

課題の提出方法

- プログラムの冒頭にコメントを必ず書くこと！
 - プログラム名とその目的(日本語でも英語でもOK)
 - プログラムの作成日(例: 2021/05/10)
 - プログラムの作成者(年-組-番号-名前)
 - 追加で他の情報を記載してもOK

```
// Olympic.pde
// Draw the Olympic rings
// 2021/05/10
// 1-3-99 Satoshi Nakamura
background( 255, 255, 255 );
fill( 255, 0, 0 );
:
```

修正：課題の評価について

- 基本課題は、講義中の指定時間までに提出必須
 - それ以降の提出は採点対象外となります
- 発展課題は、講義中の指定時間までに提出したものを100%として、指定時間以降、次の講義開始までに提出されたものを50%として評価
 - なので、発展課題ができていなくても、次回までに提出したら50%分は点が付きます！
 - 他人に見せない！見せてもらわないように！

今日のテクニック

ランダムな値を発生したいときは random を利用しよう！

0以上で10より小さい値をランダムに発生する方法は

```
float num = random( 10 );
```

0以上で5より小さい整数値 (0, 1, 2, 3, 4) を同一の確率でランダムに発生する方法は...

```
int num = (int)random( 5 );
```

(0.000... ~ 4.9999... の値が生成されそれを整数化)

10以上20以下の整数値をランダムに発生する方法は

```
int num = (int)random( 10, 21 );
```

プログラミング演習I (第4回) 課題

• 基本課題① スケッチ名: basic_Dice

- 400x300のウィンドウを作成し, そのウィンドウ内をクリックするたびに, Aさんが1~8の目が均等に出るサイコロ, Bさんが1~6の目が均等に出るサイコロを振った結果を示すとともに, どちらが勝ったか(または引き分けたのか)を下図のように標準出力するプログラムを作成せよ
- 値をランダムに生成するには, random を使えば良い

クリック!

Aさん: 7
Bさん: 4
Aさんの勝ち

クリック!

Aさん: 7
Bさん: 4
Aさんの勝ち
Aさん: 2
Bさん: 4
Bさんの勝ち

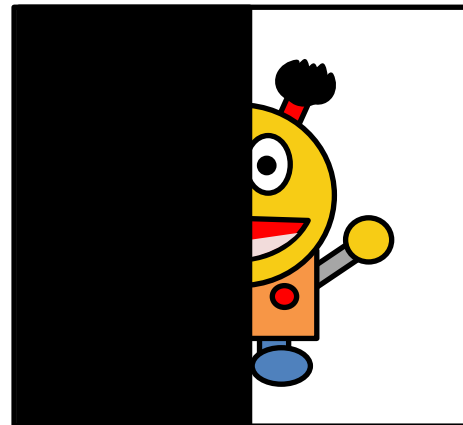
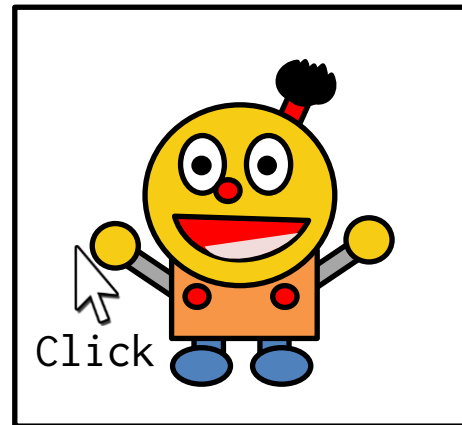
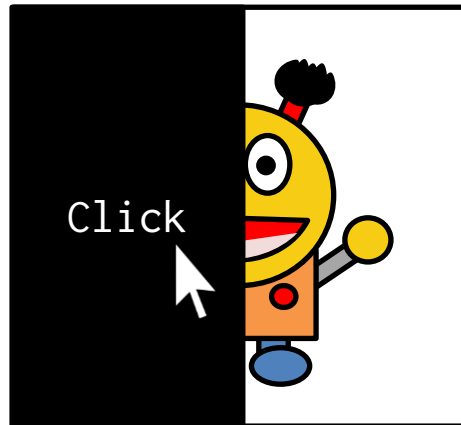
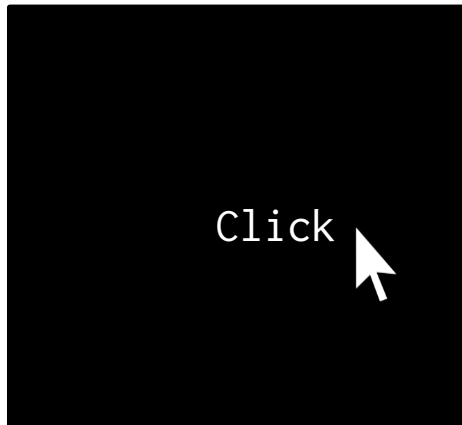
クリック!

Aさん: 7
Bさん: 4
Aさんの勝ち
Aさん: 2
Bさん: 4
Bさんの勝ち
Aさん: 4
Bさん: 4
引き分け

プログラミング演習I (第4回) 課題

• 基本課題② スケッチ名：`basic_Switch`

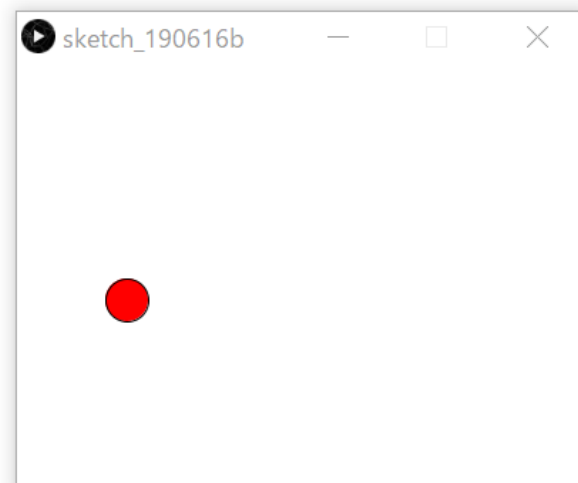
- 500x500のウィンドウを作成し、全体を黒く塗りつぶせ。
- 画面の左側をクリックすると、画面の左側の塗りつぶしをやめ、後ろにキャラクターが表示されるようにせよ、左側をもう一度クリックすると黒く塗りつぶすようにせよ。
- 画面の右側についても同じように動作するようにせよ。
- 両方が奇数回ずつクリックされている場合に、キャラクターの全体が見えることになる。
- 前に作ったdrawCharacterを使って実現せよ



プログラミング演習I (第4回) 課題

• 基本課題③ スケッチ名：**basic_ColorfulBall**

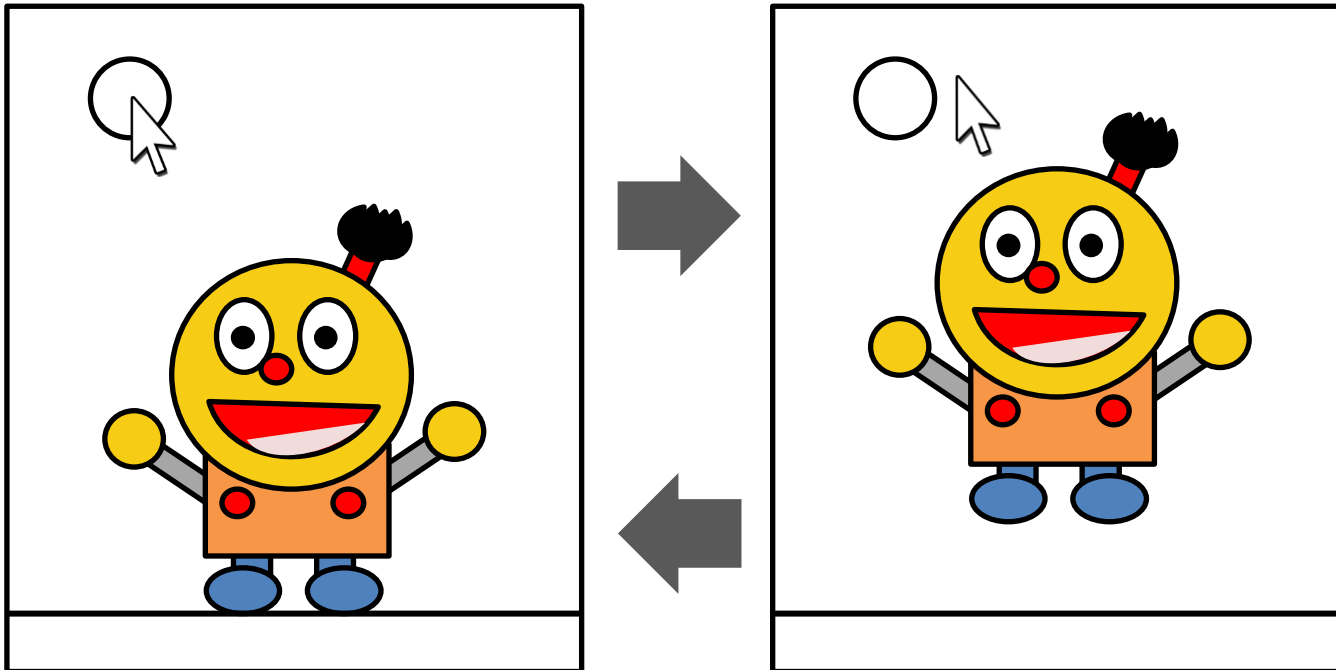
- 400x300のウィンドウ内を, 毎フレームX方向に3ピクセル, Y方向に2ピクセル動く円を作成せよ
- 円の初期位置は画面内でランダムな位置にせよ
- 円は上下左右の端まで来ると跳ね返るようにせよ (円の中心部が端に来た時に跳ね返るようにするだけでよい)
- 円の塗りつぶし色は, 跳ね返るたびに「緑→黄→赤→緑→黄→赤→」と変化するようにせよ



プログラミング演習I (第4回) 課題

発展課題① スケッチ名: advanced_Charajump

- 以前作成したdrawCharacter(int cx, int cy)を利用し、キャラクターがジャンプするプログラムを作りたい (ellipseなどのように使えます)
- キャラクターと地面となる面を表示し、画面左上に用意した直径50以下の丸ボタンをクリックすることでキャラクターがジャンプするようなプログラムを作成せよ。
- ジャンプについては初速をわかる程度に設定しつつ鉛直上方投射するものとし、地面にくると止まるようにせよ



プログラミング演習I (第4回) 課題

• 発展課題② スケッチ名 : advanced_SwipeUnlock

- 縦長のウィンドウの下部に左の方に丸型のものを用意し、それを右側にスライド（スワイプ）させると、ロックが解除され何らかの画面に遷移する仕組みを実現せよ（例えば、作成したキャラクタが描画されている画面など）
- なお、右端まで移動せずに手を離れた場合は、5ピクセル/フレームの速度で最初の位置に戻るようにせよ。また、指定の位置より右や左にはみ出ないようにせよ。

