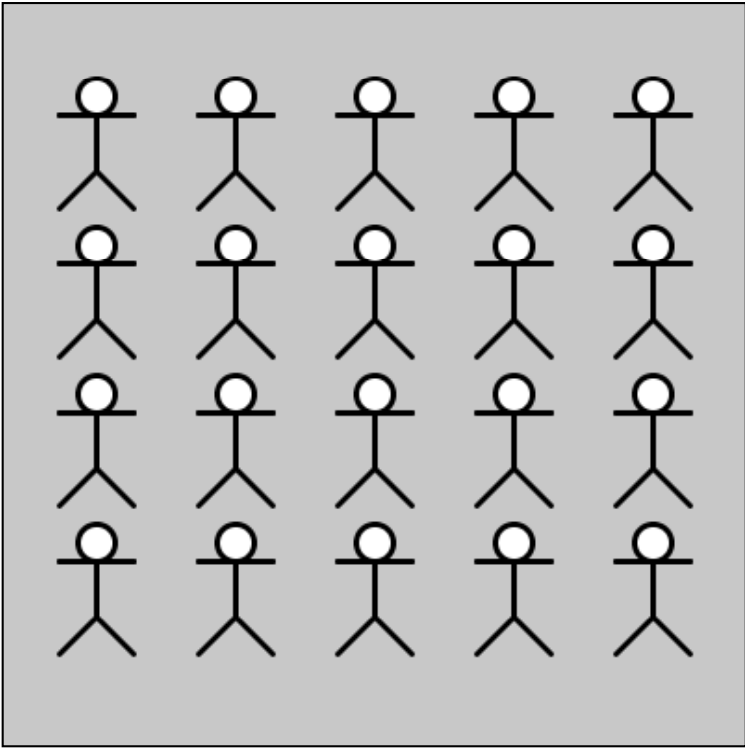


# プログラミング演習I (第10回) 課題

## • 基本① スケッチ名: human

- 棒人間を画面内に整列させるプログラムを作ってください。
- 棒人間1体を描く関数を使って、横に5例、縦に4列並べてください。



### [step1]

まずは予習テキストにある棒人間を描くプログラムを動かしてみよう。

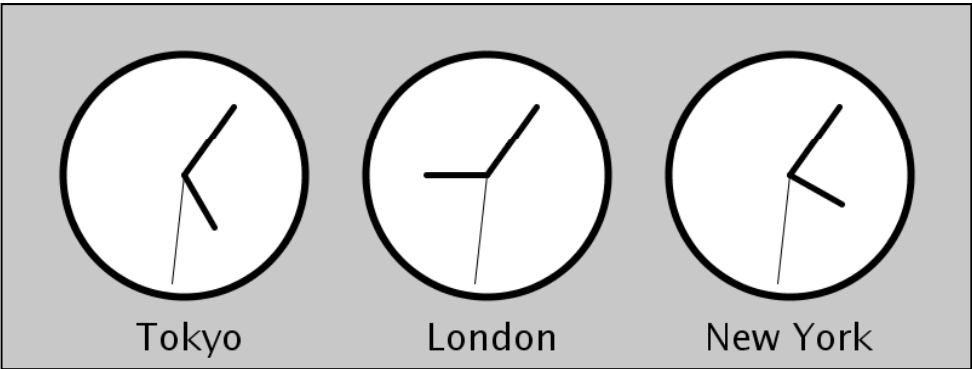
### [step2]

棒人間を描く関数を使って縦横に並べるには、2重の繰り返し文(for文またはwhile文)を使えばよい。座標の指定の仕方を工夫してみよう。

# プログラミング演習I (第10回) 課題

## • 基本② スケッチ名: `clocks`

- 東京、ロンドン、ニューヨークの時計を並べるプログラムを作成してください。ロンドンは-8時間、ニューヨークは-13時間の時差があります。
- 時計の下に地名を表示してください。



### [step1]

まずは予習テキストにある時計を描くプログラムを動かしてみよう。

### [step2]

時計を描く関数を3つ使い、それぞれ異なる位置に、異なる時刻で表示されるようにしよう。

# プログラミング演習I (第10回) 課題

---

## • 基本③ スケッチ名: money

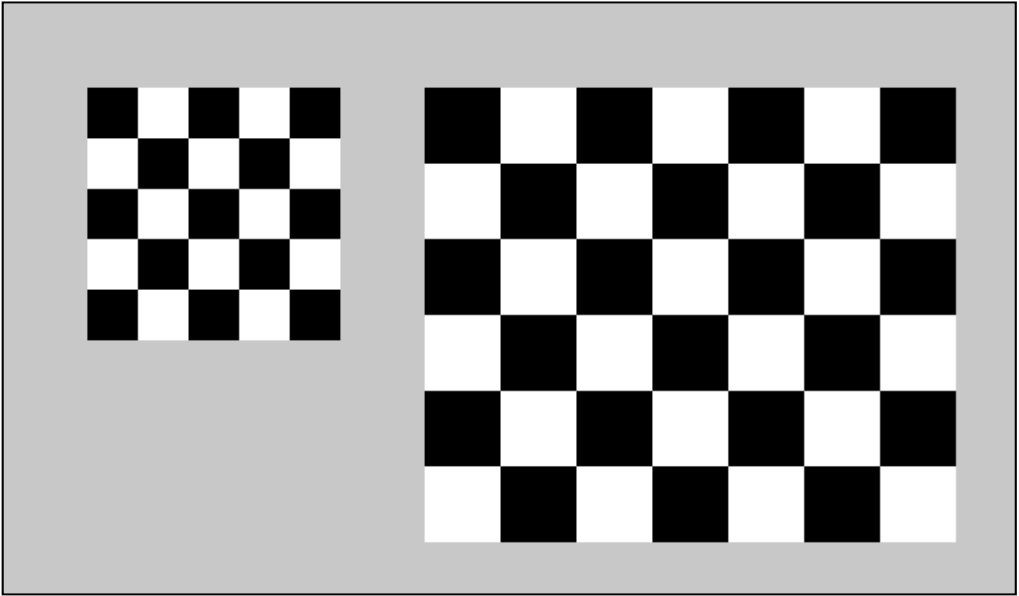
- 今日の為替レートでUSDから円へ変換する関数を作ってください。関数名は `convToJPY()` としてください。
- 画面に適当なUSDの金額と、それを日本円換算したものを表示してください。その際、`convToJPY`を使うこと。
- 変換時に生じた端数(1円未満)は切り捨てにすること。

**USD: 12**  
**JPY : 1214**

# プログラミング演習I (第10回) 課題

## • 発展① スケッチ名: chess2

- 位置、横・縦のマス数、マスの大きさを指定してチェス盤を描く関数を作ってください。また、その関数を使ってサイズ情報の異なる2種類のチェス盤を描いてください。



### [ヒント]

以前作ったチェス盤を描くプログラムをベースに、

X座標 `pos_x`

Y座標 `pos_y`

横のマス数 `size_x`

縦のマス数 `size_y`

マスの大きさ `w`

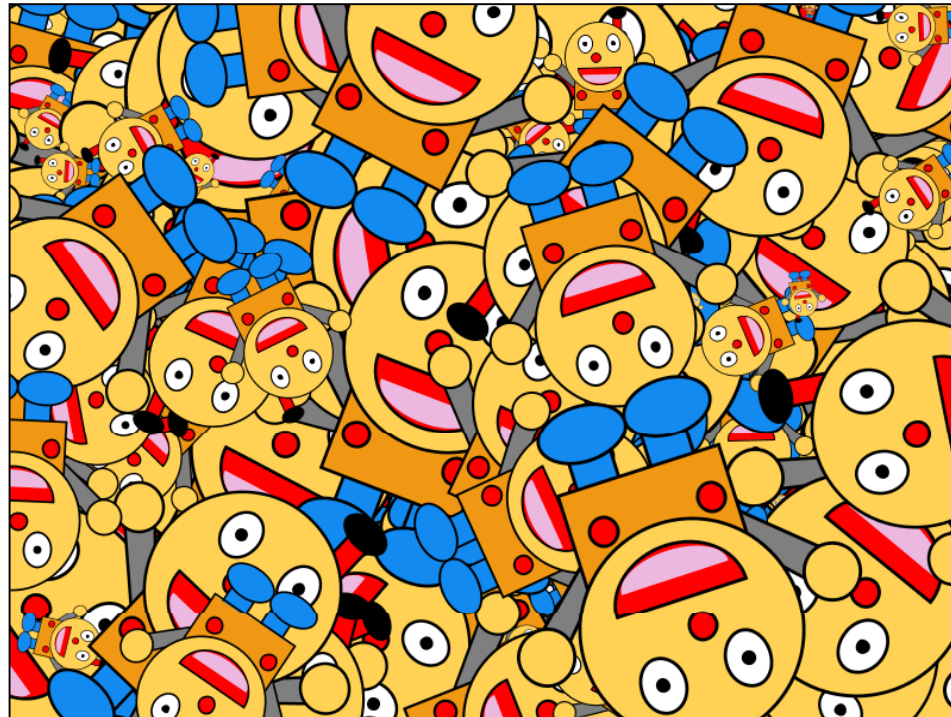
という5つの引数を持つ関数を作ろう。

# プログラミング演習I（第10回）課題

---

## • 発展② スケッチ名: chara4

- chara1で作成したキャラクターを画面内にたくさん描くプログラムを作成してください。
- 位置・大きさ・回転をランダムにしてください。



# 自分のキャラを表示する関数を作ろう

```
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

---

**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);

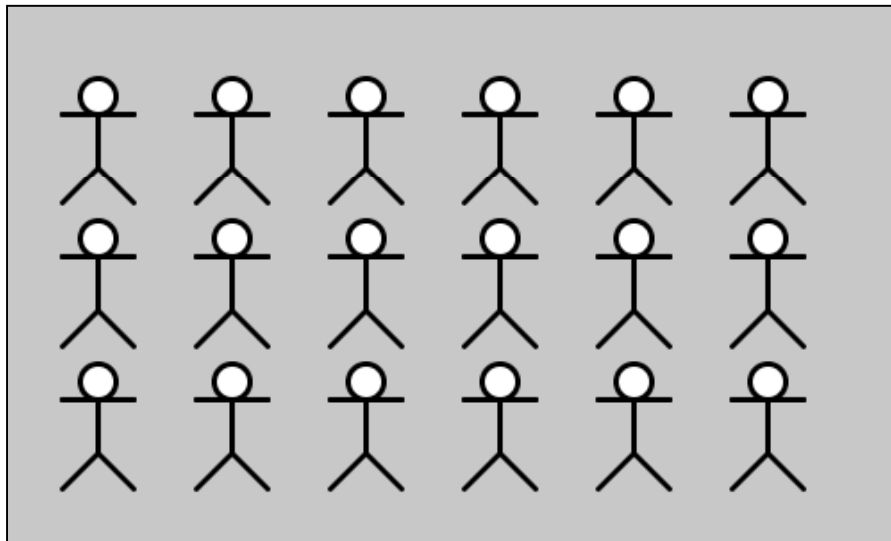
  // 例 : マウスカーソルがある位置に45度傾けて0.5倍でキャラクタを描画
  drawCharacter( mouseX, mouseY, 45.0, 0.5 );
}

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

# プログラミング演習I（第10回）課題

## • 基本① スケッチ名: human

- 棒人間を画面内に整列させるプログラムを作ってください。
- 棒人間1体を描く関数を使って、横に6列、縦に3列並べてください。



### [step1]

まずは予習テキストにある棒人間を描くプログラムを動かしてみよう。

### [step2]

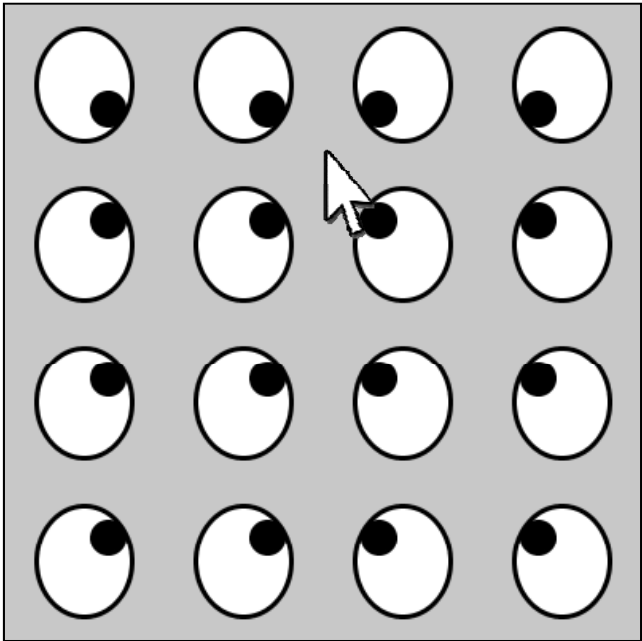
棒人間を描く関数を使って縦横に並べるには、2重の繰り返し文(for文またはwhile文)を使えばよい。座標の指定を工夫してみよう。



# プログラミング演習I (第10回) 課題

## • 基本② スケッチ名: eyes

- カーソルのある方向に向く目玉を4x4個並べるプログラムを作成してください。
- 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、4x4個の目玉を描こう。

# プログラミング演習I（第10回）課題

---

## • 基本③ スケッチ名: money

- 今日の為替レートで日本円からUSDドルへ変換する関数を作ってください。関数名は `convToUSD()` としてください。
- 画面に適当な日本円の金額と、それをUSDドル換算したものを表示してください。その際、`convToUSD`を使うこと。
- 変換時に生じた端数(1ドル未満)は四捨五入すること。

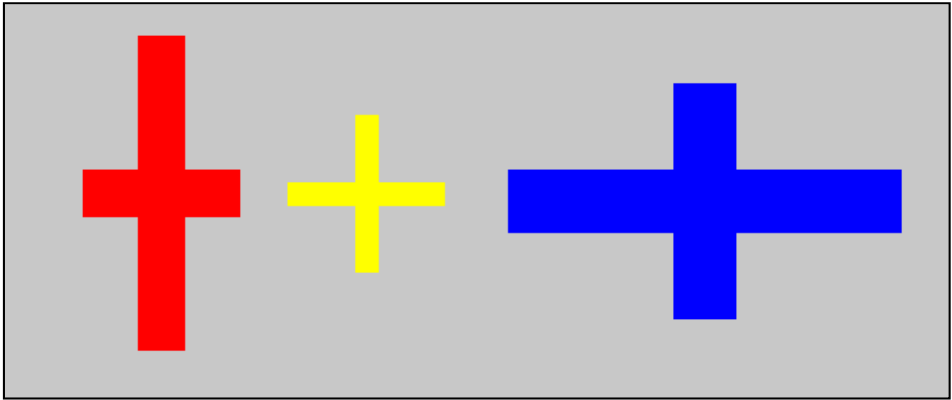
**JPY: 1980**

**USD: 20**

# プログラミング演習I (第10回) 課題

## • 発展① スケッチ名: `cross2`

- 十字型の図形を並べるプログラムを描いてください。
- 表示位置、全体の横幅・縦幅、太さを指定して十字図形を描画する関数を作ってください。それを使ってサイズの異なる3種類の十字形を描いてください。



### [ヒント]

十字形は`rect()`を2個組み合わせれば描ける。十字形を描く関数の引数は、例えば、

X座標 `pos_x`

Y座標 `pos_y`

全体の横幅 `size_x`

全体の縦幅 `size_y`

太さ `w`

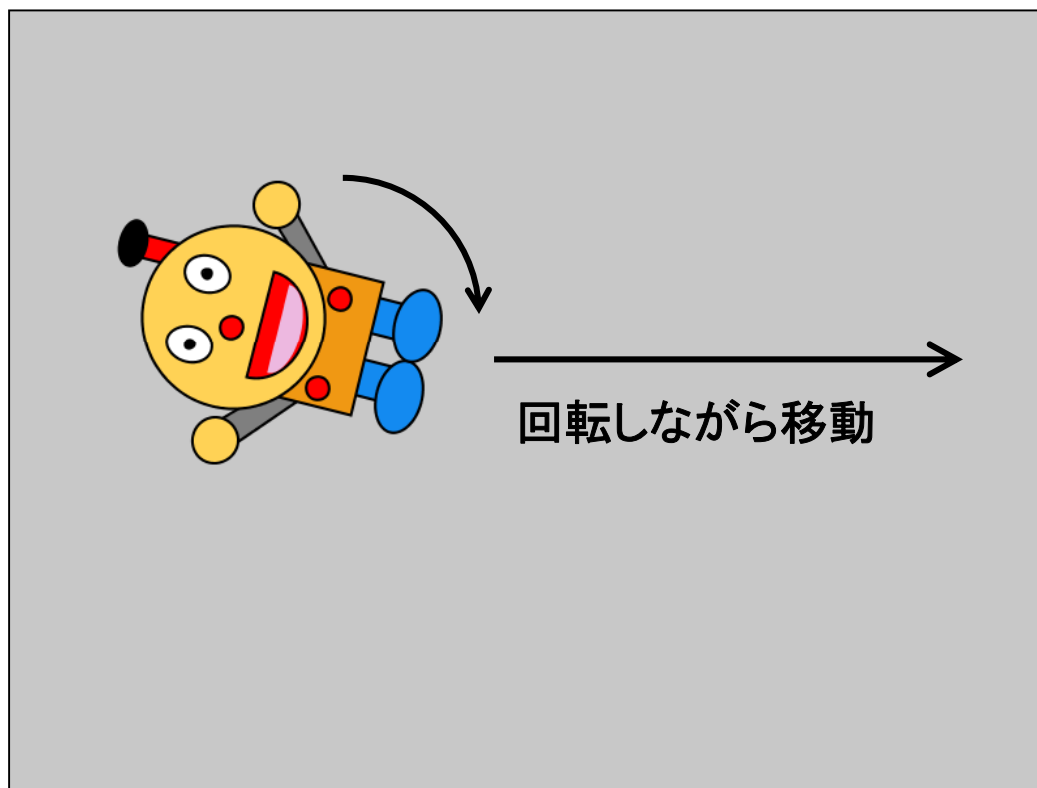
という風にしよう。

# プログラミング演習I（第10回）課題

---

- 発展② スケッチ名: chara4

- chara1で作成したキャラクターが左から右へ回転しながら移動するプログラムを作成してください。



# 自分のキャラを表示する関数を作ろう

---

```
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

---

**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 ) {
  .....
}
```