

Processing

第 10 回



松田小学校 / 寄小学校

10-ステップ 0 : 前回の復習だよ

新規のファイルを開いて、右の図に合うように空欄を埋めてみよう。

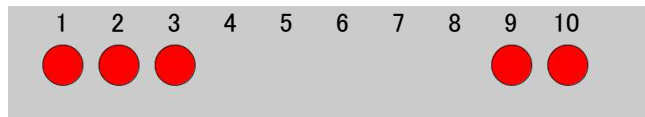
```
① size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if (  ||  ) [fill(255, 0, 0);
4     ellipse(70*x, 75, 50, 50);
5 ]
6 ]]
```



($x < 5 \parallel x > 7$)に含まれるのは、1、2、3、4、8、9、10だね。

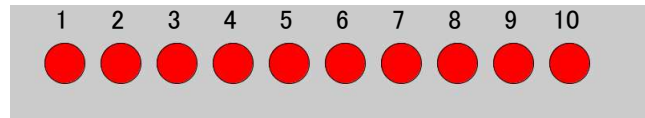
右の図に合うように、空欄を埋めてみよう。

```
② size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if (  ||  ) [fill(255, 0, 0);
4     ellipse(70*x, 75, 50, 50);
5 ]
6 ]]
```



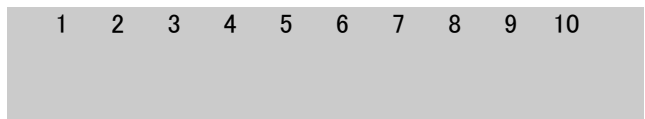
($x <= 3 \parallel x > 8$)に含まれるのは、1、2、3、9、10だね。

```
③ size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if (  ||  ) [fill(255, 0, 0);
4     ellipse(70*x, 75, 50, 50);
5 ]
6 ]]
```



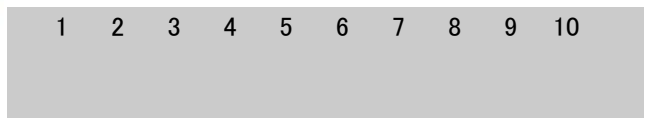
($x < 7 \parallel x > 4$)に含まれるのは、1~10の全部だね。

```
④ size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if (  &&  ) [fill(255, 0, 0);
4     ellipse(70*x, 75, 50, 50);
5 ]
6 ]]
```



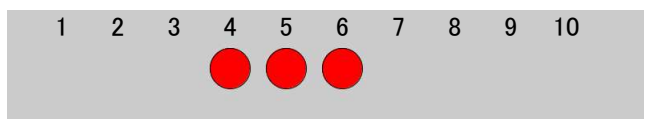
($x <= 5 \&\& x > 7$)に含まれるのは、1つもいないね。

```
⑤ size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if (  &&  ) [fill(255, 0, 0);
4     ellipse(70*x, 75, 50, 50);
5 ]
6 ]]
```



($x <= 3 \&\& x >= 8$)に含まれるのは、1つもいないね。

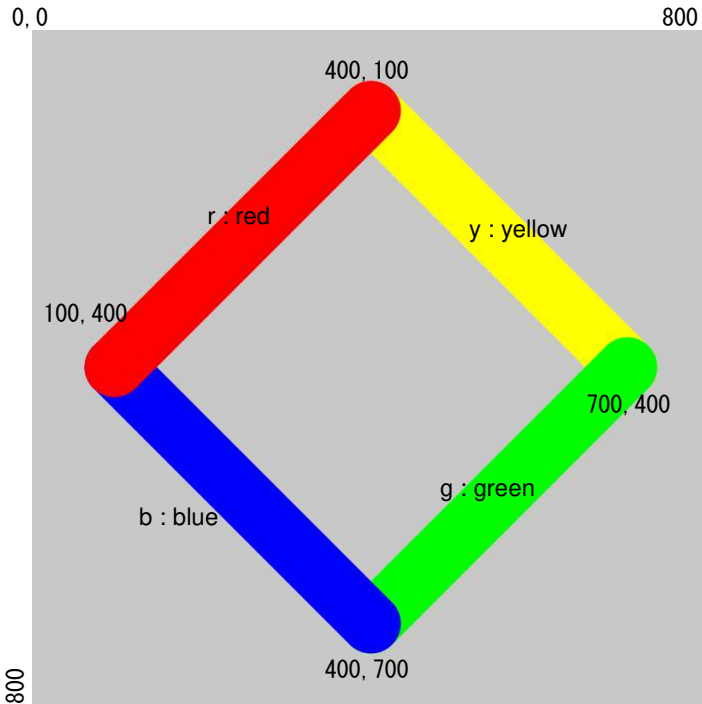
```
⑥ size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if (  &&  ) [fill(255, 0, 0);
4     ellipse(70*x, 75, 50, 50);
5 ]
6 ]]
```



($x < 7 \&\& x >= 4$)に含まれるのは、4、5、6だね。

10-ステップ 1 : 信号機へ行く前に点滅を考える

新規のファイルを開いて、下図のコードを打ってみよう。



ヒント

- ① size はいくつかな？
- ② これは4本の太い線だ。
- ③ background(); は 200 だ。
- ④ 線の太さは、すべて 70 にしている。
- ⑤ 黄色、緑色、青色、赤色の線だ。

color-line で保存しよう。

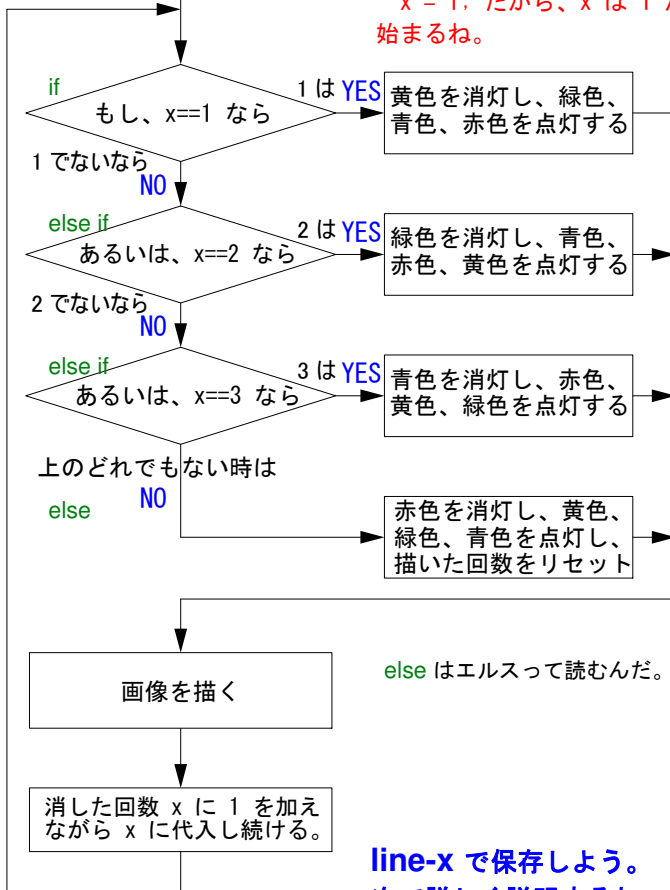
この図形を、黄色、緑色、青色、赤色の順に、消して再表示できないかな？ 第4の数字に 0 を入れて不透明にすれば、見えなくなるつまり消えるんだっけ。黄色→緑色→青色→赤色の順に消す・描くを繰り返して、いつも3本の線が残っているようにしよう。

黄色→緑色→青色→赤色の順に点滅させるには if文を使えば良いのだけれど、今までの if文とはちょっと違うよ。if だけじゃなくて、else if と else も使うんだ。考え方の流れ図を見てほしい。

プログラミングの考え方はこうだ

変数とsize、および点滅速度を決める。

点滅させることは動かすことと同じで、voidsetup と void draw を使うんだ。
x = 1; だから、x は 1 から始まるね。



line-x で保存しよう。
次で詳しく説明するね。

ミッション

1. 下の空欄を埋めてコードを完成させよう

```

int y, g, b, r, x = 1;

void setup() {
    size(800, 800);
    frameRate(1);
}

void draw() {
    if(x==1) {y=0; g=255; b=255; r=255;}
    else if(x==2) {y=255; g=0; b=255; r=255;}
    else if(x==3) {
    }
    else {y=255; g=255; b=255; r=0; x=0;}

    background(200);
    strokeWeight(70);

    stroke(255, 255, 0, y);
    line(400, 100, 700, 400);
    stroke(255, 0, 0, r);
    line(100, 400, 400, 100);

    x=x+1;
}
    
```

点滅速度

1回目は Y=0 で黄色を消灯

if の条件に合わない時、実行する

if と else if の条件に合わない時、実行する

上のどの条件にも合わない時、実行する

描いた回数をリセット!

あと2本忘れないでね!

忘れないでね!

この部分が繰り返されて点滅する。

10-ステップ 2 : 点滅の仕組みを考える

line-x を開いて、透明度について考えてほしい。透明度は、fill や stroke などの () 内に入る第4の数字だったね。0~255 であらわされ、0 は不透明だから、色を見えなくしてしまうんだ。それに対して 255 は透明度100%で、スケスケで下の色がそのまま見えるんだね。

```
stroke(255, 255, 0, y);
line(400, 100, 700, 400);

stroke(0, 255, 0, g);
line(700, 400, 400, 700);

stroke(0, 0, 255, b);
line(400, 700, 100, 400);

stroke(255, 0, 0, r);
line(100, 400, 400, 100);
```

黄色の直線が、透明度 y と変数で指定されている。ここに 0 をいれれば、不透明のカバーがかかって黄色は見えなくなる。y に 255 を入れれば、透明スクリーンになってしまうから黄色はそのまま見えるわけだね。
黄色の y に 0 を入れたときに、緑いろ、青色、赤色に 255 を入れれば、黄色だけが見えなくて、他の3色の線は見えるはずだね。同じように緑色に 0 を入れたときには、他の3色には 255 を入れてやるんだ。あとの色に対してはもう分かるよね。

```
1 int y, g, b, r, x=1;
2
3 void setup(){
4   size(800, 800);
5   frameRate(1);
6 }
7
8 void draw(){
9
10  if (x==1){y=0; g=255; b=255; r=255;}
11  else if(x==2){y=255; g=0; b=255; r=255;}
12  else if(x==3){y=255; g=255; b=0; r=255;}
13  else {y=255; g=255; b=255; r=0; x=0;}
14
15  background(200);
16  strokeWeight(70);
17
18  stroke(255, 255, 0, y); //黄色と透明度 y
19  line(400, 100, 700, 400);
20
21  stroke(0, 255, 0, g); //緑色と透明度 g
22  line(700, 400, 400, 700);
23
24  stroke(0, 0, 255, b); //青色と透明度 b
25  line(400, 700, 100, 400);
26
27  stroke(255, 0, 0, r); //赤色と透明度 r
28  line(100, 400, 400, 100);
29  x=x+1;
30 }
31
```

同じ種類の変数なら、まとめて書いてもOKだよ。

点滅の間隔を調整する。
数字が小さくなると遅くなる。

黄色が 0 で不透明の時、他は 255 で透明度100%

緑色が 0 で不透明の時、他は 255 で透明度100%

青色が 0 で不透明の時、他は 255 で透明度100%

赤色が 0 で不透明の時、他は 255 で透明度100%

x=0 は描いた回数をリセットし、最初に戻る意味

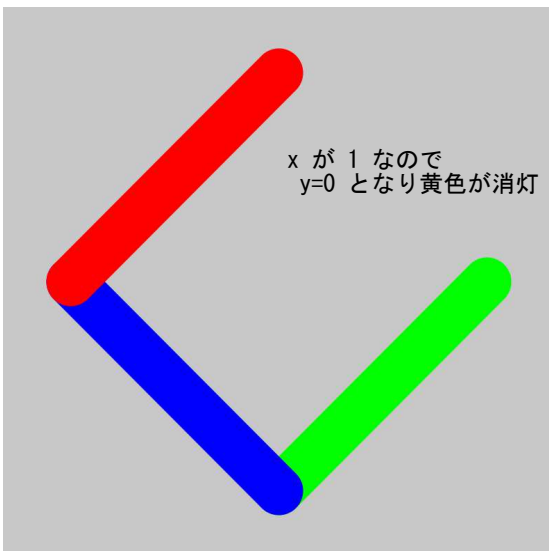
コピーして、貼り付け、
そして直した

最初の設定 x = 1 のとき、
x = x + 1 の意味は下の通りだ。

- 1回目 x 代入 = 1 ← x が 1 なので y=0 となり黄色が消灯
- 2回目 x = 1+1 = 2 ← x が 2 なので g=0 となり黄色が消灯
- 3回目 x = 2+1 = 3 ← x が 3 なので b=0 となり黄色が消灯
- 4回目 x = 3+1 = 4
x が 3 ではないので、r=0 となる。
1回転したので、x=0 としてリセットする。

コピーして、貼り付け、
そして直した

すでに保存した line-x は手直しをしているね。
ここで上書き保存(Ctrl を押しながら S を押す)をしよう。



図形を動かしたり、点滅させるには、1回だけ実行すれば良い部分と、何回も実行する部分をわける。そして、図形を描き、変化させたい部分に変数を仕込んでやる。変数を変化させて、その変化に応じて変数に数字を代入してやれば良いんだ。コードを書くのは、次のような手順だよ。

- ① 変数をいくつかから始めるか、つまり変数の初期値を図形に応じて決めてやるのが大事なんだ。
- ② どのように、void setup() { } と void draw() { } を使い分けるか。
- ③ 分からなくなったら考え方の筋道を紙に書いてみよう。
- ④ 動かす速さや点滅させる間隔は、どこで制御するのかを決めてやる。変数の範囲をかえたり、frameRate(?); の数字を調整しても変わるからね。
- ⑤ コードを実行しても走らないときには、エラーが出ていないか確認しよう。

10-ステップ 3 : 信号機を部品にばらして描く

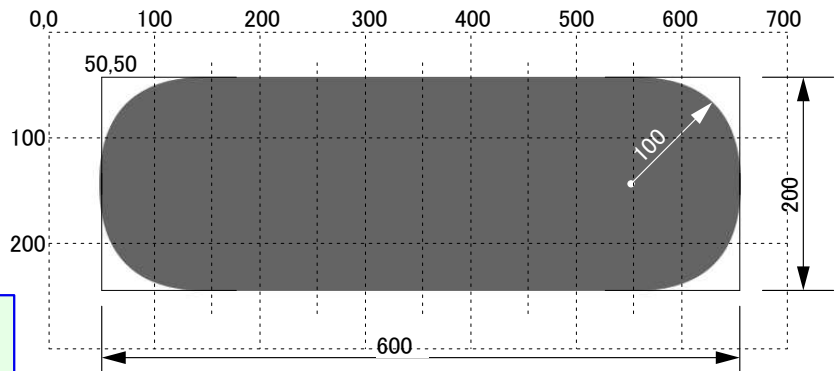
新規のファイルを開こう。プログラムを書く順番は、①信号機の背景=濃いグレーの部分を書き、次に ②信号機の光る部分を書こう。そして、最後に光る部分を点滅させよう。

① 信号機の背景は

```
1 size( , );
2 noStroke();
3 background(255);
4 fill(100);
5 rect( , , , , 100);
```

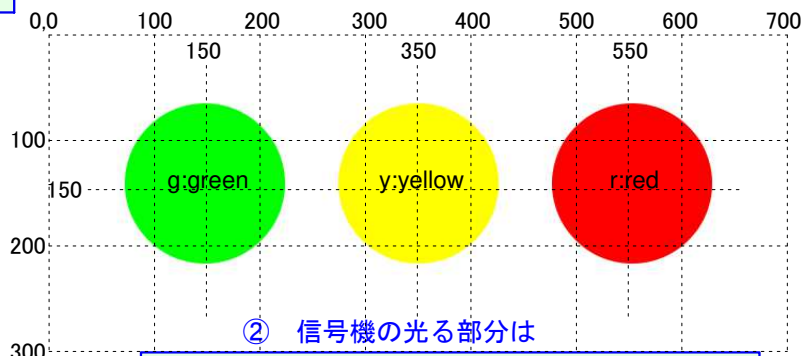
① 信号機の背景は

600*200 の四角形に第5の数字 (=100)を入れて、四隅を丸くした図形をつくった。



② 信号機の光る部分は

```
1 size(700, 400);
2 noStroke();
3 fill(0, 255, 0);
4 ellipse( , , 150, 150);
5 fill(255, 255, 0);
6 ellipse(350, 150, 150, 150);
7 fill(255, 0, 0);
8 ellipse( , 150, 150, 150);
```



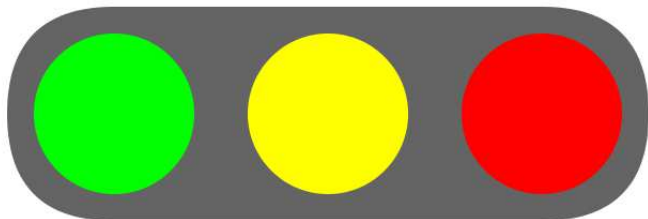
② 信号機の光る部分は

繰り返し(for)を使わずに、直径 150 の円を3つ横に並べて書いている。

ミッション

信号機の背景と、信号機の光る部分のコードとを合体させて、右のような図を描いてみよう。

できたら、signal-1 で保存しよう



合体した信号機のコード

```
1 size(700, 300);
2 noStroke();
3 background(255);
4 fill(100);
5 rect(50, 50, 600, 200, 100);
6
7 fill(0, 255, 0); // 緑色
8 ellipse(150, 150, 150, 150);
9 fill(255, 255, 0); // 黄色
10 ellipse(350, 150, 150, 150);
11 fill(255, 0, 0); // 赤色
12 ellipse(550, 150, 150, 150);
```

③ 信号機を点滅させるには

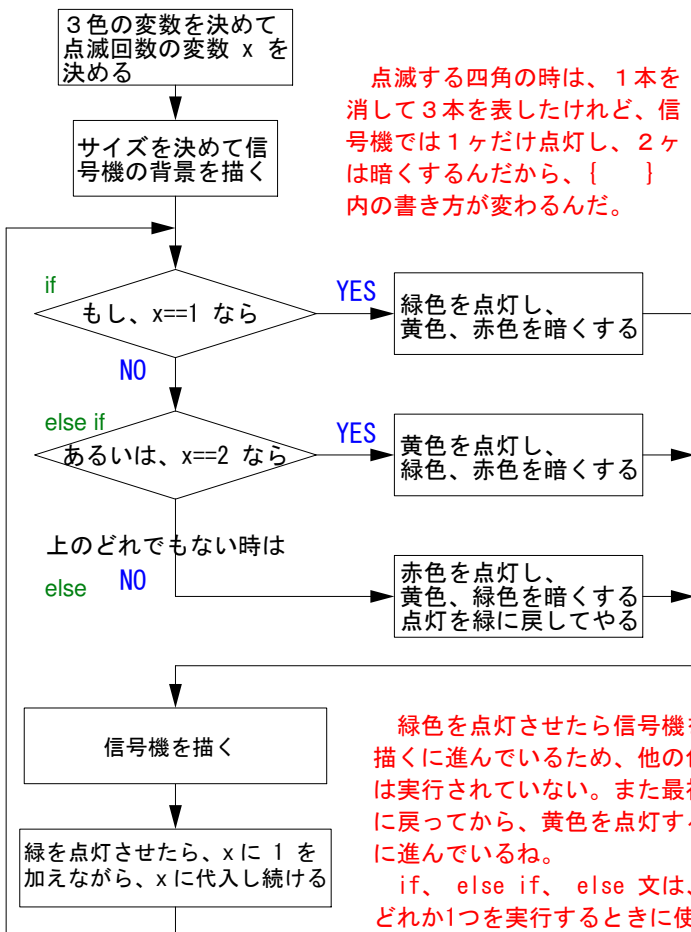
点滅とは点いたり消えたりすることだよ。信号機の場合は、消えたと言っても見えなくなるわけではないよ。そこで考えなければいけないのが透明の程度なんだ。緑色が点いている間は、黄色と赤色には半透明のカバーをかけてやれば、明るい緑色に暗い黄色と赤色が見えて、進めになるわけだから、50 が良いか 100 が良いか考えよう。
透明度の変数を g, y, r として size(700, 400); の前に宣言する。次に、左のコードの ↓ のところに、第4の各変数を入れてやる。そしてもう1つ、時間の経過を変数 x として宣言し、これを if文で制御すれば良いんだ。

点滅させるには if文を使うんだけど、今までとはちょっと違う構文だよ。菱形(ひしがた)を点滅させた if、else if、else 文を使うんだ。

10-ステップ 4 : 信号機を点滅させる

点滅させるには if文でも、if だけではなく else if と else を入れた構文を使うんだ。

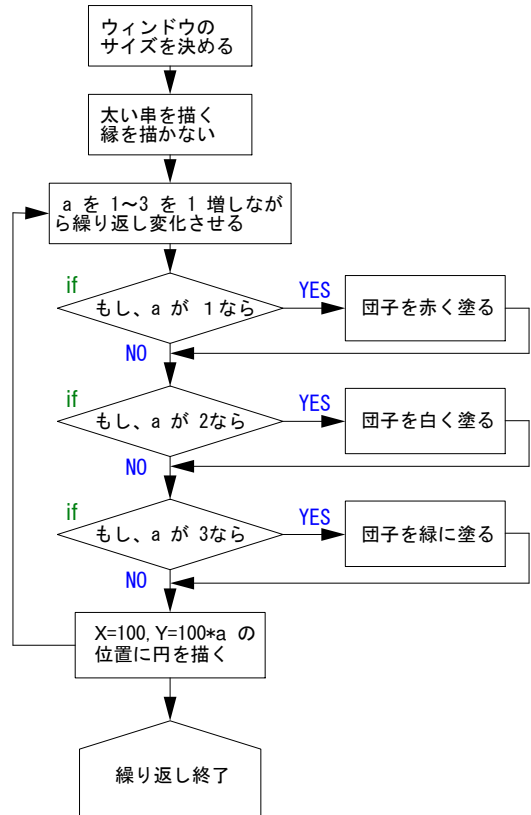
プログラミングの考え方はこうだ



点滅する四角の時は、1本を消して3本を表したけれど、信号機では1ヶだけ点灯し、2ヶは暗くするんだから、{ } 内の書き方が変わるんだ。

緑色を点灯させたら信号機を描くに進んでいるため、他の色は実行されていない。また最初に戻ってから、黄色を点灯するに進んでいるね。
if、else if、else 文は、どれか1つを実行するときを使うんだ。

三色団子のときの、プログラミングの考え方



各団子に色を塗った後、毎回主な流れに戻っているんで、3つとも色が塗られている。つまり、すべて実行されたわけだね。
全部を実行するときには、if 文はすべて同じで if(a==?) を使うんだ。

```

1 int g, y, r, x=1;
2
3 void setup() {
4   size(700, 300);
5   frameRate(0.5);
6   noStroke();
7 }
8
9 void draw() {
10  background(255);
11  fill(100);
12  rect(50, 50, 600, 200, 100);
13
14  if ( x==1 ) {g=255; y=50; r=50;}
15  else if( x==2 ) {g=50; y=255; r=50;}
16  else {g=50; y=50; r=255; x=0;}
17
18  fill(0, 255, 0, g); //緑色
19  ellipse(150, 150, 150, 150);
20  fill(255, 255, 0, y); //黄色
21  ellipse(350, 150, 150, 150);
22  fill(255, 0, 0, r); //赤色
23  ellipse(550, 150, 150, 150);
24  x=x+1;
25 }

```

小数を使って点滅をゆっくりさせる。

条件は変化させる図形の前に書くのだ。

変化する(点滅する)図形

x = 1 のとき、x = x + 1 の意味は下の通りだよ。

- 1回目 x 代入 = 1 ← x が 1 なので、g=255 となり緑色が点灯
 - 2回目 x = 1+1 = 2 ← x が 2 なので、y=255 となり黄色が点灯
 - 3回目 x = 2+1 = 3
- x が 2 ではないので、r=255 となり赤色が点灯する。緑色、黄色、赤色と一巡したから、x=0 で緑色に戻る。

信号機の点滅では、x==1 で緑色を点灯し、黄色と赤色を暗くしたら、そのまま一番下まで来て信号機を描いて最初に戻ってしまう。同じように x==2 なら黄色を点灯し、緑色と黄色を暗くしたら最初に戻るんだ。そして、x が 2 ではない時は赤色を点灯させて、一巡したので緑色に戻るんだ。
x=0; とは一巡したという意味なんだ。矢印の流れを追って確認してね。

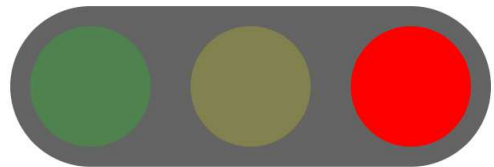
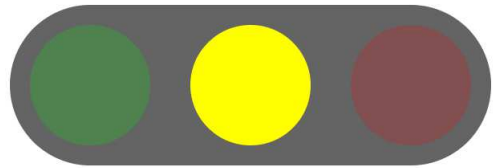
できたら、signal-1 を上書き保存しよう。

10-ステップ 5 : 信号器の点滅を止める

信号機はお巡りさんが一時止めることがあるね。そこで一時止めるコードを考えてみよう。このコードは一寸変わっているけど、簡単だよ。

```
1 int g, y, r, x=1;
2
3 void setup() {
4   size(700, 300);
5   frameRate(0.5);
6   noStroke();
7 }
8
9 void draw() {
10  background(255);
11  fill(100);
12  rect(50, 50, 600, 200, 100);
13
14  if ( x==1 ) {g=255; y=50; r=50;}
15  else if( x==2 ) {g=50; y=255; r=50;}
16  else {g=50; y=50; r=255; x=0;}
17
18  fill(0, 255, 0, g); //緑色
19  ellipse(150, 150, 150, 150);
20  fill(255, 255, 0, y); //黄色
21  ellipse(350, 150, 150, 150);
22  fill(255, 0, 0, r); //赤色
23  ellipse(550, 150, 150, 150);
24  x=x+1;
25 }
26
27 void mousePressed() {
28   noLoop(); //ボタンを押すと点滅が停止
29 }
30
31 void mouseReleased() {
32   loop(); //ボタンを離すと点滅が再開
33 }
```

これは signal-1 として
保存したコードだよ。



ここまでは signal-1 として
保存したコードだよ。

信号機の点滅を止めるには

一番最後の行に void mousePressed() { } を書いて { } のなかに noLoop(); と書けばOKだ。再開も簡単で、void mouseReleased() { } を書いて、{ } のなかに loop(); と書けば用意万端。実行したあと、点滅している所でマウスを押すと点滅が一時停止するよ。

!! これだけ大文字だよ

Release とは放すという意味だよ。

loop とは輪をつくる意味で、nolop
といえは輪をほどくって感じかな。

signal-final で保存しよう。

送る言葉

ここまでやってきた皆は、プロセッシングでプログラムを書く実力が付いているよ。これからは自力でコードを書いていこう。一緒にプロセッシングを学ぶことができて楽しかったよ。ボランティア・講師一同から感謝の言葉をおくります。

Bye-bye!