

この時間のねらい：① マウスの操作により、実行画面での操作をできるようになろう。

前回は条件文の使い方、どのようなときに使うべきかを学びました。(少なくともプリントのほうに記述してあるはずです。)今回は、条件文の使用方法を広げるため、特別な変数として利用されているものやマウスの情報を引数にできる変数を効果的に活用できるように勉強していきます。

特別な変数のことを_____と呼びます。その名の通り_____内に勝手に代入される変数であるので、以前学んだ変数とは違い、宣言する必要はありません。(というよりしたくてもできません。)

以下によく使われる変数をまとめておきます。

mouseX	_____の位置の x 座標を格納
mouseY	_____の位置の y 座標を格納
pmouseX	1 フレーム前の _____の位置の x 座標を格納
pmouseY	1 フレーム前の _____の位置の y 座標を格納
width	ウィンドウの_____の大きさを格納
height	ウィンドウの_____の大きさを格納
mousePressed	_____が押されているかどうか
mouseButton	押されている_____はどれか

上から6つのものに関しては、実際に使用して、確認しておきましょう。

下2つに関しては少々特殊な使用方法となっていますので、どのように使うのかを深く説明していきます。

```

1 if(mousePressed){
2   if(mouseButton == LEFT){
3     /* 処理内容1 */
4   }else if(mouseButton == CENTER){
5     /* 処理内容2 */
6   }else if(mouseButton == RIGHT){
7     /* 処理内容3 */
8   }
9 }
```

左のように記述することでマウスとの反応を実現できます。

やっていることを細かく見ていくと一番最初に mousePressed の判定があり、その上でどのボタンが押されているかを判定して、ボタンによって処理を変更しています。

ここで演習問題を行うために基本的な関数を学びましょう。

void setup()	1度のみ実行する
void draw()	何度も実行する

基本的にこの2つは、開発する際に利用すべきものですので、しっかりと理解しておきましょう。

基本的に setup()の中に入れるべきものは_____や、ウィンドウを開くための_____など、一度実行してしまえば、その後もずっと(勝手に)実行してくれるものを入れるべきです。

それらを draw()内に記述してしまうとパソコンへの負荷が多くなるため、重くなってしまいます。どちらに入れるべきかわからない場合はすぐに聞くと沼にはまらないと思われます。(下手にデスマーチに入るとなおさら変になってしまうこともあります。)

以上の内容を踏まえて、以下の演習問題に取り組んでみましょう。

問題 1

マウスカーソルに沿って移動する円を作成しなさい。

また、背景を更新し、円の移動した軌跡などが表示されないように改良しなさい。

問題 2

1で作成した円の色をマウスのボタンをクリックすることで以下の条件のように変更させなさい。

条件：左クリックで赤色、ホイールをクリックで青色、右クリックで黄色

クリック無しで白色

問題 3

2まで作成したプログラムに四角形を追加して、その図形の色を円とは別の色に変更できるようにプログラムを追加しなさい。

問題 4

3までに条件文が2つほど出てきています。どちらかに処理を追加して、図形を移動できるように改良しなさい。