

この時間のねらい:

1. 変数の宣言方法を学ぼう。(今回は論理・色型について)
2. 論理演算と色を図形に適用する方法を学ぼう。

## 論理演算とは

メモ欄

そもそも、演算というのは( )を行うことです。

今回の論理演算とは、( )だと思ってください。

例えば、Aボタンを押せば、キャラクターがジャンプするとすれば、

右のときコントローラーのAボタンは押されていますか？

→( )

このように、はい or いいえ で答えることが出来るような演算を

( )と呼びます。



## 論理型変数の宣言方法・利用

`boolean ronri = false; //又はtrue`

右上のように宣言する。

格納する値はtrueまたはfalseのみしかないが、意味を考えながら名前や値を入れないと、バグの原因になりやすい。

なるべく利用したくない人が多いと思いますが、ゲームクリアの条件や、ブロック崩しでのブロックを消す条件に利用する必要があるため、逃れることは無理でしょう。諦めな。

## 特別な論理型変数

## ◆マウス

- `mouseX` ... ( )
- `mouseY` ... ( )
- `pmouseX` ... ( )
- `pmouseY` ... ( )
- `mousePressed` ... ( )
- `mouseButton` ... ( )

## ◆キーボード

- `key` ... ( )
- `keyCode` ... ( )
- `keyPressed` ... ( )

以上のものは決められている変数です。使用できるように覚えちゃいましょう。

## 画面上での色

パソコンのモニターには右の小さく黒い四角が無数にあります。

これを( )と呼びます。

この画素には特徴があり、場所によって色が光るようになっています。

左から( )、( )、( )に光ります。

それぞれの色の度合いによって、その部分の色を決めていく方法で

色付けを行います。これを( )と呼びます。

また、パソコンの電源がオフのとき、それぞれの値は( )です。→( )

それぞれの値の最大値は( )です。3色を足すと明るさが増すので、( )とも呼ぶ。



## 紙面上での色

続いてプリント等の色についてです。

小学校の頃に図工などで絵を描いたとき色を塗るときには

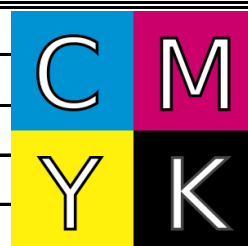
( )、( )、( )の三色を使いましたよね？

コピー機も同じような色です。

色は( )、( )、( )、( )

が使用されます。これを( )と呼びます。

ちなみに、色を混ぜれば混ぜるほど( )に近づくので、明るさが減る( )とも呼ぶ。



Processing上で利用するカラーモデルは RGB になります。少なくともそれは覚えましょう。

色を変更する関数を見ておきましょう。

① `fill( 赤, 緑, 青 ); ... ( )`

② `fill( 赤, 緑, 青, 透明度 ); ... ( )`

③ `background( 赤, 緑, 青 ); ... ( )`

④ `stroke( 赤, 緑, 青 ); ... ( )`

これらを利用すると、色を変更しカラフルな見た目を実現できます。

最後に色の変更方法、変数宣言方法を紹介します。

右のように混合する色の度合いを調整することで、たくさんの

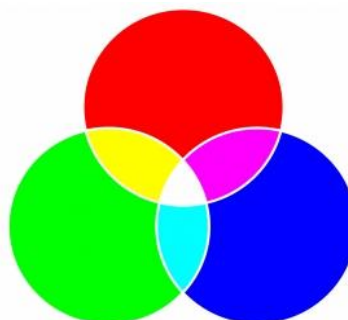
色を作ることができます。

たとえば、赤 + 緑 = ( )のような感覚です。

それぞれの色をどれくらい混ぜるかで、色の違いが出ます。

困った人は( )を検索し、利用しましょう。

変数宣言方法は右下のようにして下さい。



```
color iromei = color( 赤, 緑, 青 );
```