

プログラミング演習I (第12回) 課題

● 基本① スケッチ名: **coincount**

- 10円玉、50円玉、100円玉、500円玉の枚数を与えると、その合計金額を返す以下の関数を作ってください。

```
int coincount( int n_coin10, int n_coin50, int n_coin100, int n_coin500 ) {  
    return 合計金額;  
}
```

- 適当な硬貨枚数を設定し、上記の関数を使って合計金額を計算し、枚数と合計金額を画面に表示してください。グローバル変数は使わないこと!

10yen x 4
50yen x 3
100yen x 2
500yen x 1

890yen

```
textSize(30);  
text( "10yen x " + 10円の枚数, 30, 50 );  
text( "50yen x " + 50円の枚数, 30, 90 );  
text( "100yen x " + 100円の枚数, 30, 130 );  
text( "500yen x " + 500円の枚数, 30, 170 );
```

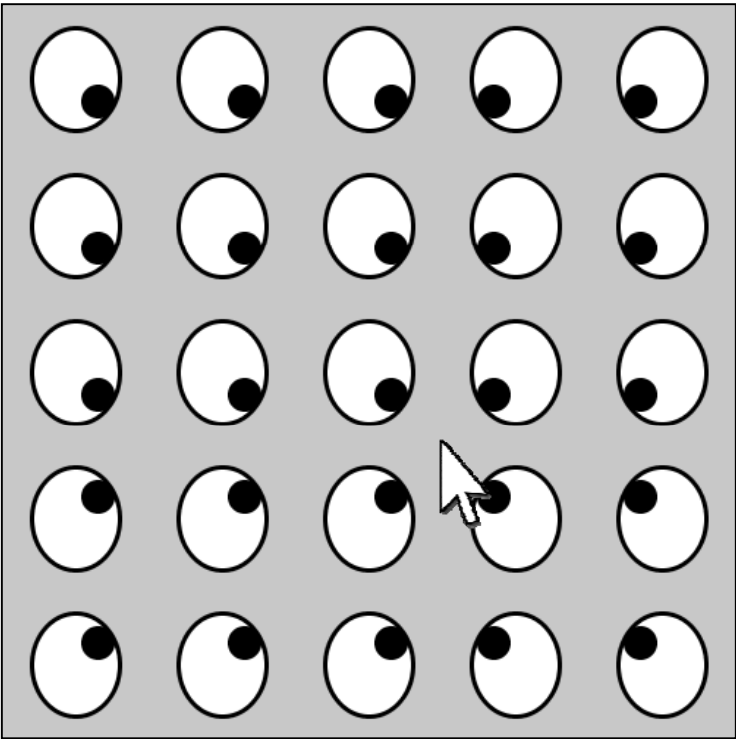
```
textSize(80);  
text( 計算結果 + "yen", 30, 260 );
```

文字表示に関する処理の抜粋

プログラミング演習I (第12回) 課題

• 基本② スケッチ名: eyes55

- カーソルのある方向に向く目玉を5x5個並べるプログラムを作ってください。
- 目玉が向く方向は 8方向(+中央) とします。



[step1]

まずはカーソルのある方向に黒目が向く目玉を1つ描く関数を作ろう。引数によって位置を指定できるようにしよう。

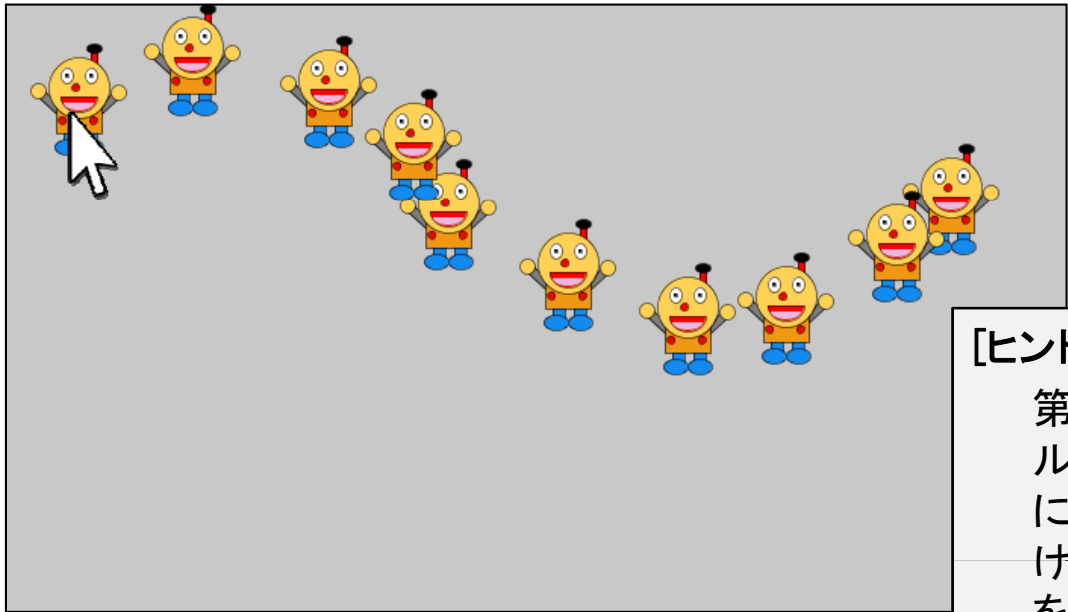
[step2]

2重の繰り返し文と目玉を描く関数を組み合わせて、5x5個の目玉を描こう。

プログラミング演習I（第12回）課題

• 基本③ スケッチ名 : mousetracking

- マウスカーソルの後を、自作のキャラクタが列をなして追尾するプログラムを作ってください。
- キャラクタの数は10体とします。また、余裕のある人はキャラクタ同士の間隔がある程度あくように工夫してください。



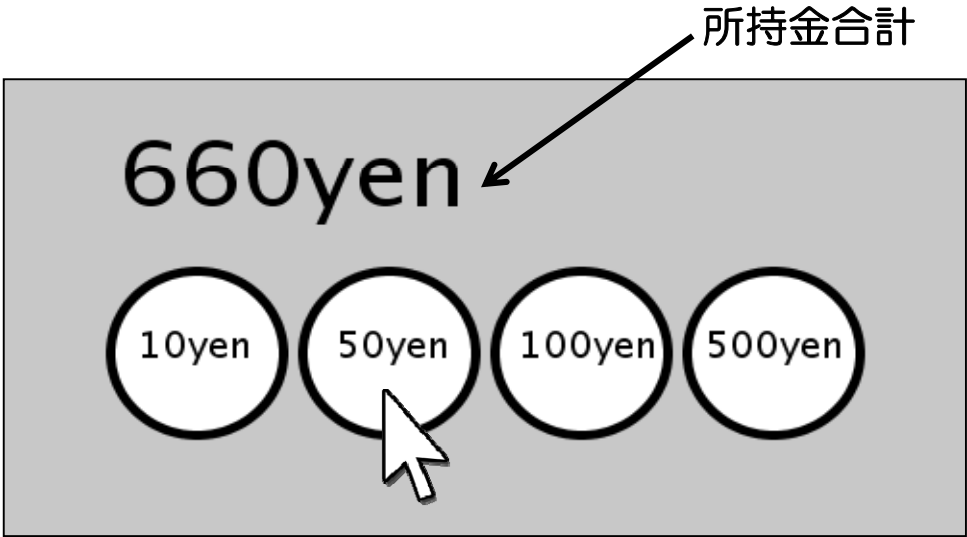
[ヒント]

第8回のテキストにある「円がカーソルを追尾するプログラム」をベースに考えよう。キャラ同士の間隔をあげるには、例えば、カーソルの位置を100個保存して、その中から10個飛ばしで表示させればよい。

プログラミング演習I (第12回) 課題

• 発展① スケッチ名: coincount2

- 画面に硬貨(10円、50円、100円、500円)を模した円が表示され、それらをクリックするごとに合計金額が増えていくプログラムを作成してください。
- 合計金額の計算には基本①で自作した関数を使うこと。



[ヒント]

硬貨のx座標、y座標、枚数をそれぞれ保存する配列を作ろう。繰り返し文と配列の組み合わせによって、クリックの当り判定とカウントアップの処理をスマートに書けるはずだ。

円をクリックすると所持金に足される

自分のキャラを表示する関数を作ろう(再掲)

```
void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
```

```
int offset_x = 190; ← offset_x と offset_y はキャラの中心座標。  
int offset_y = 200; ← 自分のキャラクタに合わせて値を変えること。
```

```
pushMatrix();  
translate( x, y );  
scale( scale_factor );  
rotate( radians( angle_deg ) );  
translate( -offset_x, -offset_y );
```

ここにchara1で作った、キャラクタを描画する処理をコピーしてください。
ただし、size()とbackground()は入れないこと。

```
popMatrix();  
}
```

drawCharacter() の使い方 (再掲)

drawCharacter(X座標, Y座標, 回転角度, 拡大率);

回転角度の単位は【度】、拡大率は1.0で等倍。

```
void setup() {
  size( 600, 400 );
}

void draw() {
  background(255);

  // 例 : マウスカーソルがある位置に45度傾けて0.5倍でキャラクタを描画
  drawCharacter( mouseX, mouseY, 45.0, 0.5 );
}

void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
  .....
}
```

プログラミング演習I (第12回) 課題

● 基本① スケッチ名: `timeconvert`

- 時刻(時・分・秒)を与えると、秒に換算して返す関数を作ってください。
- 0時0分0秒からの経過秒数と考える。例えば、1時00分12秒は3612秒。

```
int convTimeToSecond( int hour, int minute, int second ) {  
    return 秒換算した値;  
}
```

- 現在時刻を上記の関数を使って秒に換算し、時刻と秒換算した値の両方を表示してください。グローバル変数は使わないこと!

```
textSize(30);  
text( 時 + ":" + 分 + ":" + 秒 + " is", 30, 50 );  
  
textSize(50);  
text( 計算結果 + "sec", 30, 120 );
```

文字表示に関する処理の抜粋

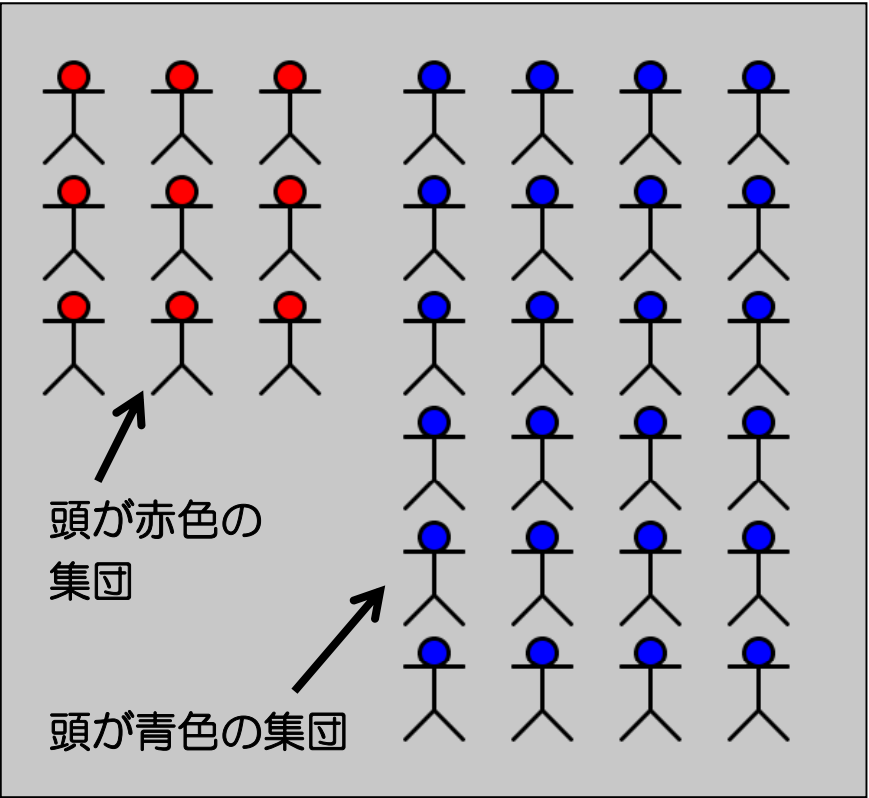
12:30:45 is

45045sec

プログラミング演習I (第12回) 課題

• 基本② スケッチ名: `people`

- 3x3に並べられた【赤色】の棒人間の集団と、4x6に並べられた【青色】の棒人間の集団の両方を描くプログラムを作ってください。
- なお、2つの集団が重ならないようにすること。



[ヒント]

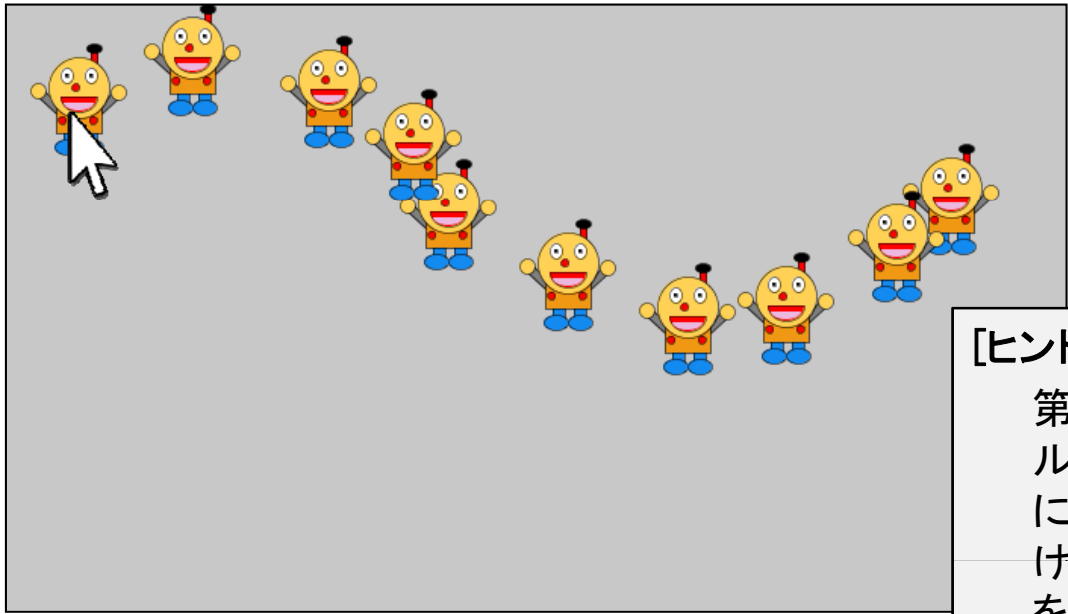
第10回基本①(`human`)を思い出そう。
棒人間を描く `drawHuman()` を上手に使う。

集団の左上座標と、縦横の列数を指定すると棒人間集団を描いてくれる関数を作るとスマートなプログラムになる。

プログラミング演習I（第12回）課題

• 基本③ スケッチ名 : mousetracking

- マウスカーソルの後を、自作のキャラクタが列をなして追尾するプログラムを作ってください。
- キャラクタの数は10体とします。また、余裕のある人はキャラクタ同士の間隔がある程度あくように工夫してください。



[ヒント]

第8回のテキストにある「円がカーソルを追尾するプログラム」をベースに考えよう。キャラ同士の間隔をあげるには、例えば、カーソルの位置を100個保存して、その中から10個飛ばしで表示させればよい。

プログラミング演習I (第12回) 課題

• 発展① スケッチ名: countdown

- 授業終了(19:30)までの残り時間をカウントダウン表示するプログラムを作ってください。グローバル変数は使わないこと。
- 残り時間を【秒換算した値】と【分換算した値】の2つを表示してください。

3658sec ← 授業終了までの残り秒数
61min ← それを分換算したもの

- 基本①で作った自作関数 `convTimeToSecond()` を使うこと。また、秒を分に換算する関数 `convSecondToMinute()` も新たに作ること。計算時には、`round()` を使って四捨五入すること。

```
int convSecondToMinute( int second ) {  
  
    return 分換算した値;  
}
```

自分のキャラを表示する関数を作ろう(再掲)

```
void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
```

```
int offset_x = 190; ← offset_x と offset_y はキャラの中心座標。  
int offset_y = 200; ← 自分のキャラクタに合わせて値を変えること。
```

```
pushMatrix();  
translate( x, y );  
scale( scale_factor );  
rotate( radians( angle_deg ) );  
translate( -offset_x, -offset_y );
```

ここにchara1で作った、キャラクタを描画する処理をコピーしてください。
ただし、size()とbackground()は入れないこと。

```
popMatrix();  
}
```

drawCharacter() の使い方 (再掲)

drawCharacter(X座標, Y座標, 回転角度, 拡大率);

回転角度の単位は【度】、拡大率は1.0で等倍。

```
void setup() {
  size( 600, 400 );
}

void draw() {
  background(255);

  // 例 : マウスカーソルがある位置に45度傾けて0.5倍でキャラクタを描画
  drawCharacter( mouseX, mouseY, 45.0, 0.5 );
}

void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
  .....
}
```