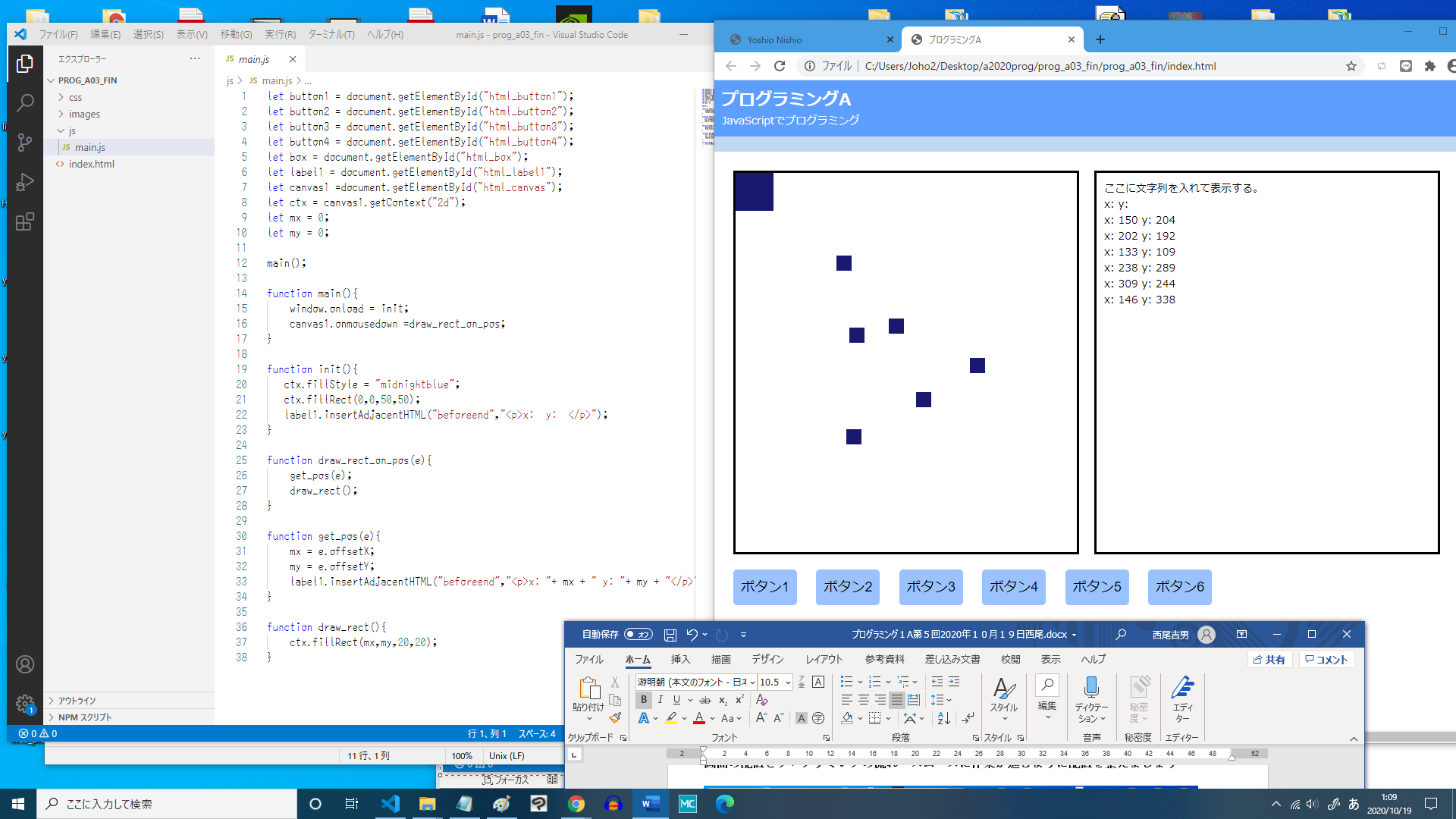
VSCodeのmain.jsの画面と

GoogleChromeでデスクトップの下のa2020prog/prog\_a03\_fin/prog\_a03\_fin/index.htmlを

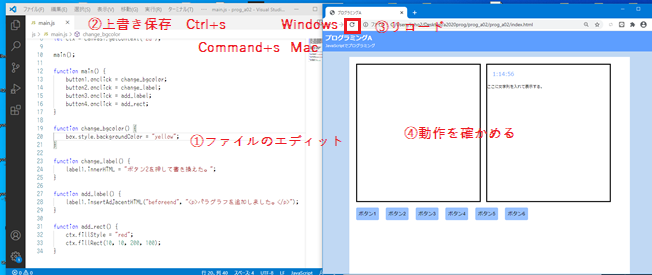
ダブルクリックして開いた画面を

左右に並べています。

教材のWordの画面も下に待機していますね。



前々節を思い出しましょう。



プログラミングとデバッグのながれは次のような①~③を繰り返すことです。

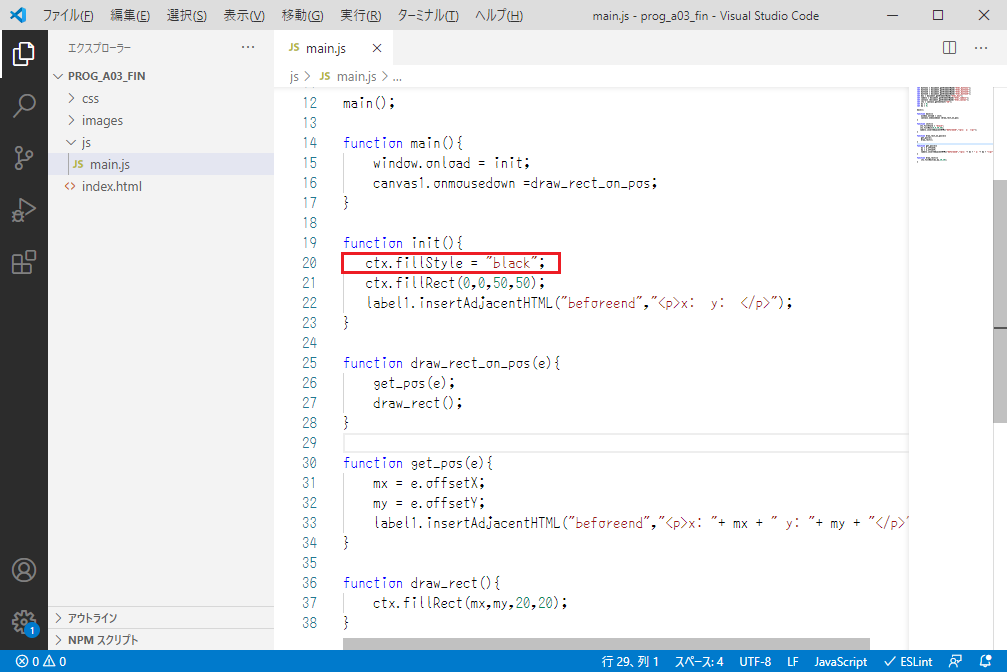
①　VSCodeを編集（エディット）する。Ctrl＋ｓキーでファイルを保存する。

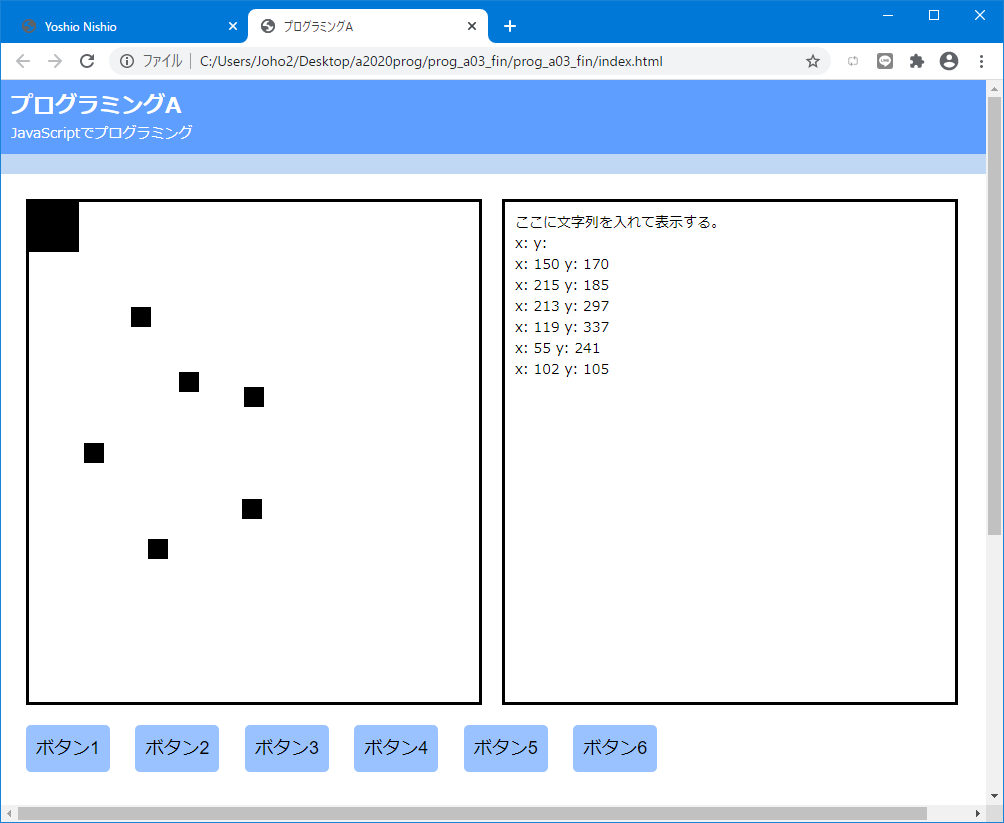
②　Google Chromeでリロードする。

③　ボタンを押すなどして、動作を確かめる。

VSCodeでmain.jsファイル20行目の色指定をblackに変更します。

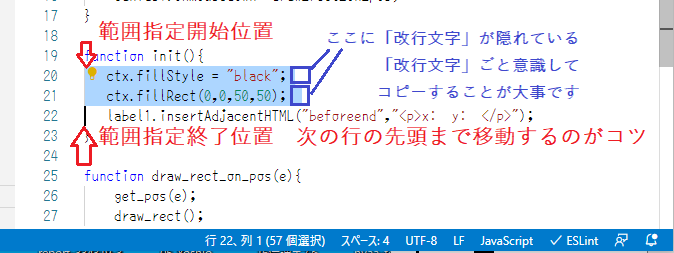
編集したものを保存し、実行してみましょう。





コピー＆ペーストのうまいやり方を伝授します。他でも使えるので覚えてください。

コピーする範囲指定がコツです。横に伸ばすひとが多いですが、縦に3行上から下にまっすぐ範囲を指定します。



この後Windowsの人は

　Ctrl+cを1回、Ctrl＋ｖを1回（これは捨てる分です）の後にCtrl＋ｖを回数分行います

Macの人は、Ctrlキーの代わりにCommandキーで同様に行います。

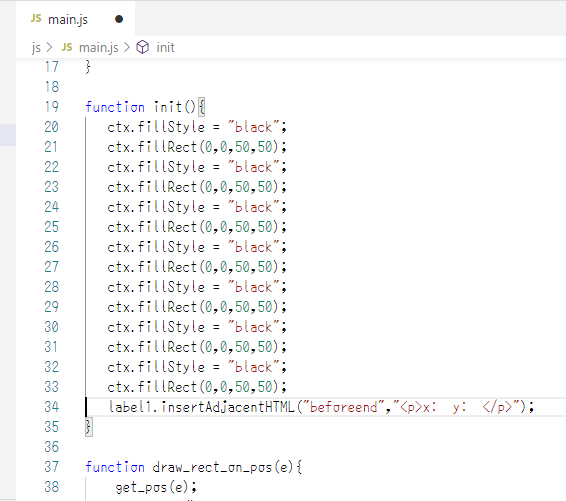
Command +cを1回、Command＋ｖを1回（これは捨てる分です）の後にCommand＋ｖを回数分行います

今回はコピーを1回、捨てるペーストを1回とペーストを8回行います。

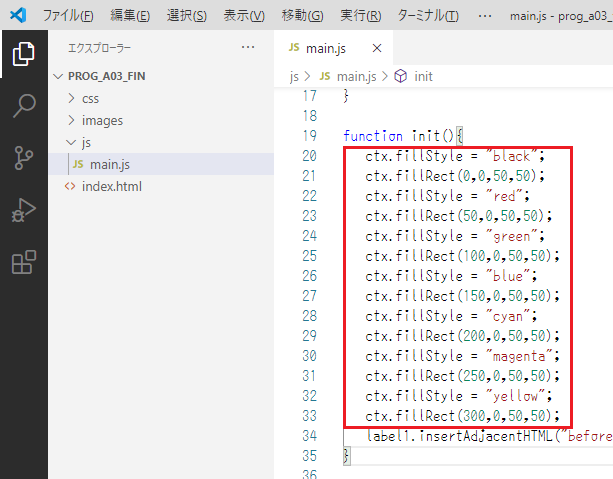
実際にはCtrlキーやCommandキーは先に押しっぱなしにしておいて

ｃ、ｖ、　ｖ、ｖ、ｖ、ｖ、ｖ、ｖ、ｖ、ｖ

とキーを押した後、最後に押しっぱなしにしていたCtrlキーやCommandキーを放します。



さらに色や数値を変更します。



マウスでエディットするだけでなくキーボードの４つの矢印キー（↑↓→←）も使うと効率よく打ち込めます。



パレットの機能はありませんが、黄色で四角が描けるようになりました。

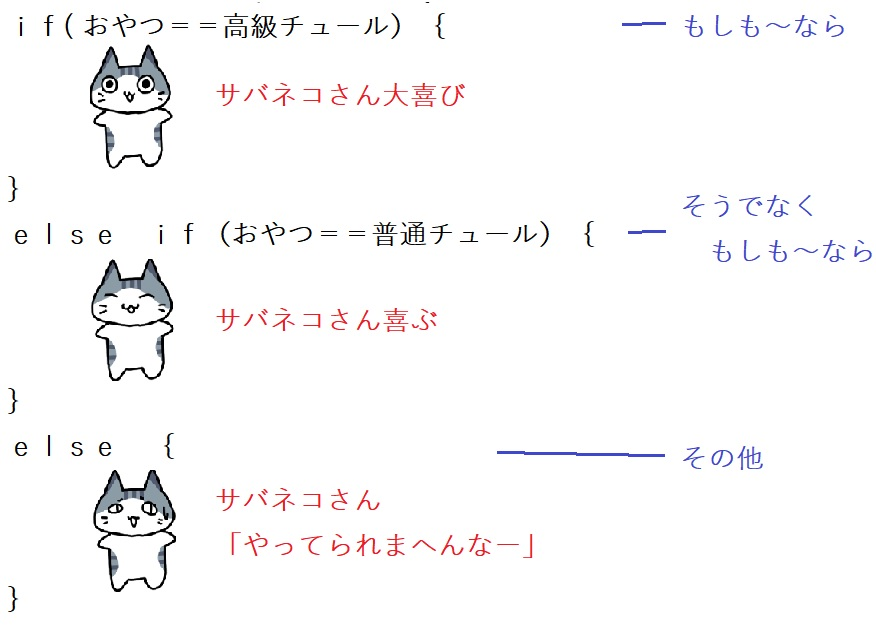
**【音声ファイル７】**

**８．main.jsファイルのプログラムの改良と実行２**

**色を変更できるようにします。**

ｉｆ文を使います

プログラムが条件によって違うように動くのはｉｆ文があるからです。



プログラムを変更しましょう

2重でｉｆ文を使っています。

48行：　ｍｙが５０より小さいか？　小さい場合だけ49行～66行の命令を行います

49行：　ｍｘを５０で割ったときの整数部分を求めｃｎに入れています

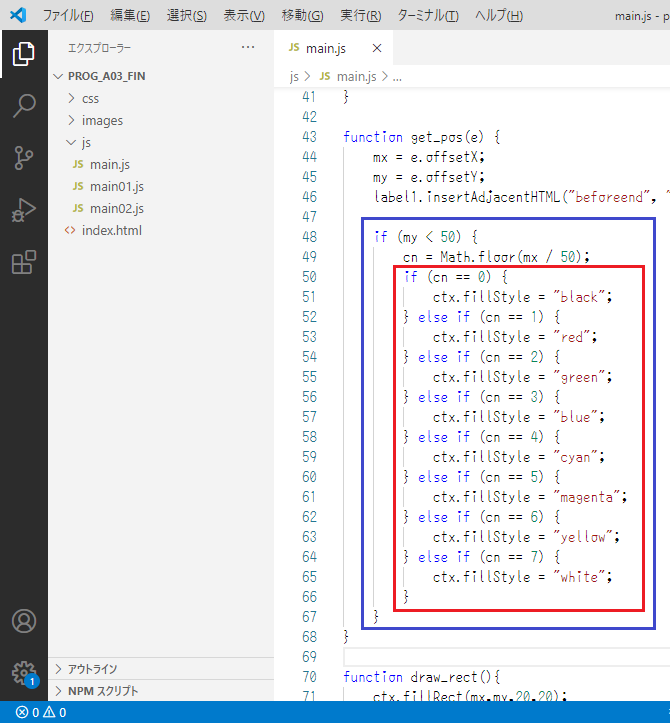
例　ｍｘ＝１４０のとき

mx/５０は2.８となる

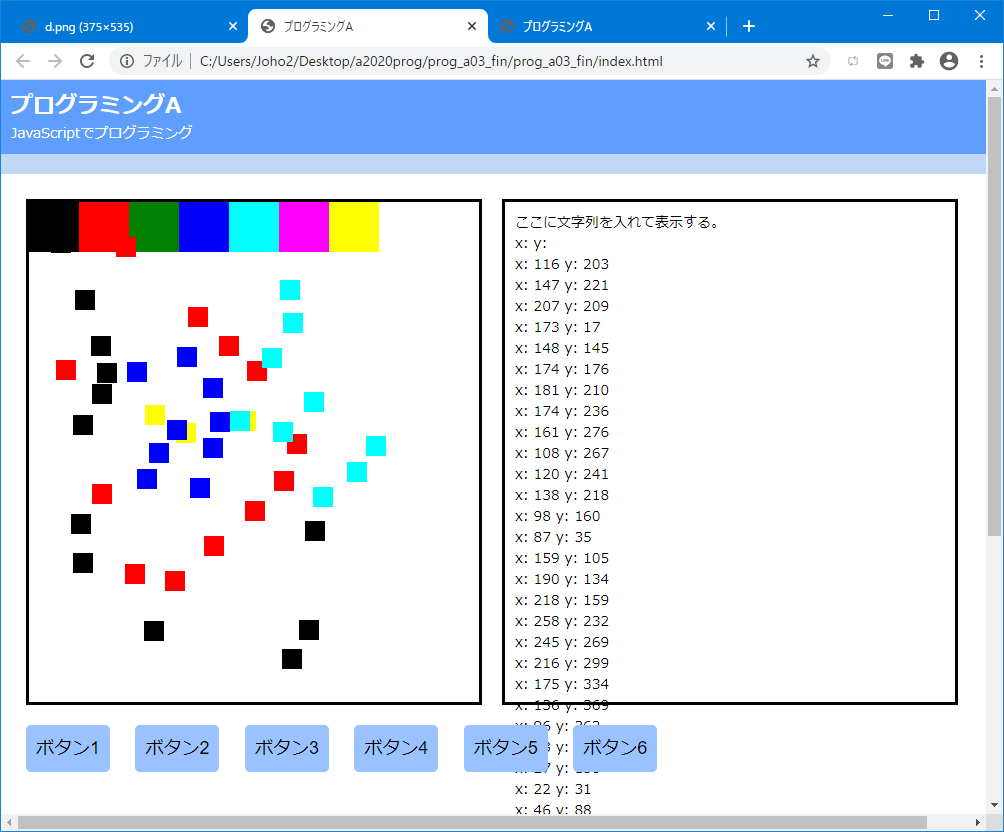
Math.floorで小数部分を切り落とし２になる

ｃｎ＝２となる

50行～66行：　ｃｎが０，１，２、・・・、７の場合に分けて、塗りつぶす色をセットしている。



青色の四角の部分を打ち込み、実行しましょう。



**【音声ファイル８】**

**９．main.jsファイルのプログラムの改良と実行３　ドット絵描き**

**四角の位置を切りの良いところにする。量子化、クオンタイズ**

前節でMath.floorを使いました。

**cn = Math.floor(mx / 50);**

例　ｍｘ＝１４０のとき mx/５０は2.８となる

Math.floorで小数部分を切り落とし２になる　ｃｎ＝２となる

では　１４０のとき１００を、８８のとき５０を、２０２のとき２００を得るには

出てきた答えに５０をかけてやればよいわけです

ただ、描画用の四角は２０×２０の大きさなので、５０ではなく２０としました

ｍｘとｍｙは再利用しています

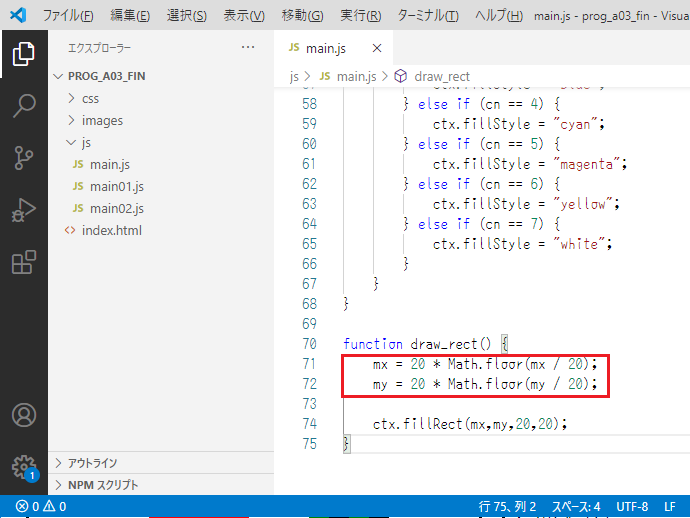
**mx = 20 \* Math.floor(mx / 20);**

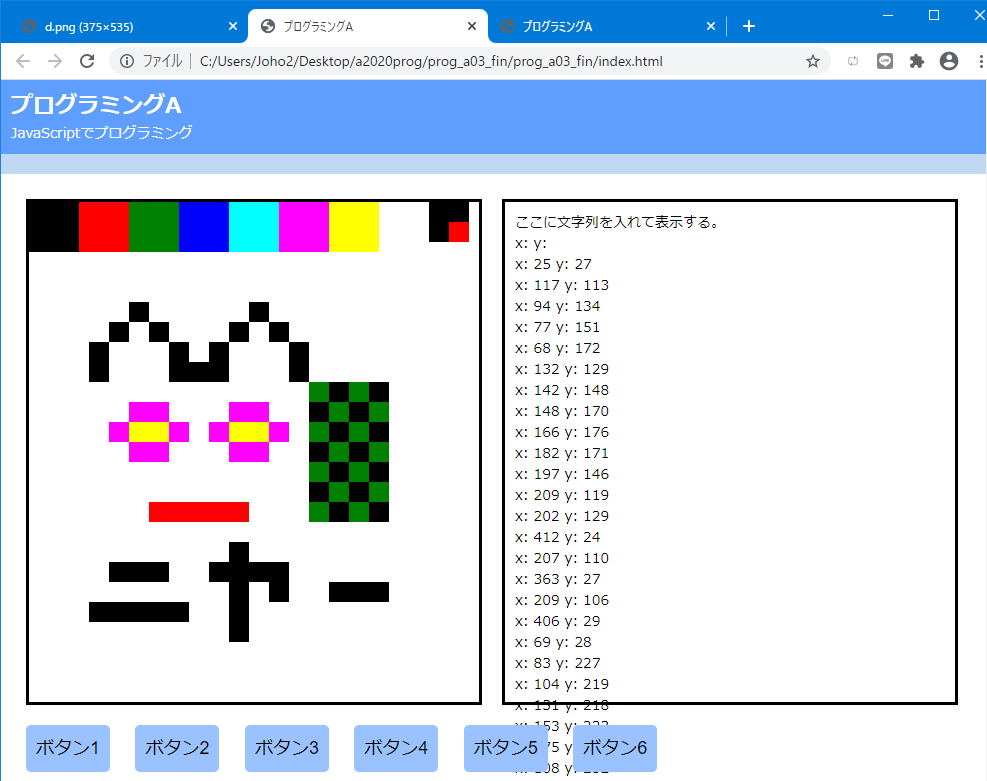
**my = 20 \* Math.floor(my / 20);**

これで、１１→０、２２→２０、３２→２０、３９→２０、４０→４０、１３２→１２０

のように、タイルの壁のようなぐあいのドット絵が描けるようになります。

赤枠の中の2行を追加しましょう





**【音声ファイル９】**

**１０．課題**

「９．main.jsファイルのプログラムの改良と実行３」までを終えて、できたプログラムを実行し、

マウスで絵を描いて提出してください。

感想・コメントを３行以上付けてください

問題を1つ出します

**７．main.jsファイルのプログラムの改良と実行１の結果、黄色い四角で描けるようになりますが**

**なぜ黄色になったのか、プログラム中の１行とその行数、理由を簡潔に答えてください。**

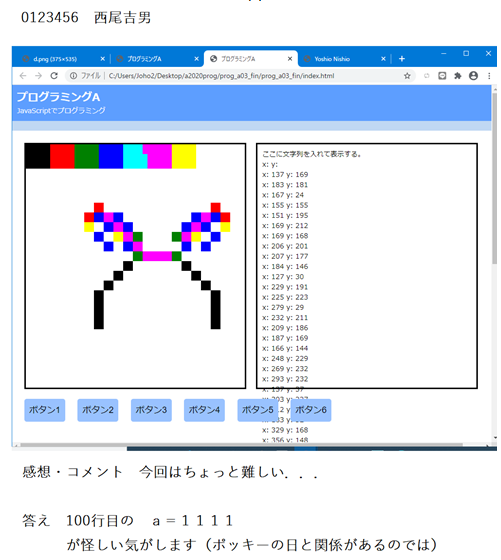
**答えれる人だけでいいので答えてください。追加点を差し上げます。**

この課題レポートは、１週間後の金曜日の２３時５５分です。

学情システムの方で受け付けています。

提出例

**芸工情報処理基礎　2024年5月17日**

****

どうしてもうまく動かないという人にはヒントがあります。

let button1 = document.getElementById("html\_button1");

let button2 = document.getElementById("html\_button2");

let button3 = document.getElementById("html\_button3");

let button4 = document.getElementById("html\_button4");

let box = document.getElementById("html\_box");

let label1 = document.getElementById("html\_label1");

let canvas1 =document.getElementById("html\_canvas");

let ctx = canvas1.getContext("2d");

let mx = 0;

let my = 0;

let palette\_colors = ["black", "red", "green", "blue", "cyan", "magenta", "yellow"]

main();

function main(){

window.onload = init;

canvas1.onmousedown =draw\_rect\_on\_pos;

}

function init(){

for (let i=0; i< palette\_colors.length; i++){

ctx.fillStyle = palette\_colors[i];

ctx.fillRect(0+50\*i, 0, 50, 50);

}

label1.insertAdjacentHTML("beforeend","<p>x: y: </p>");

}

function draw\_rect\_on\_pos(e){

get\_pos(e);

}

function get\_pos(e){

mx = e.offsetX;

my = e.offsetY;

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

if (my > 50){

draw\_rect();

}else{

if (mx >= 0 & mx < 50){

ctx.fillStyle = palette\_colors[0];

}else if(mx >=50 && mx < 100){

ctx.fillStyle = palette\_colors[1];

}else if(mx >= 100 && mx < 150){

ctx.fillStyle = palette\_colors[2];

}else if(mx >= 150 && mx < 200){

ctx.fillStyle = palette\_colors[3];

}else if(mx >= 200 && mx < 250){

ctx.fillStyle = palette\_colors[4];

}else if(mx >= 250 && mx < 300){

ctx.fillStyle = palette\_colors[5];

}else if(mx >= 300 && mx < 350){

ctx.fillStyle = palette\_colors[6];

}else{

ctx.fillStyle = "white";

}

label1.insertAdjacentHTML("beforeend", "色を変更した。ヒント版です");

}

}

function draw\_rect(){

ctx.fillRect(mx,my,20,20);

}