

# プログラミング演習I (第7回) 課題

## • 基本① スケッチ名: `table99`

- 九九の表を書くプログラムを作成してください。
- ただし【偶数】のみ赤文字にしてください。

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

### [step1]

まずは2重のwhile文(またはfor文)を使って、2つの変数の値を変化させて九九の表を作ってみよう。文字が思い通りの位置に表示されるようにするにはどうしたらいいだろうか？

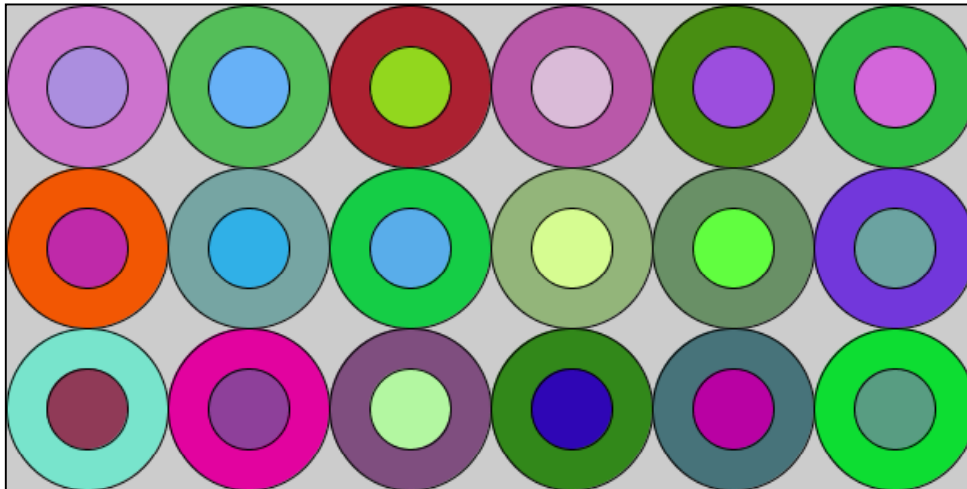
### [step2]

if文を使って各数値が偶数かどうかを調べ、条件分岐によって塗り色が変わるようにしてみよう。

# プログラミング演習I（第7回）課題

## • 基本② スケッチ名: circle2

- 画面内を2重の円で敷き詰めるプログラムを作成してください。
- 外円の直径を100、内円の直径を50とし、横に6個、縦に3個並べてください。ウィンドウのサイズは600x300とします。
- 円が画面内にぴったり収まるようにしましょう。また、塗り色は外円、内円ともにランダムにしてください。



### [step1]

まずは外円だけを指定された数だけ並べてみよう。円が欠けないようにピッタリ収まるためには、どの位置からスタートさせたらよいだろうか？

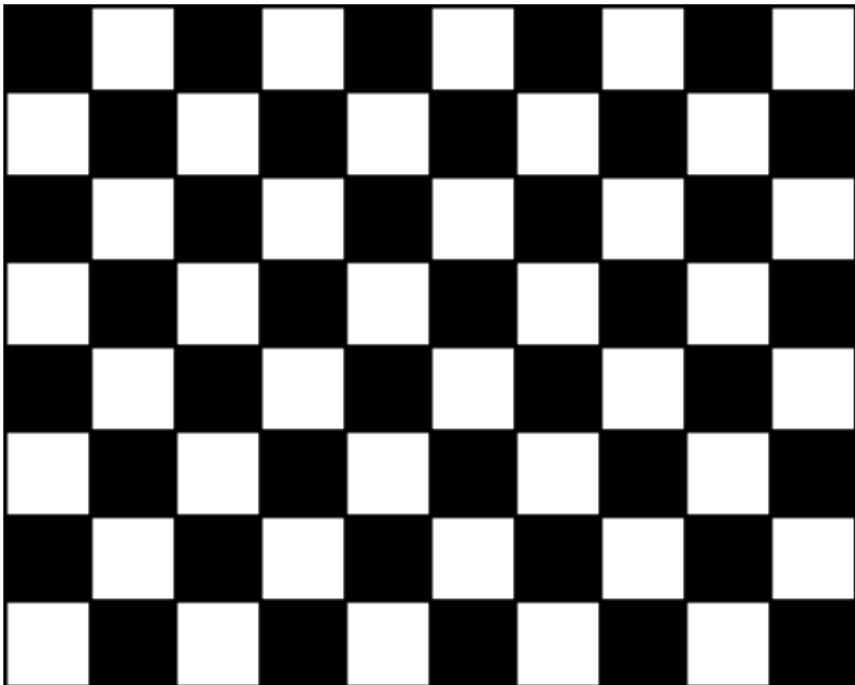
### [step2]

内円を描く処理を追加し、外円と内円に対してランダムに塗り色を指定する処理を追加してみよう。

# プログラミング演習I (第7回) 課題

## • 基本③ スケッチ名 : chess

- 白と黒の四角形をチェス盤状に並べるプログラムを作ってください。
- ウィンドウのサイズを500x400とし、1つのマスは50x50とします。
- 【左上が黒】のマスになるようにしてください。



[step1]

まずは2重の繰り返し文を使って、四角形を敷き詰めてみよう。

[step2]

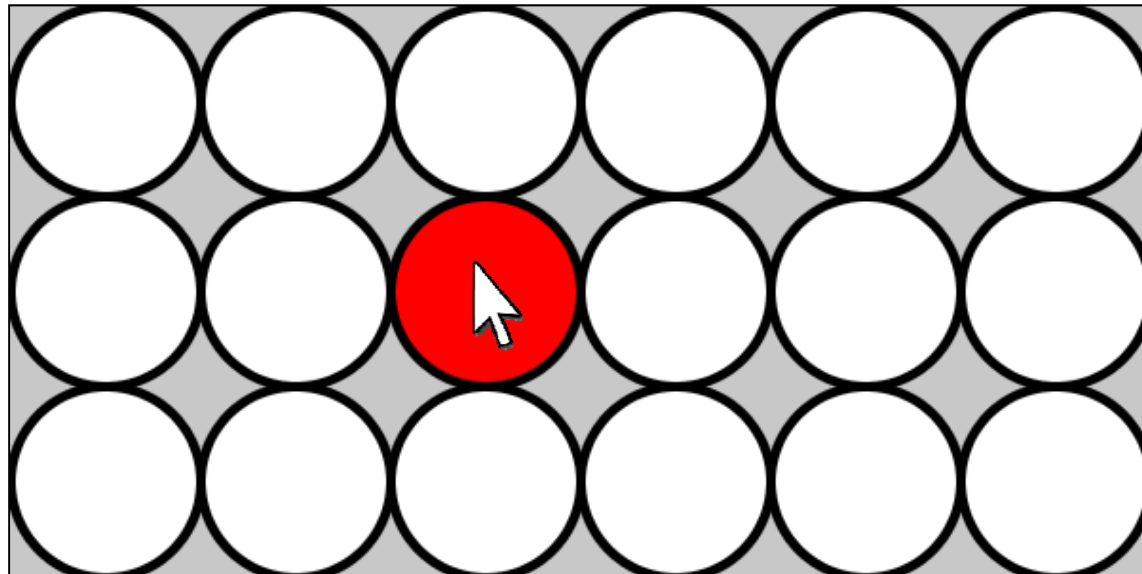
if文の分岐によって白または黒で塗り潰されるようにしてみよう。各マス目が白または黒になる条件とはなんだろうか？

# プログラミング演習I (第7回) 課題

---

## • 発展① スケッチ名: select

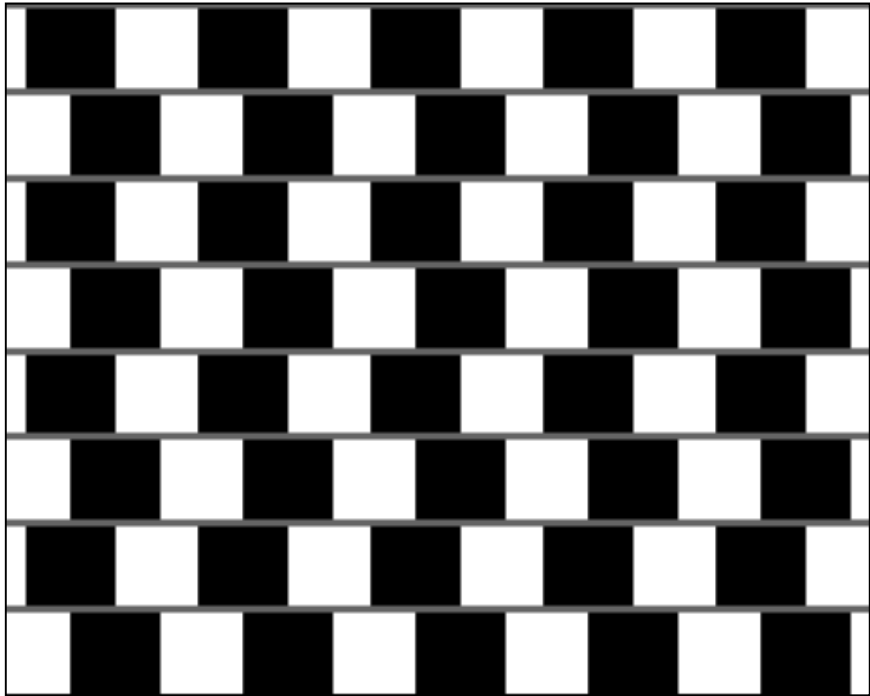
- 画面内を円で敷き詰め、カーソルがある場所の円は赤に、それ以外は白になるようにしてください。
- 直径100の円を横に6個、縦に3個並べてください。ウィンドウのサイズは600x300とします。また、円が画面内にぴったり収まるようにしてください(基本②をベースに作ろう)。



# プログラミング演習I (第7回) 課題

## • 発展② スケッチ名 : `cafeWall`

- 基本3のプログラムを改造して「カフェウォール錯視」のプログラムを作ってください。
- 参考: <http://www.brl.ntt.co.jp/IllusionForum/v/cafeWall/ja/>
- 1行おきに太さ4の灰色の横線を入れよう。



**横線が歪んで見える？**

メモ


# プログラミングの技

- 0, 1, 2, 3, 0, 1, 2, 3, 0, 1 ... と値がループするプログラムを作るにはどうしたらいいだろうか？

## ① if文を使う方法

カウンタの値を1ずつ増やしていき、3を越えたら強制的に0に戻す。

```
void setup() {  
  
}  
  
int x = 0;  
  
void draw() {  
  println(x);  
  x = x + 1;  
  if ( x>3 ) {  
    x = 0;  
  }  
}
```



# プログラミングの技

## ②「余り」を使う方法

1ずつ増えていくカウンタの値を4で割った余り  $x\%4$  を使う。

0, 1, 2, ...N でループさせたいときは、N+1で割った余りを使う。

```
void setup() {  
}  
  
int x = 0;  
  
void draw() {  
  int y = x%4;  
  x = x + 1;  
  println(y);  
}
```

このテクを知っておくと便利です



# プログラミング演習I (第7回) 課題

## • 基本① スケッチ名: table99

- 九九の表を書くプログラムを作成してください。
- ただし【奇数】のみ赤文字にしてください。

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

### [step1]

まずは2重のwhile文(またはfor文)を使って、2つの変数の値を変化させて九九の表を作ってみよう。文字が思い通りの位置に表示されるようにするにはどうしたらいいだろうか？

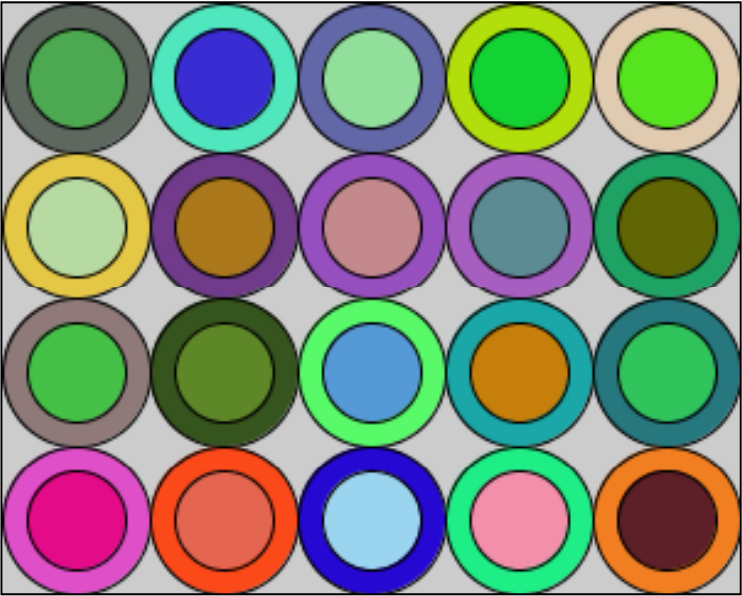
### [step2]

if文を使って各数値が奇数かどうかを調べ、条件分岐によって塗り色が変わるようにしてみよう。

# プログラミング演習I (第7回) 課題

## 基本② スケッチ名: circle2

- 画面内を2重の円で敷き詰めるプログラムを作成してください。
- 外円の直径を60、内円の直径を40とし、横に5個、縦に4個並べてください。ウィンドウのサイズは300x240とします。
- 円が画面内にぴったり収まるようにしましょう。また、塗り色は外円、内円ともにランダムにしてください。



### [step1]

まずは外円だけを指定された数だけ並べてみよう。円が欠けないようにピッタリ収まるためには、どの位置からスタートさせたらよいだろうか？

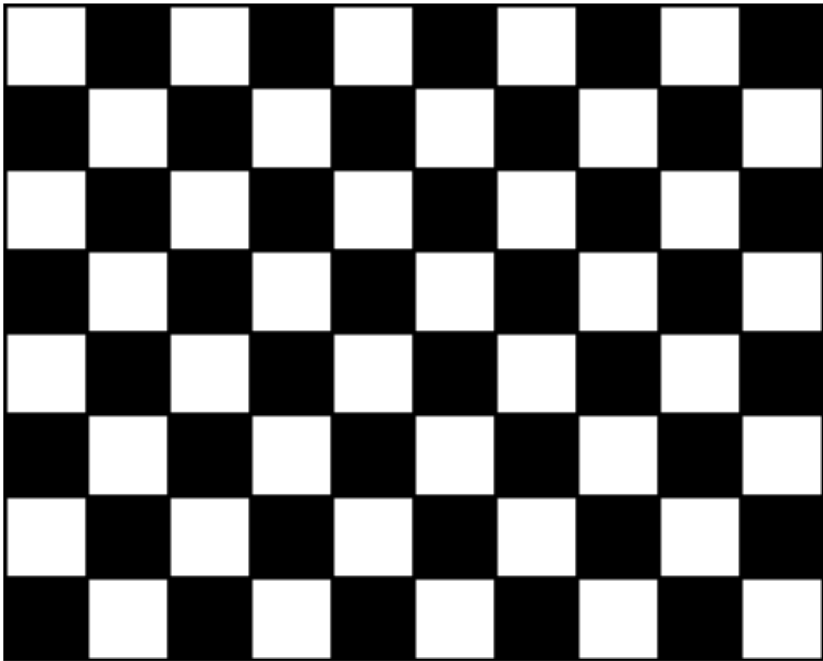
### [step2]

内円を描く処理を追加し、外円と内円に対してランダムに塗り色を指定する処理を追加してみよう。

# プログラミング演習I (第7回) 課題

## • 基本③ スケッチ名 : chess

- 白と黒の四角形をチェス盤状に並べるプログラムを作ってください。
- ウィンドウのサイズを500x400とし、1つのマスは50x50とします。
- 【左上が白】のマスになるようにしてください。



[step1]

まずは2重の繰り返し文を使って、四角形を敷き詰めてみよう。

[step2]

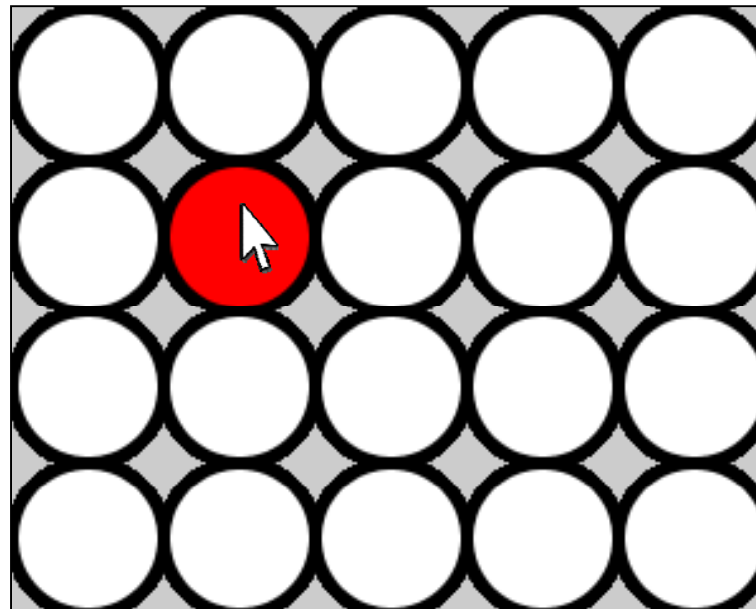
if文の分岐によって白または黒で塗り潰されるようにしてみよう。各マス目が白または黒になる条件とはなんだろうか？

# プログラミング演習I (第7回) 課題

---

## • 発展① スケッチ名: `select`

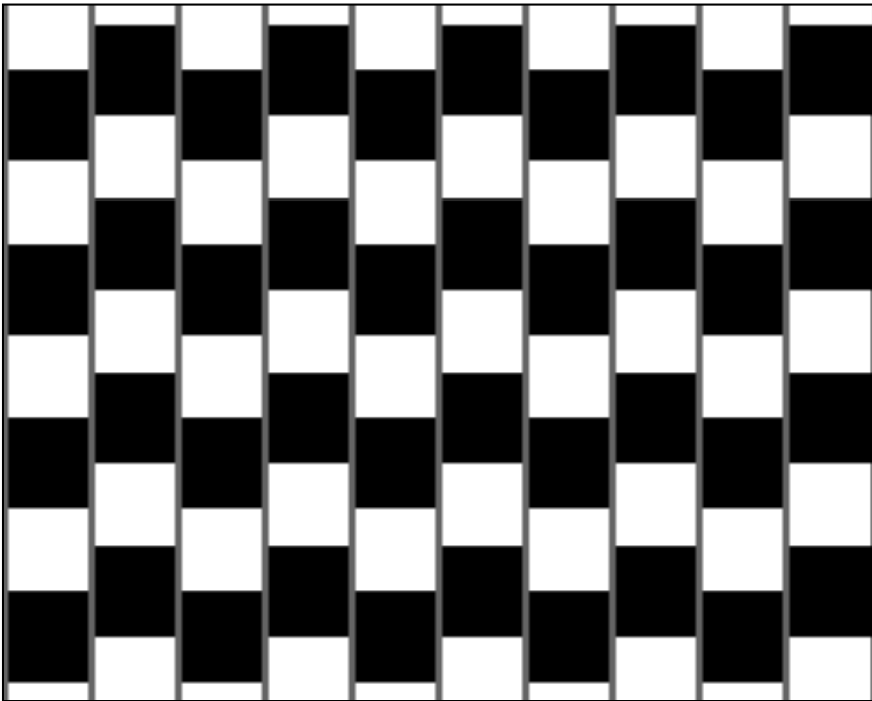
- 画面内を円で敷き詰め、カーソルがある場所の円は赤に、それ以外は白になるようにしてください。
- 直径60の円を横に5個、縦に4個並べてください。ウィンドウのサイズは300x240とします。また、円が画面内にぴったり収まるようにしてください(基本②をベースに作ろう)。



# プログラミング演習I (第7回) 課題

## • 発展② スケッチ名 : **cafeWall**

- 基本3のプログラムを改造して「カフェウォール錯視」のプログラムを作ってください。
- 参考: <http://www.brl.ntt.co.jp/IllusionForum/v/cafeWall/ja/>
- 1列おきに太さ4の灰色の縦線を入れよう。



**縦線が斜めに  
見える？**

メモ

# プログラミングの技

- 0, 1, 2, 3, 0, 1, 2, 3, 0, 1 ... と値がループするプログラムを作るにはどうしたらいいだろうか？

## ① if文を使う方法

カウンタの値を1ずつ増やしていき、3を越えたら強制的に0に戻す。

```
void setup() {  
  
}  
  
int x = 0;  
  
void draw() {  
  println(x);  
  x = x + 1;  
  if ( x>3 ) {  
    x = 0;  
  }  
}
```

# プログラミングの技

## ②「余り」を使う方法

1ずつ増えていくカウンタの値を4で割った余り  $x\%4$  を使う。

0, 1, 2, ...N でループさせたいときは、N+1で割った余りを使う。

```
void setup() {  
}  
  
int x = 0;  
  
void draw() {  
  int y = x%4;  
  x = x + 1;  
  println(y);  
}
```

このテクを知っておくと便利です