明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


# プログラミング演習 (5)
# 条件分岐 (2)

中村，高橋
小林，橋本

# 目標

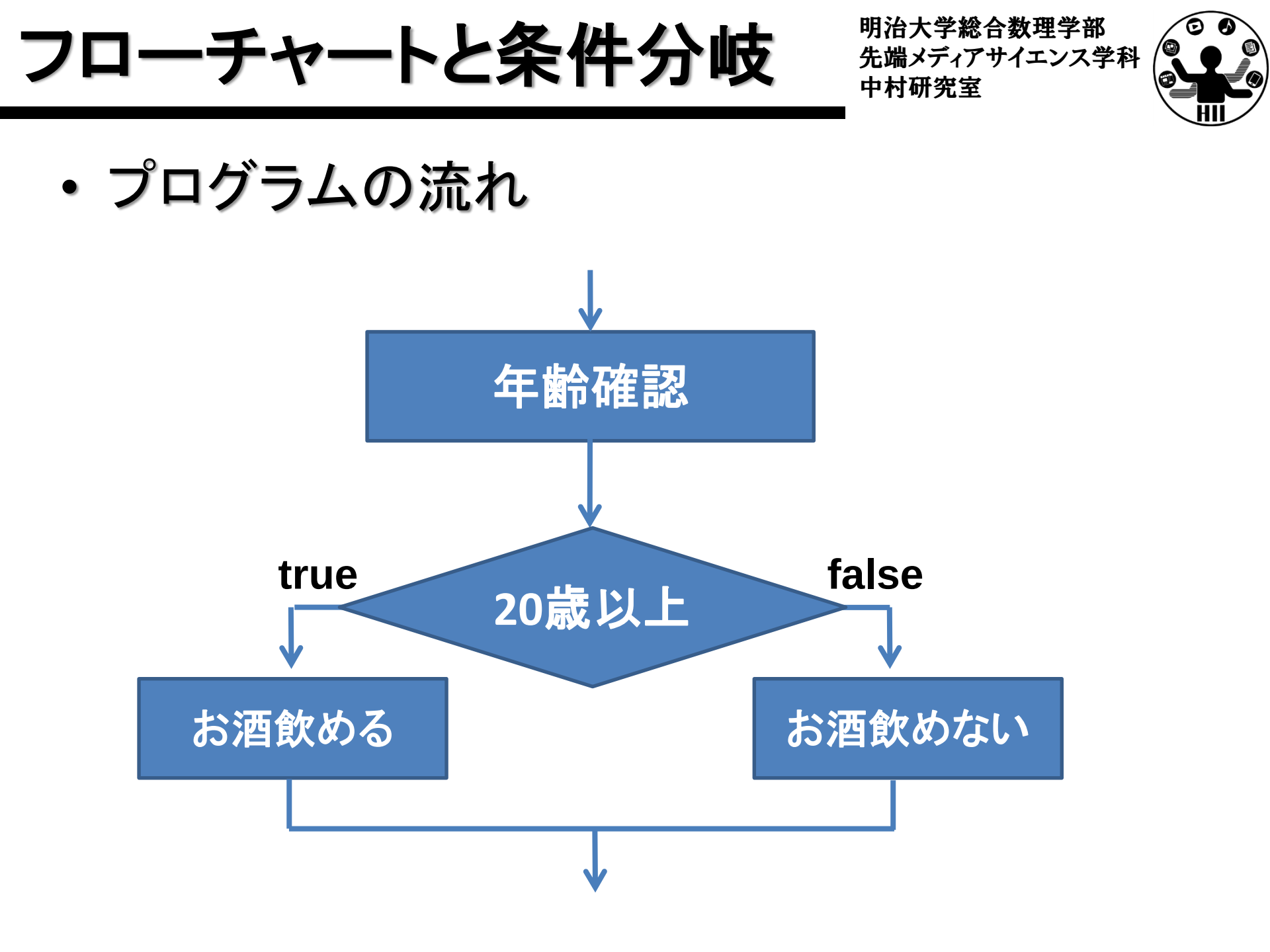
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


- Processing で当たり判定に挑戦！
  - 条件分岐を理解する
  - 何らかの条件を満たした時に色を変える！
  - マウスカーソルと動いている円がぶつかったら終了
  - シューティングゲームやもぐらたたきに挑戦！

- 課題:
  - Processing でゲームを作ろう！
  - 占いを作ってみよう

# フローチャートと条件分岐

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室




- プログラムの流れ

# 制御文

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
if( 条件A ){
    // 条件Aの時の処理内容
} else if( 条件B ){
    // 条件Aでなく，条件Bの時の処理内容
} else {
    // 条件AおよびB以外の時の処理内容
}
```

**よくあるミス**
**if( 条件 );{ 条件A の処理 }**

# 論理値，真偽値（boolean）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- true か false かを値として持つ
  - それ以外の値は持たない
- 制御文で「条件Aを満たす時」というのは「条件Aが真（true）である」と同じ意味
- 制御文で「条件Aを満たさない時」というのは「条件Aが偽（false）である」と同じ意味
  - x > y の条件をみたす場合，x > y は true，みたさない場合は false
  - x == y は x と y が同じ値の場合 true に，違う値の場合に false となる

# 条件の記述方法

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


| 演算子 | 意味 | プログラム上 |
|---|---|---|
| x > y | x が y より大きい | 左記の時に true<br>それ以外で false |
| x < y | x が y より小さい | 同上 |
| x >= y | x が y 以上 | 同上 |
| x <= y | x が y 以下 | 同上 |
| x == y | x と y が等しい | 同上 |
| x != y | x と y が等しくない | 同上 |
| !x | x は false | 同上 |

# ～かつ～はどうするのか？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 論理積演算子と論理和演算子
- ～ かつ ～ のとき　　&&
- ～ または ～ のとき　　||

~~if( 200 < x < 400 )~~

```
if ( x > 200 && x < 400 ){
   x が200より大きく，400より小さい時はここに来る
}
if ( n < -10 || n > 10 ){
   nが-10より小さいか，10より大きければここに来る
}
```

# ～かつ～はどうするのか？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
if ( x > 200 && x < 400 ){
    x が200より大きく，400より小さい時はここに来る
}
```

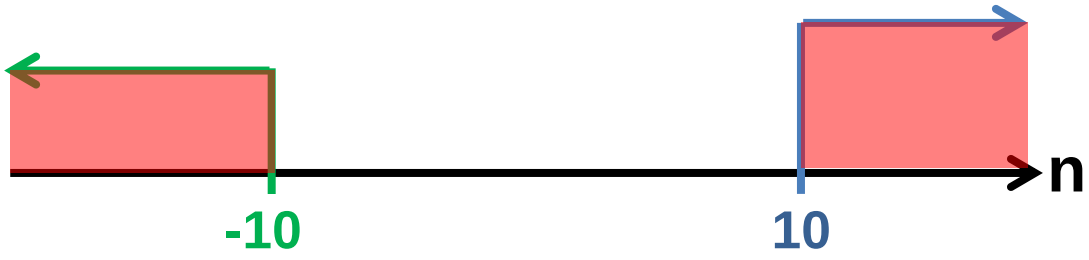

詳しく説明すると・・・

- x = 100 のときは，「x>200がfalse」で「x<400がtrue」となるため，「false && true」となり，「false」となる
- x = 300 のときは，「x>200がtrue」で「x<400がtrue」となるため，「true && true」となり，「true」となる
- x = 500 のときは，「x>200がtrue」で「x<400がfalse」となるため，「true && false」となり，「false」となる

# ～かつ～はどうするのか？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
if ( n < -10 || n > 10 ){
    nが-10より小さいか，10より大きければここに来る
}
```




詳しく説明すると・・・

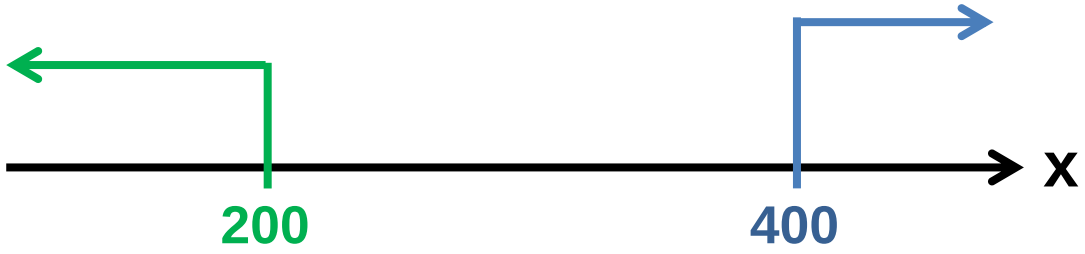
- n = -20 のときは，「n<-10がtrue」で「n>10がfalse」となるため，「true || false」となり，「true」となる
- n = 0 のときは，「n<-10がfalse」で「n>10がfalse」となるため，「false || false」となり，「false」となる
- n = 20 のときは，「n<-10がfalse」で「n>10がtrue」となるため，「false || true」となり，「true」となる

# 色々とつまずくポイント

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


## true に何時まで経ってもならない例

```
if( x < 200 && x > 400 ){
    // だめ
}
```




## ずっと true になってしまう例

```
if( n > -10 || n < 10 ){
    // だめ
}
```

# (Q) 占い

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

プログラムを起動したときに，0～9までの値を作成し，その値に応じて占いの結果を表示せよ

0～3の時は「凶」

4～6の時は「吉」

7～9の時は「大吉」

と表示するようにせよ

0から9までの数字を出力する場合は…

```
int kuji = (int)random( 0, 10 );
```

# (A) 占い

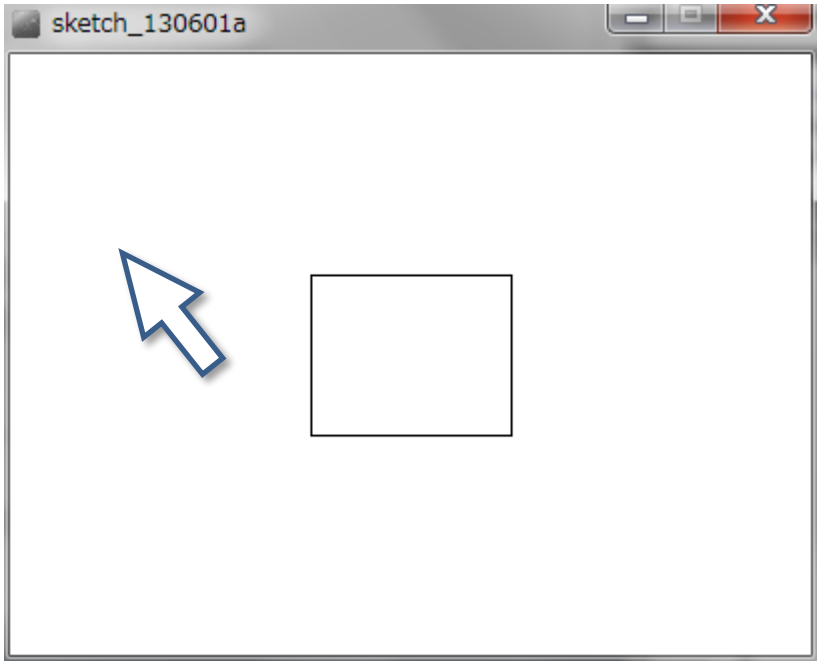
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

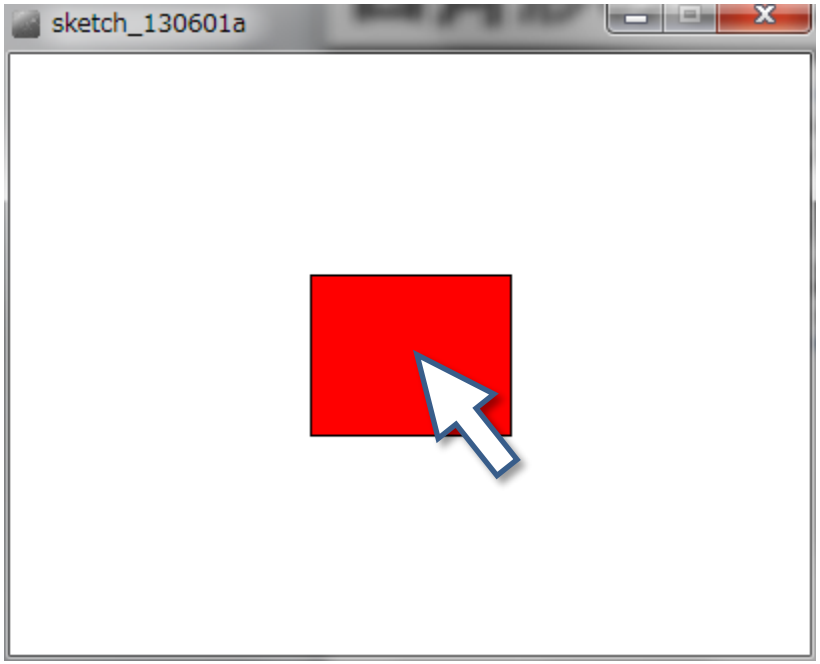

```
// kujiをひく
int kuji = (int)random( 0, 10 );

// 0-3, 4-6, 7-9
if( kuji >= 0 && kuji <= 3 ){
  println( "凶" );
} else if( kuji >= 4 && kuji <= 6 ){
  println( "吉" );
} else {
  println( "大吉" );
}
```

# ボタン(四角形)の判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 400x300のウインドウの中央に表示された横100，縦80のボタンの上にカーソルがあるとボタンを赤色に，そうでなければ白色にするには？

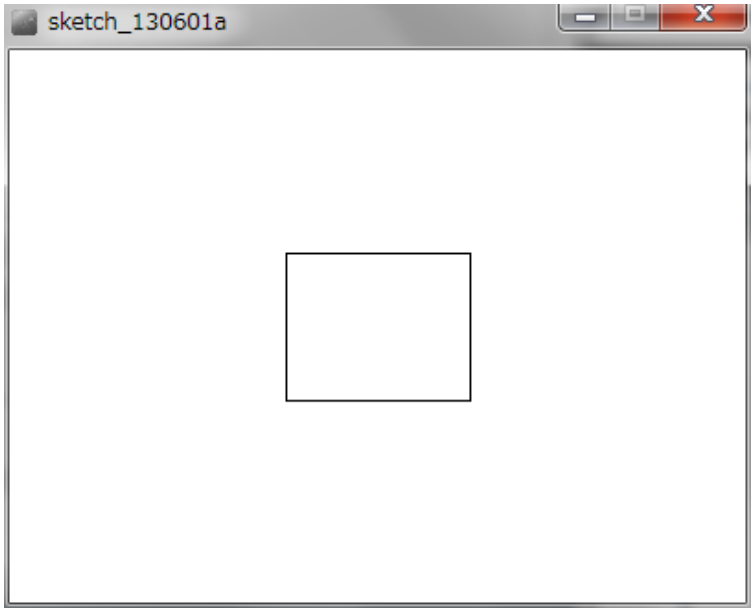


# ボタン(四角形)の判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

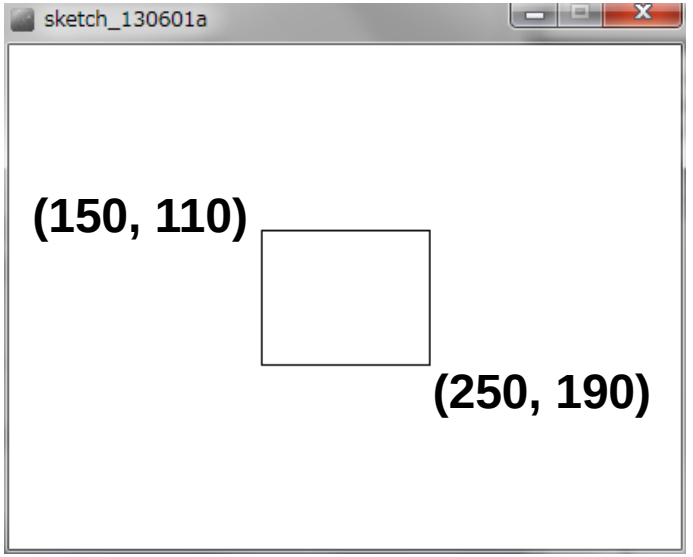
- 考え方
  - 画面の中央は (200, 150)
  - ボタンの左上と右下の座標は？？
  - ボタンは rect( 左上x, 左上y, 横幅, 縦幅 ); で描画
  - マウスカーソルの座標は (mouseX, mouseY)



# ボタン(四角形)の判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


- 中央(200, 150)で，横幅100, 縦幅80なので
  - 左上の座標は(200-100/2, 150-80/2) = (150, 110)
  - 右下の座標は(200+100/2, 150+80/2) = (250, 190)
- 150≦mouseX ≦250 かつ 110≦mouseY≦190なら赤色，そうでなければ白色で塗りつぶす
  - つまり，mouseX>=150 かつ mouseX<=250 かつ mouseY>=110 かつ mouseY<=190 の時！
- 「かつ」は，「&&」で表現する



# ボタン(四角形)の判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

（注意）150 <= mouseX <= 250 はダメ！

150 <= mouseX && mouseX <= 250 に分解

```
void setup(){
   size( 400, 300 );
}

void draw(){
   background( 255, 255, 255 );
   if( mouseX >= 150 && mouseX <= 250 && mouseY >= 100 && mouseY <= 190 ){
       fill( 255, 0, 0 );
   } else {
       fill( 255, 255, 255 );
   }
   rect( 150, 110, 100, 80 );
}
```

# ボタン（四角形）の判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


## if や else if，else の{}の中に if を入れてもOK!!（多段階の条件分岐）

```
void draw(){
    background( 255, 255, 255 );
    if( mouseX >= 150 && mouseX <= 250 ){
        if( mouseY >= 100 && mouseY <= 190 ){
            fill( 255, 0, 0 );
        } else {
            fill( 255, 255, 255 );
        }
    } else {
        fill( 255, 255, 255 );
    }
    rect( 150, 110, 100, 80 );
}
```

さらに110≦mouseY≦190
ならここに入ってくる

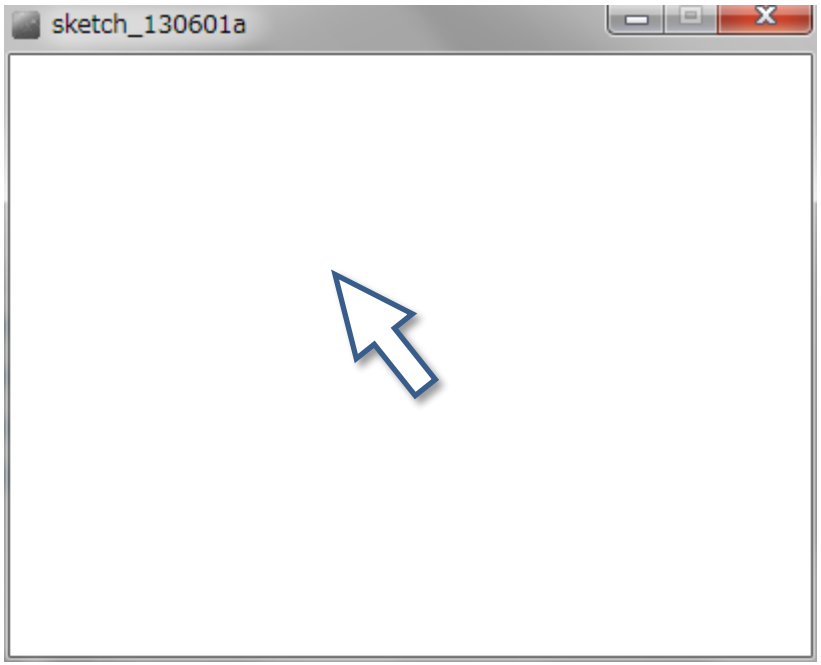
150≦mouseX≦250なら
ここに入ってくる

さらに110≦mouseY≦190
でない場合はここに入ってくる

# 宝探し(四角形を探す)

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) ランダムな場所に，縦幅30x横幅20の大きさで配置された四角形を探す(カーソルがその上にあるときだけ表示する)プログラムはどう作るか？




# 宝探し(四角形を探す)

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - 四角形の左上の座標を実数の変数に
    - `float  leftTopX;`  // 他の変数名でもOK
    - `float  leftTopY;`  // hidariueX, hidariueY でもOK
  - 四角形の左上の座標をランダムに決める
    - `random( 最小値, 最大値 );` で, 最小値から最大値までの値を求めることができる
      - `leftTopX = random( 50, 350 );` // 50～350のランダムな値

      などのように･･･

# 宝探し(四角形を探す)

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方(続き)
  - mouseX がどういった条件を満たせば表示する？
    - ➔ leftTopX ≦ mouseX ≦ leftTopX+30
  - mouseY がどういった条件を満たせば表示する？
    - ➔leftTopY ≦ mouseY ≦ leftTopY+20
  - 2つの条件を満たした時に長方形を表示する！
    - 長方形は rect( 左上x, 左上y, 横幅, 縦幅 ); で描画

# 宝探し(四角形を探す)

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
float leftTopX;
float leftTopY;

void setup(){
    size( 400, 300 );
    leftTopX = random(0, 390); // 0～390までの任意の値
    leftTopY = random(0, 290); // 0～290までの任意の値
}

void draw(){
    background(255,255,255);
    if( mouseX >= leftTopX && mouseX <= leftTopX+30 ){
        if( mouseY >= leftTopY && mouseY <= leftTopY+20 ){
            rect( leftTopX, leftTopY, 30, 20 );
        }
    }
}
```

400までにしちゃうとはみ出て見つからない

# 宝探し(四角形を探す)

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
float leftTopX;
float leftTopY;

void setup(){
    size( 400, 300 );
    leftTopX = random(width-10);
    leftTopY = random(height-10);
}

void draw(){
    background(255,255,255);
    if( mouseX >= leftTopX && mouseX <= leftTopX+30
        && mouseY >= leftTopY && mouseY <= leftTopY+20 ){
         rect( leftTopX, leftTopY, 30, 20 );
    }
}
```

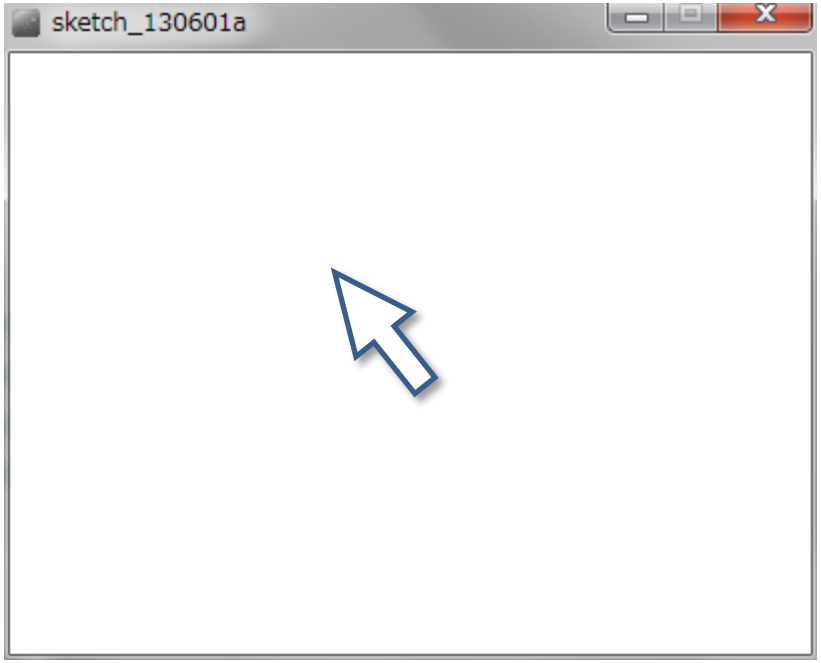
もちろん全部&&でつないでもOK

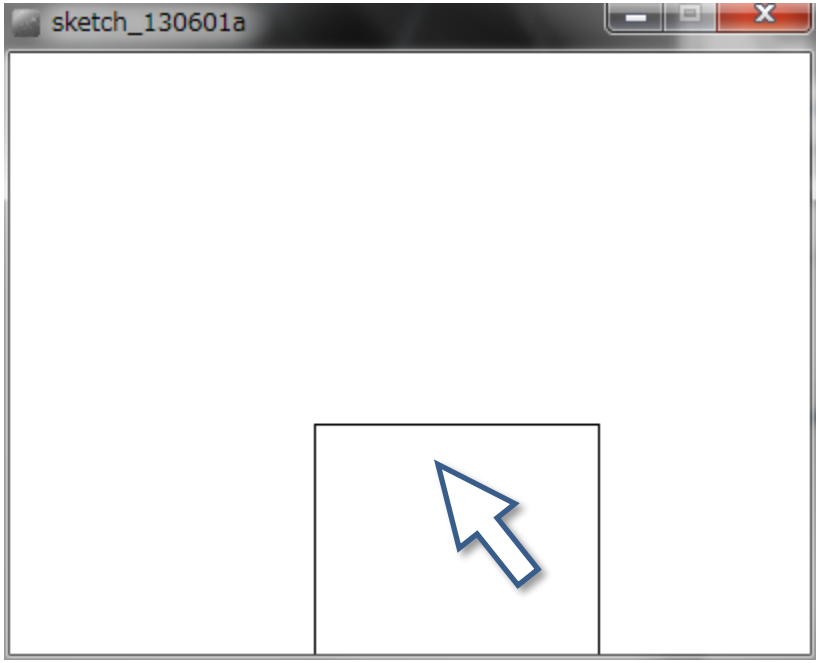
2行で書いてもOK!

# 演習

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ランダムな場所に，ランダムな大きさで配置された四角形を探す（カーソルがその上にあるときだけ表示する）プログラムを作成しよう！
  - 四角形の左上の座標，縦幅横幅を実数の変数に（leftTopX, leftTopY, rectWidth, rectHeightなど）

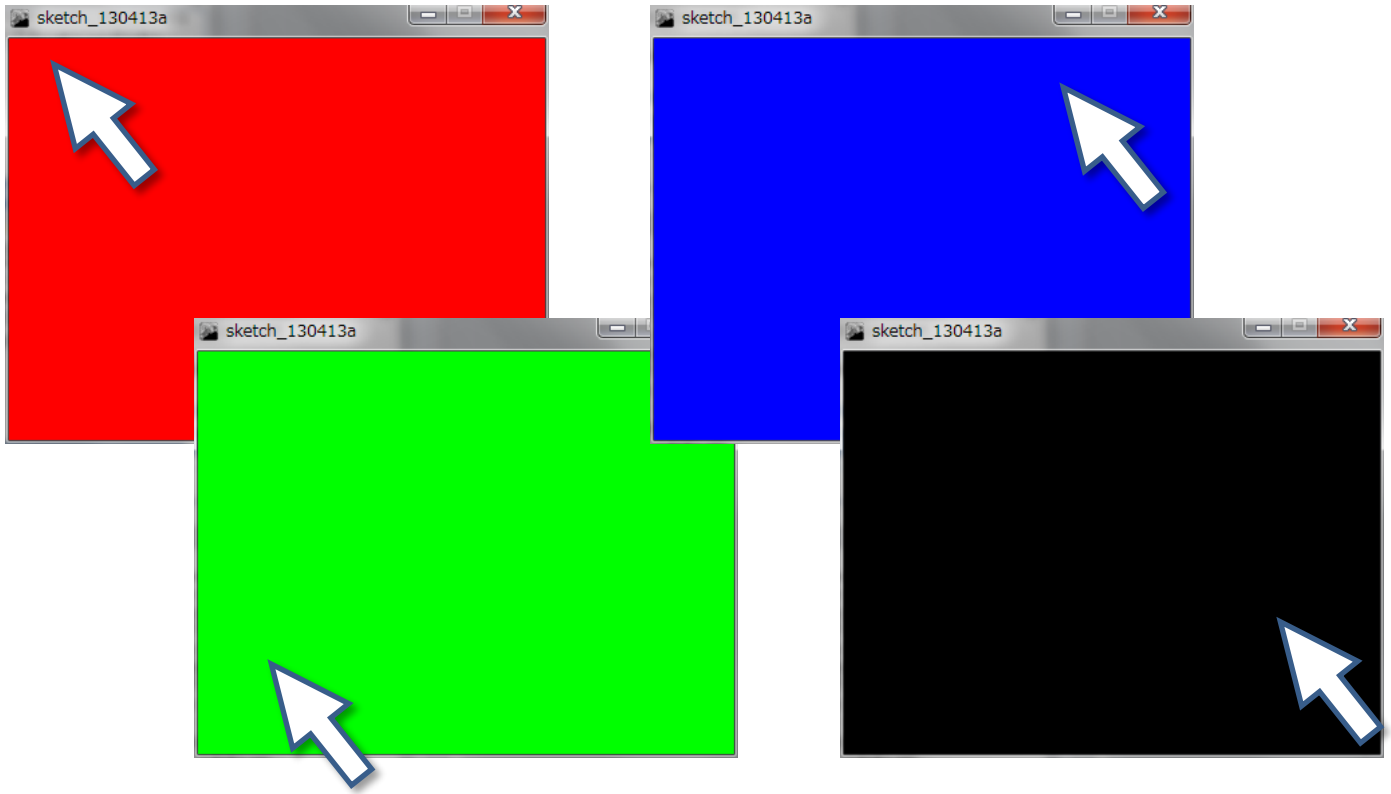


# 色々な条件に挑戦

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 400x300のウインドウでマウスが画面の左上で赤背景，右上で青背景，左下で緑背景，右下で黒背景にするには？



# 色々な条件に挑戦

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 条件(「<」にするか「<=」にするかは人次第)
  - 左上は mouseX < 200 かつ mouseY < 150 ➔ 赤色
  - 右上は mouseX >= 200 かつ mouseY < 150 ➔ 青色
  - 左下は mouseX < 200 かつ mouseY >= 150 ➔ 緑色
  - 右下は mouseX >= 200 かつ mouseY >= 150 ➔ 黒色

```
void draw(){
   if( mouseX < 200 && mouseY < 150){
       background(255,0,0);
   } else if( mouseX >= 200 && mouseY < 150 ){
       background(0,0,255);
   } else if( mouseX < 200 && mouseY >= 150 ){
       background(0,255,0);
   } else {
       background(0,0,0);
   }
}
```

# 補足

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- if の処理内容が1つだけのときは{}を省略可能

```
if( mouseX < 200 && mouseY < 150){
    background(255,0,0);
}
```

```
if( mouseX < 200 && mouseY < 150)
    background(255,0,0);
```

**ただ，慣れるまでは省略しない！**

# 値で沢山分岐する場合は？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- キーボードからの入力は「void keyPressed()」で取得，入力キーは変数の「key」を調べるだけ！
  - 沢山 if – else if – else if – else if – else if - … と繋げても良いが，見通しがやや悪くなる

```
void keyPressed(){
    if( key == 'a' ){
        println( "Aが押されました" );
    } else if( key == 'b' ){
        println( "Bが押されました" );
    } else if( key == 'c' ){
        println( "Cが押されました" );
    } else {
        println( "それ以外のキー" );
    }
}
```

```
void keyPressed(){
    switch( key ){
    case 'a':
        println( "Aが押されました" );
        break;
    case 'b':
        println( "Bが押されました" );
        break;
    default:
        println( "それ以外のキー" );
        break;
    }
}
```

# switch – case – break

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
switch( 分岐させる変数 ){
case 値1:
    値1の時の処理
    break; // 値1の時の処理ここまで
case 値2:
    値2の時の処理
    break; // 値2の時の処理ここまで
default:
    // 値1でも2でもない場合の処理
    break;
}
```

# up/down/left/right

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

key ではなく
keyCode という
変数を利用

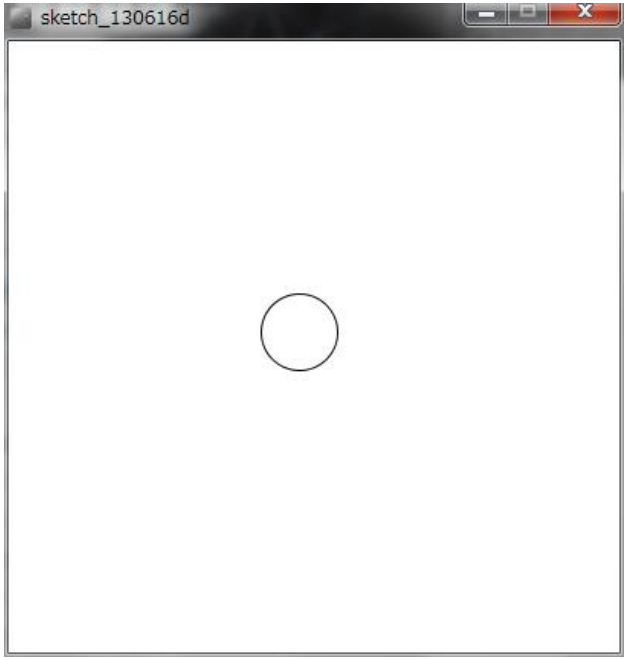
```
void keyPressed() {
   switch( keyCode ) {
   case UP:
      println( "upが押されました" );
      break;
   case DOWN:
      println( "downが押されました" );
      break;
   case LEFT:
      println( "leftが押されました" );
      break;
   case RIGHT:
      println( "rightが押されました" );
      break;
   default:
      println( "それ以外のキー" );
      break;
   }
}
```

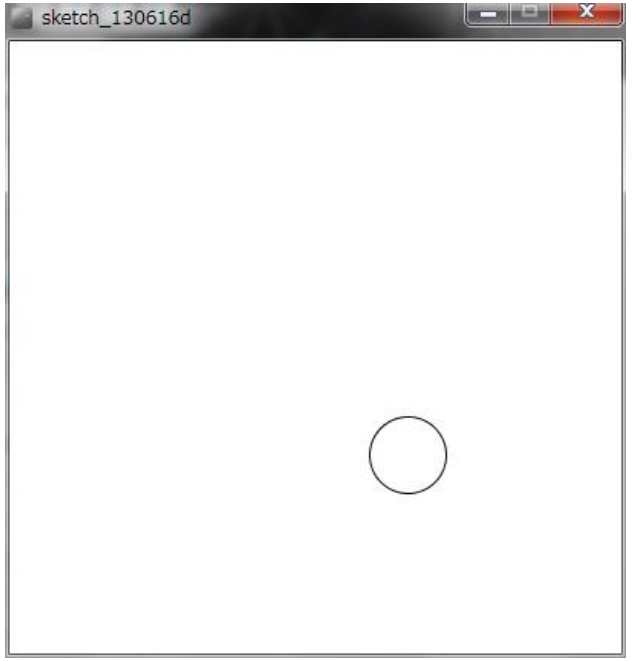
# up/down/left/rightで移動

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 上下左右キーで円を動かすにはどうするか？

- keyPressed() で入力を取得
- keyCode で上下左右ボタンを取得
- 座標を up/down/right/left で変更



# up/down/left/rightで移動

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


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

int x = 200;
int y = 200;
void draw(){
    background( 255, 255, 255 );
    ellipse( x, y, 50, 50 );
}
```

```
void keyPressed() {
   switch( keyCode ) {
   case UP:
      y = y - 10;
      break;
   case DOWN:
      y = y + 10;
      break;
   case LEFT:
      x = x - 10;
      break;
   case RIGHT:
      x = x + 10;
      break;
   }
}
```

# 占いを作ってみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


(Q) マウスクリックするたびに標準出力に占い結果を表示するプログラムを作ってみよう！

# 占いを作ってみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - マウスクリックした際に 0〜9 の乱数を発生
  - 乱数を発生させるのは random( ... );
    - random( 数字 ); 数字までのランダムな値を取得
    - random( 開始, 終了 ); 開始〜終了のランダムな値を取得
    - 注意点：乱数は実数（0.0000, 0.00001, 0.00002, ...）になるので整数と単純に比較してはダメ．整数に変換してあげる必要がある！（実数から変数へ (int) で変換！）

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

  - 乱数の値を変数に格納
  - 変数の値に応じて占い結果を表示する！

# 占いを作ってみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
void setup(){
  size( 100, 100 );
}

void draw(){
}

void mousePressed(){
  num = (int)random(0,10);
  switch( num ) {
  case 0:
    println( "DAIKICHi!! (^o^)" );
    break;
  case 1:
    println( "DAIKYO!! (X_X)" );
    break;
  case 2:
    println( "KICHI  (-_-)" );
    break;
    // ～以降 3 - 9 まで繰り返し
  }
}
```

# 予習問題

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 左から右に動く白色の四角形をマウスでクリックすると赤色で塗りつぶすプログラムに挑戦
- 画面に表示されていない左から右に動く四角形を探すプログラムに挑戦
  - 惜しい場合はヒントを表示しましょう！
- 簡単なゲームを作ってみましょう
  - ただ表示されているボタンを全て押すだけのゲーム
  - 動いてくる敵を避けるゲームなど．なんでもOKです
- 画面のクリック場所によって占いの結果を変更するプログラムを作ってみましょう