

この時間のねらい: ① 変数を使ってみよう(カラーコード編)

② 変数を使ってみよう(真偽の判定編)

① カラーコードを管理する変数

カラーコードの変数宣言は1つだけです。※前々回のプリントにもあります。

color 型	カラーコードを格納するための変数を宣言
---------	---------------------

・使い方

以前は使い方を一切話さず終了しました。そのため、試してみたところ実行できない状態になってしまった人がほとんどかと思います。以下のように使うと良いでしょう。

`color red = color(255, 0, 0);` // color 型の変数 red を宣言し、red に `color(255, 0, 0)` を代入または、

`color red;` // color 型の変数 red を宣言

`red = color(255, 0, 0);` // red に `color(255, 0, 0)` を代入

今回の代入方法は特殊ですね。(私は最近まで忘れてました。)

color 宣言を行った後に color 関数を使用しています。これはパソコンがこの数字の並びが何なのかを把握するためです。

※RGB ではなく、#〇〇〇を利用する場合、color 関数は使わない

また、今回の演習を行うために window 内に文字を書くために関連する命令もまとめておきましょう。

<code>text("言葉"or 数値, 左上 x 座標, 左上 y 座標)</code>	ウィンドウ上に言葉や数値を表示する
<code>textAlign(alignX, alignY);</code>	文字の表示位置を指定する。 alignX : LEFT, CENTER, RIGHT alignY : TOP, BOTTOM, UNDER
<code>textSize(数値);</code>	文字の大きさを設定する

以上の関数を駆使して、右のような表示に出来るだけ似せてみてください。

※英語以外を表示するためには PFont 宣言をする必要があります。使いたい場合はリファレンスにて、使い方を調べてください。



① 真偽判定のための変数

真偽を判定するための変数宣言は 1 つだけです。※前々回のプリントにもあります。

boolean 型	真偽判定を格納するための変数を宣言
-----------	-------------------

・そもそも真偽判定とは

簡単に言えば、**どのような状態か**をパソコンが把握するために必要なものです。

例えば、ブロック崩しにてブロックと玉が衝突したかどうかや、RPG にて体力があるかないか(生きているか死んでいるか)などの状態が保存されるようなものです。

また、ほかの変数宣言では入る値がいろいろとありましたが、この変数では 2 つに決まっています。それは true(真), false(偽)です。基本的にあとから行う**条件文**のところで学ぶ if という関数と組み合わせて使用します。

使い方は以下の通りです。

```
boolean block_break = false; //boolean 型変数の block_break を宣言し false を代入
```

もしくは

```
boolean block_break; //boolean 型変数の block_break を宣言
```

```
block_break = false; //block_break に false を代入
```

実際にこれを使うとどうなるかを見てみましょう。以下のようにプログラムを書いてみましょう。

```
block_break_boolean ▼
1 boolean block_break = true;
2
3 if(block_break == false){
4   rect(10, 20, 75, 50);
5 }else if(block_break == true){
6
7 }
```

実際に入力して実行すると四角形が表示されるはずですが。

また、block_break に代入する値を false に変更するとどのように変わのでしょうか。

答えは「**四角形の表示が消える**」です。

変数の中でも使い方が理解しづらい宣言の一つなので積極的に使用して、使い方の研究をしてみましょう。