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（2）
## アニメーションしてみよう

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


- アニメーションに挑戦してみる
  - 勝手に動くものを作る
  - マウスに連動して動くものを作る
- 標準出力を理解する

# 基本的な事

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


命令名(命令の詳細, 命令の詳細, ...);

- 例: size, background, line, ellipse, ...
- すべて半角英数字
  - 日本語はダメ！大文字小文字に注意！
- 命令の詳細は括弧の中に！
  - 複数あるときはカンマで区切る
- 最後はセミコロン！
- プログラムは上から順に実行される

# 準備と描画

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 には，大きく分けて

  setup （準備）と draw（描画）がある

  - 「準備」では，プログラムが実行されるときに最初に1回だけ何を行うかということを記述する
    - ウインドウサイズの指定
    - 利用する画像や音声の読み込み
  - 「描画」では，プログラムが実行されている際に，毎回繰り返し何を描画するかということを記述する
    - 画面上での何らかのアニメーション
    - 画像の表示や音声の再生

# 準備の記述方法

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

```
void setup()
{
  size(400, 300);
  background(255, 255, 255);
}
```

- 「**void setup(){**」から「**}**」までの間に準備の内容を記述する
- 上記の例では，400x300のウインドウを作り，背景の色を白色（255,255,255）に指定している

# 描画の記述方法

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

```
void draw()
{
  fill(255, 0, 0);
  ellipse(200, 150, 150, 150);
}
```

- 「**void draw(){**」から「**}**」までの間に毎回描画する内容を記述する
- 上の例では，塗り色を赤色(255,0,0)に指定し，(200,150)の位置に直径150の円を描画
  - 円の位置が動かないのであまり意味が無いが…

# voidとか()とか{}とか

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

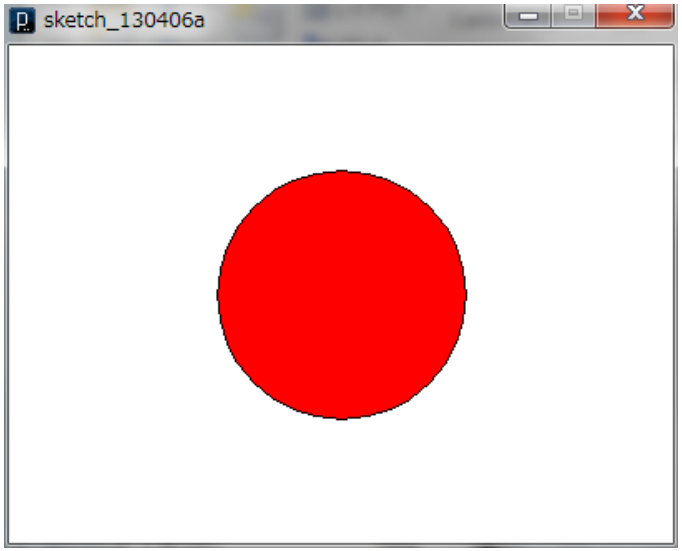

- setup や draw の前後の void と () は，まずはおまじないだと思って下さい（後に説明します）
- 重要なのは，「{」から「}」までが1まとまりで，括弧内が上から下にまとめて実行されるという事
- 開いたら閉じる！！
  - 「(」には必ず対応する「)」が必要！
  - 「{」には必ず対応する「}」が必要！
  - 後に登場しますが「[」にも「]」が必要！

# 日本の国旗の例を

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

- 試しに入力してみましょう

```
void setup()
{
  size(400, 300);
  background(255, 255, 255);
}

void draw()
{
  fill(255, 0, 0);
  ellipse(200, 150, 150, 150);
}
```



# 日本の国旗の例を

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

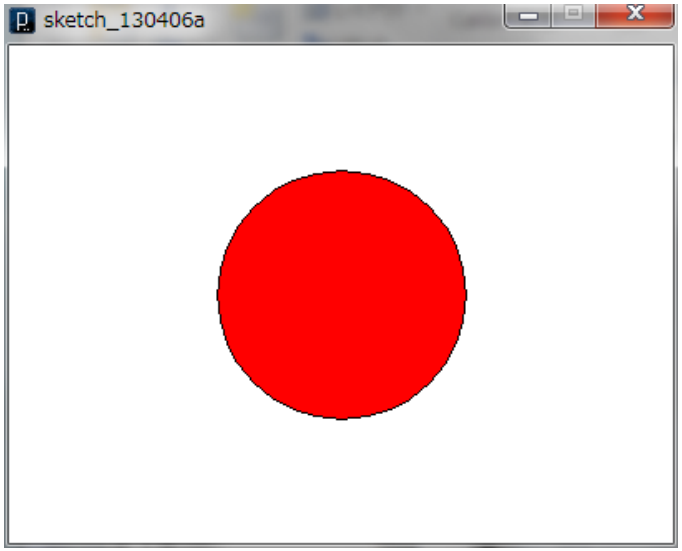
- 試しに入力してみましょう

```
void setup()
{
  size(400, 300);
  background(255, 255, 255);
}
```

最初に一度だけ

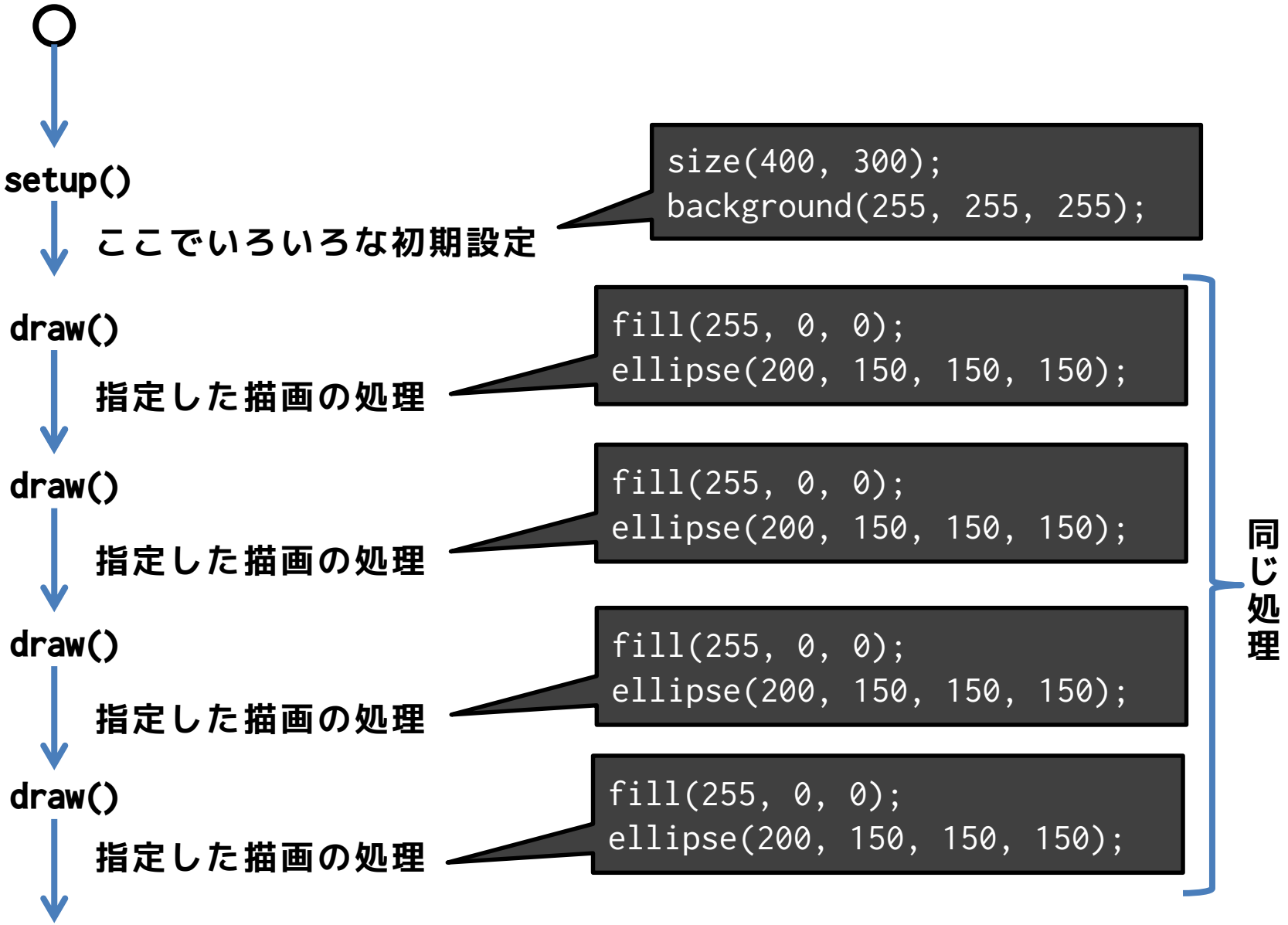
```
void draw()
{
  fill(255, 0, 0);
  ellipse(200, 150, 150, 150);
}
```

描画の度に



# 流れ

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



# 今回使うもの

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

- draw が何回呼び出されたか
  - frameCount
- マウスのX座標
  - mouseX
- マウスのY座標
  - mouseY

# 赤丸を動かす

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

- 試しに入力してみましょう

```
void setup()
{
  size(400, 300);
  background(255, 255, 255);
}
```

最初に一度だけ

```
void draw()
{
  fill(255, 0, 0);
  ellipse(frameCount, 150, 150, 150);
}
```

描画の度に

# 赤丸を動かす

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

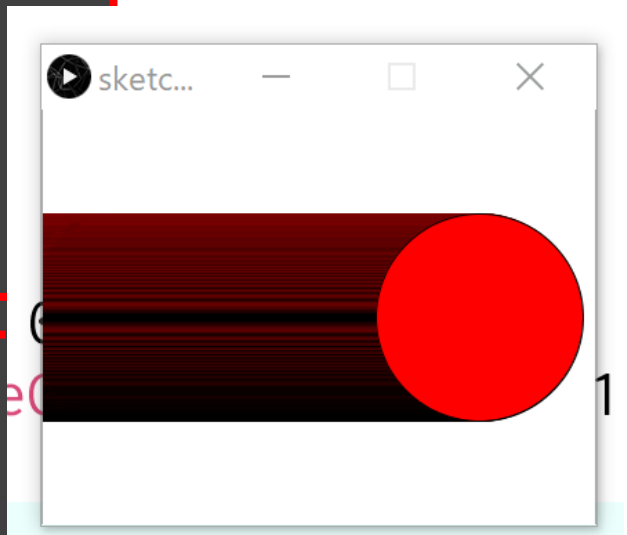
- 試しに入力してみましょう

```
void setup()
{
  size(400, 300);
  background(255, 255, 255);
}
```

最初に一度だけ

```
void draw()
{
  fill(255, 0, 0);
  ellipse(frameCount, 150, 150, 150);
}
```

描画の度に




前のが残って汚い感じ

# 流れ（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

setup()

ここでいろいろな初期設定

```
size(400, 300);
background(255, 255, 255);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

# 流れ（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

setup()

ここでいろいろな初期設定

```
size(400, 300);
background(255, 255, 255);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw

背景を白色に塗りなおしていないため
どんどん重ね塗りされてしまっている...

# 流れ（解決）

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

**背景を毎回白色で塗り直す！**

setup()

ここでいろいろな初期設定

```
size(400, 300);
```

draw()

指定した描画の処理

```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

draw()

指定した描画の処理

```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(frameCount, 150, 150, 150);
```

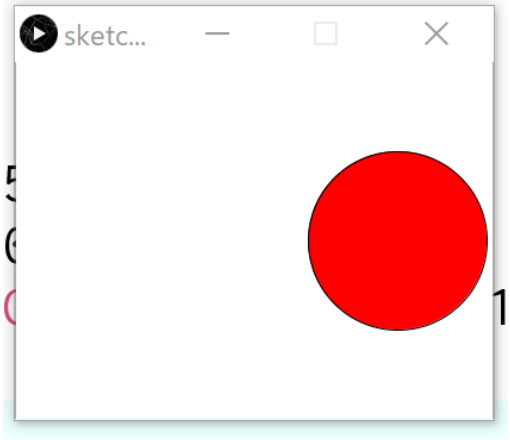
# 赤丸を動かす

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

- 試しに入力してみましょう

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

void draw()
{
  background(255, 255, 255);
  fill(255, 0, 0);
  ellipse(frameCount, 150, 150, 150);
}
```

描画の度に背景を白色に

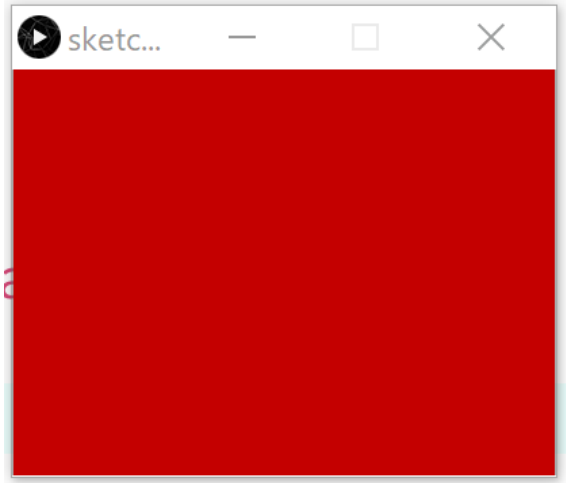


# 背景を黒色→赤色にする

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

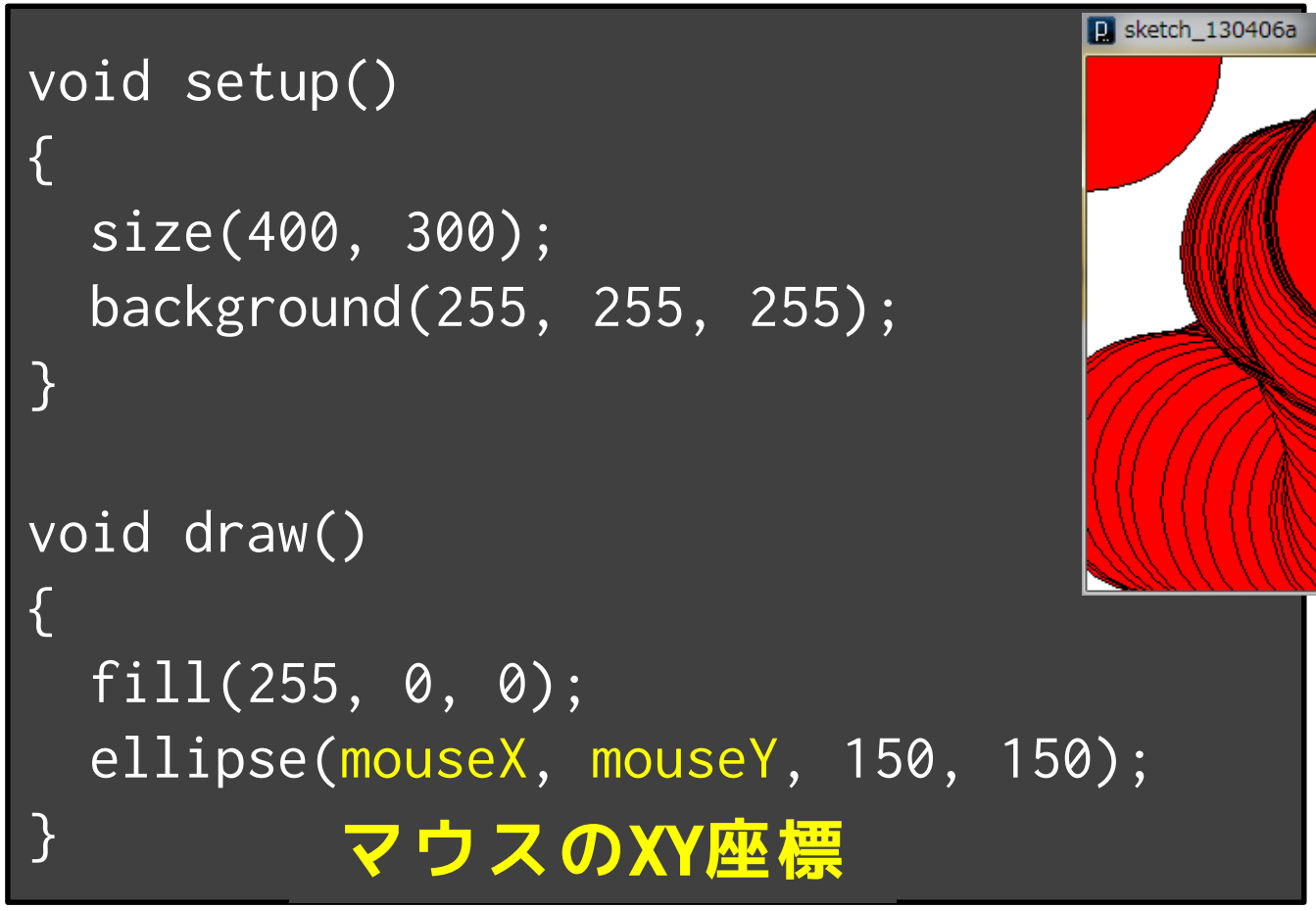
- 試しに入力してみましょう
  - 黒色(0，0，0) から赤色(255，0，0)へと変化させる

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

void draw()
{
  background(frameCount, 0, 0);
}
```



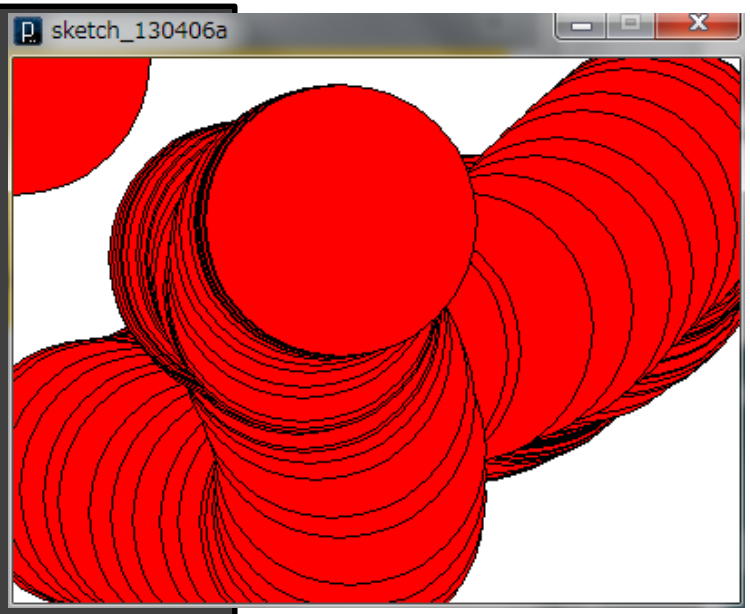
# 赤丸をマウスの場所に

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, 300);
  background(255, 255, 255);
}

void draw()
{
  fill(255, 0, 0);
  ellipse(mouseX, mouseY, 150, 150);
}
```

マウスのXY座標




ぐちゃぐちゃになってしまう

# 流れ（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

setup()

ここでいろいろな初期設定

```
size(400, 300);
background(255, 255, 255);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(mouseX, mouseY, 150, 150);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(mouseX, mouseY, 150, 150);
```

draw()

指定した描画の処理

```
fill(255, 0, 0);
ellipse(mouseX, mouseY, 150, 150);
```

draw()

指定

背景を白色に塗りなおしていないため
どんどん重ね塗りされてしまっている...

# 流れ（解決）

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

**背景を白色で塗り直す！**

setup()

```
size(400, 300);
```

ここでいろいろな初期設定

draw()

指定した描画の処理

```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(mouseX, mouseY, 150, 150);
```

draw()

指定した描画の処理

```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(mouseX, mouseY, 150, 150);
```

draw()

指定した描画の処理

```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(mouseX, mouseY, 150, 150);
```

draw()

指定した描画の処理

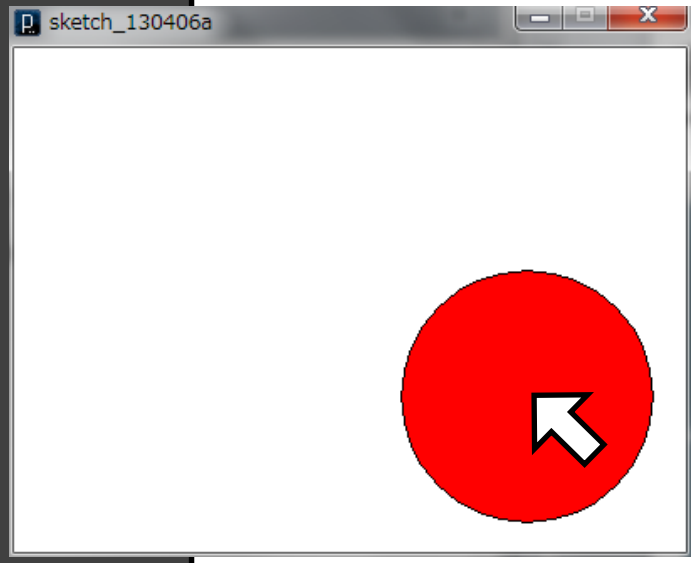
```
background(255, 255, 255);
fill(255, 0, 0);
ellipse(mouseX, mouseY, 150, 150);
```

# 赤丸をマウスの場所に

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, 300);
}

void draw()
{
  background(255, 255, 255);
  fill(255, 0, 0);
  ellipse(mouseX, mouseY, 150, 150);
}
```

マウスのXY座標

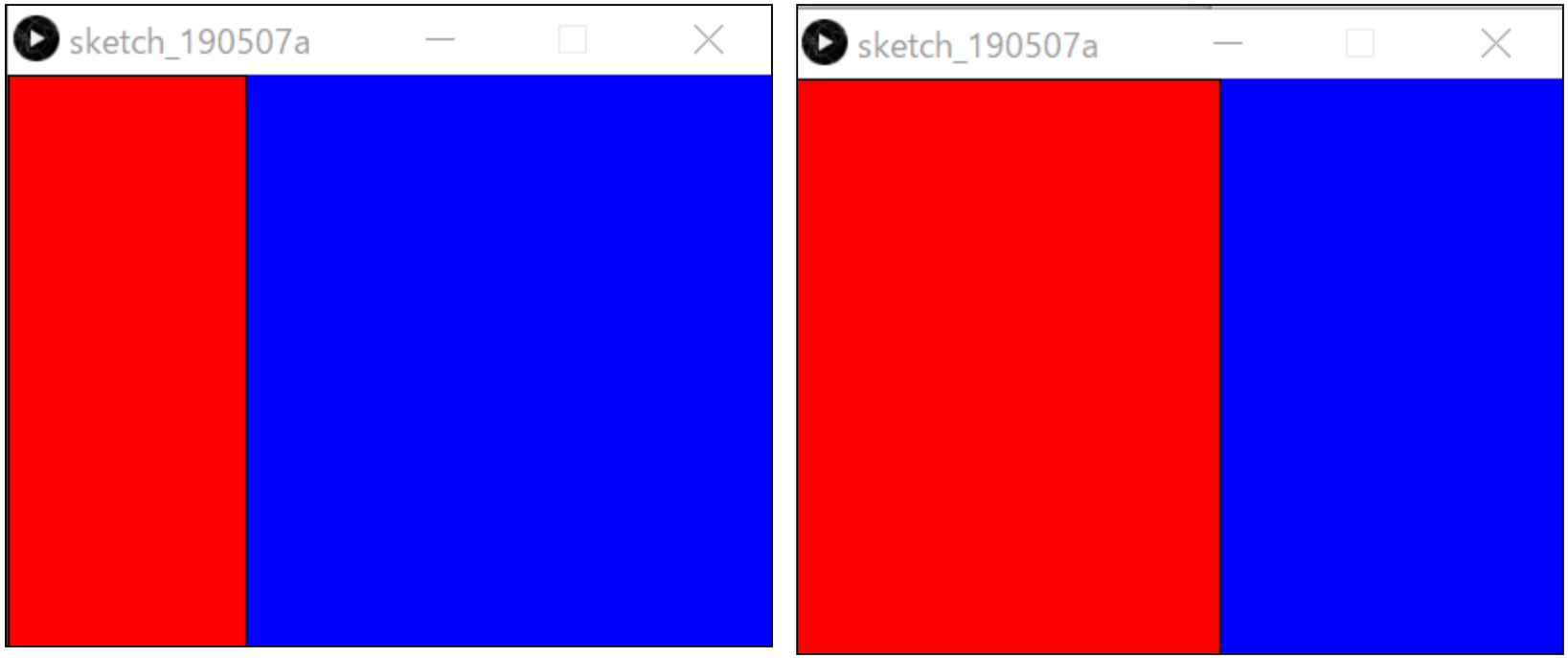



- mouseX と mouseY はカーソルの位置
- draw の中で mouseX や mouseY を利用するとその点に絡めた描画が可能

# プログレスバーを実現

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

- 最初青一色の画面の左端から赤色の領域が増えてきて，最終的には全面が赤色になるプログラムを作成せよ




– 背景を青色で塗りつぶして，frameCountの横幅になる赤色の四角形を描けばよい！

小テスト

# プログレスバーを実現

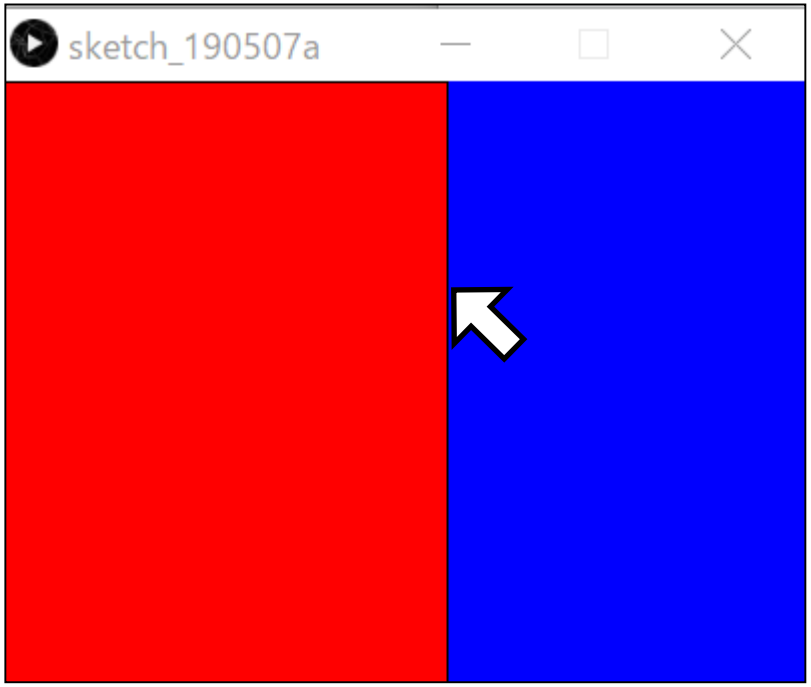
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

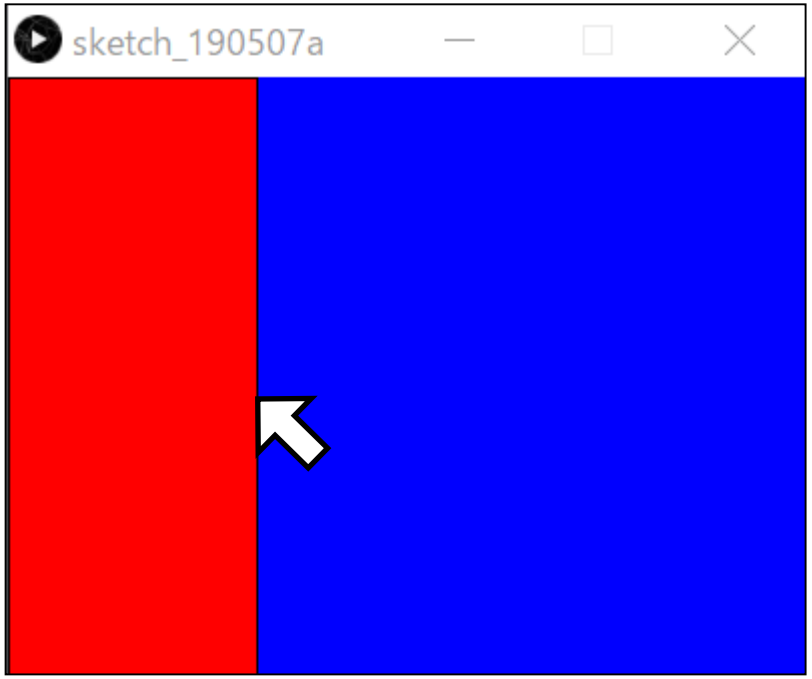
```
void setup()
{
  size(400, 300);
}

void draw()
{
  background(0, 0, 255);
  fill(255, 0, 0);
  rect(0, 0, frameCount, 300);
}
```

# マウス位置で2つの領域を

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

- マウスの位置に応じて赤色と青色の領域が変わるようにするにはどうしたら良いか？




– 背景を青色で塗りつぶして，マウスの位置まで赤色の四角形を描けばよい！

小テスト

# マウス位置で2つの領域を

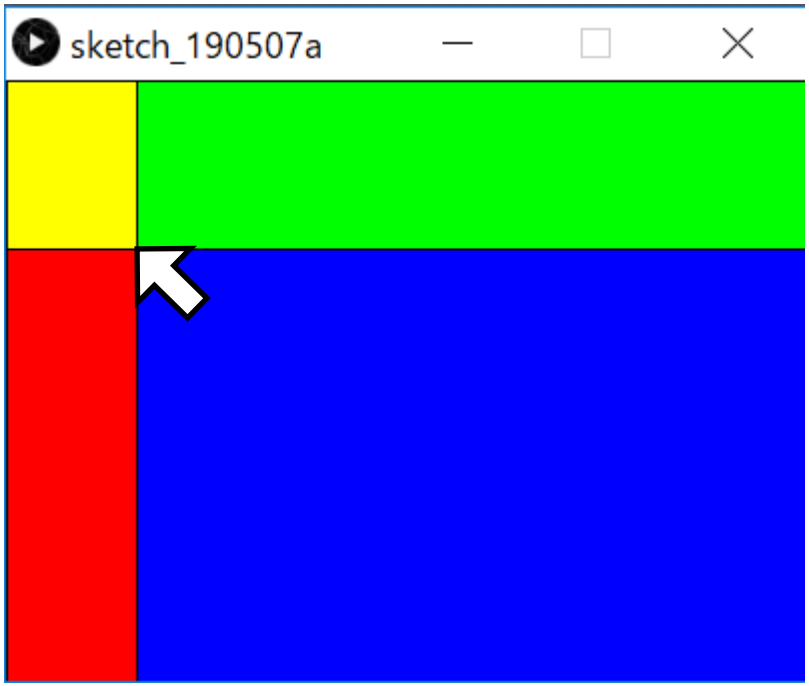
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


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

void draw()
{
  background(0, 0, 255);
  fill(255, 0, 0);
  rect(0, 0, mouseX, 300);
}
```

# マウス位置で4つの領域を

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


- マウスの位置に応じて赤色と青色と黄色と緑色の領域が変わるようにするには？




  - 背景を青色で塗りつぶして，赤色と緑色と黄色の四角形をそれぞれ描けばよい！

# マウス位置で4つの領域を

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, 300);
}

void draw()
{
  background(0, 0, 255);
  fill(255, 0, 0);
  rect(0, 0, mouseX, 300);
  fill(0, 255, 0);
  rect(0, 0, 400, mouseY);
  fill(255, 255, 0);
  rect(0, 0, mouseX, mouseY);
}
```

# アニメーション速度調整

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

- アニメーションは，1秒当たり何回 draw() で描画するかによってその速度や滑らかさが変わる
  - 1秒間に60フレーム（60 fps）なのか
  - 1秒間に10フレーム（10 fps）なのか
  - 1秒間に1フレーム（1 fps）なのか

```
frameRate(1秒間のフレーム数);
```

```
void setup(){
  size(400, 400);
  frameRate(60);
}
```

```
void setup(){
  size(400, 400);
  frameRate(10);
}
```

```
void setup(){
  size(400, 400);
  frameRate(1);
}
```

# プログレスバーをゆっくり

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

- 青一色の画面の左端から赤色の領域が増え，40秒後に全面が赤色になるようにせよ
  - 横幅400ピクセルで，40秒後なので 400 / 40 = 10 の速度で徐々に描画していけばよい！

```
void setup()
{
  size(400, 300);
  frameRate(10);
}

void draw()
{
  background(0, 0, 255);
  fill(255, 0, 0);
  rect(0, 0, frameCount, 300);
}
```

# 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

- 400x300のウインドウの画面上端から下端に向けて動いていく直径150ピクセルの赤い丸を描け（なお，円の動く速度は一定とし，視認できるレベルとせよ）
  - 赤丸を動かすプログラムを改良するだけ！

# 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

- 255x255のウインドウを用意し，マウスのX座標に応じて背景の色を黒色から赤色に変更するようにせよ
  - 左端の場合に黒色(0，0，0)で，右端の場合に赤色(255，0，0)になるようにする

# 違いを見るために

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

println("表示したい文字列");

（ぷりんとえるえぬです．ぷりんとあいえぬではないです）

を入れてどのように動作しているか調べてみよう

```
void setup()
{
  size(400, 300);
  println("background");
  background(255, 255, 255);
}

void draw()
{
  println("fill");
  fill(255, 0, 0);
  println("ellipse");
  ellipse(mouseX, mouseY, 150, 150);
}
```

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

void draw()
{
  println("background");
  background(255, 255, 255);
  println("fill");
  fill(255, 0, 0);
  println("ellipse");
  ellipse(mouseX, mouseY, 150, 150);
}
```

# エディタ下の表示に注目

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

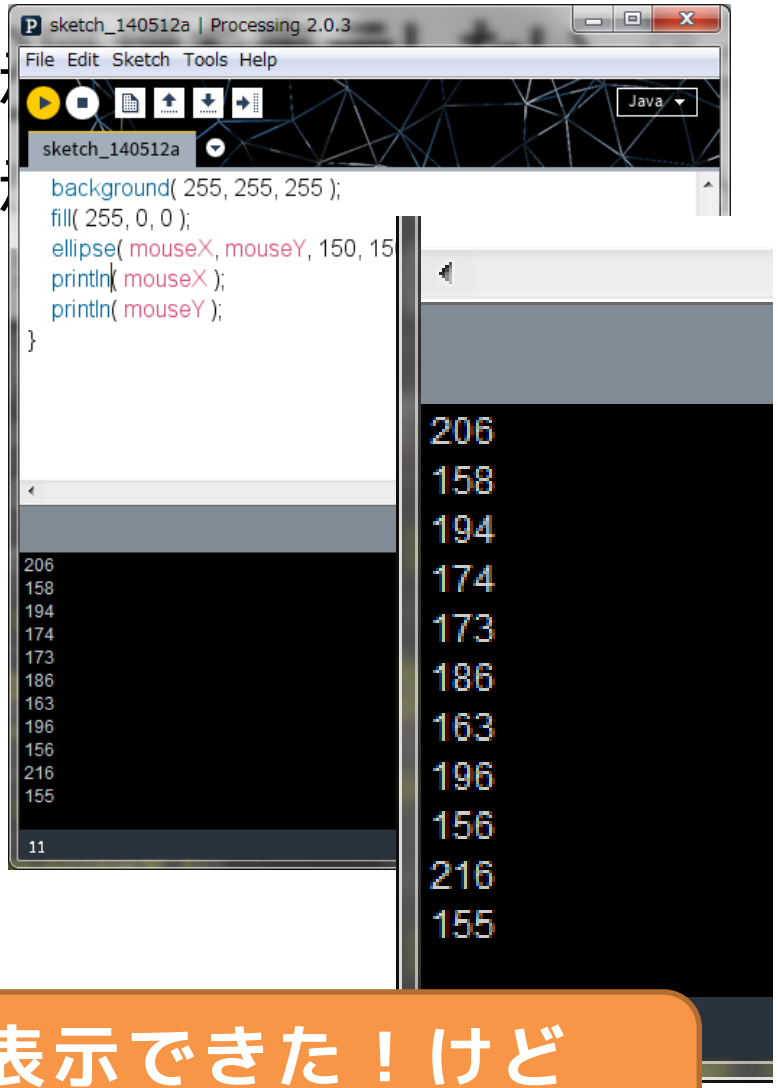


# マウスの座標を知りたい

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


- println で mouseX の値を表
- println で mouseY の値を表

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

void draw()
{
  background(255, 255, 255);
  fill(255, 0, 0);
  ellipse(mouseX, mouseY, 150, 150);
  println(mouseX);
  println(mouseY);
}
```




値を表示できた！けど
ちょっとわかりにくい...

# println と print

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

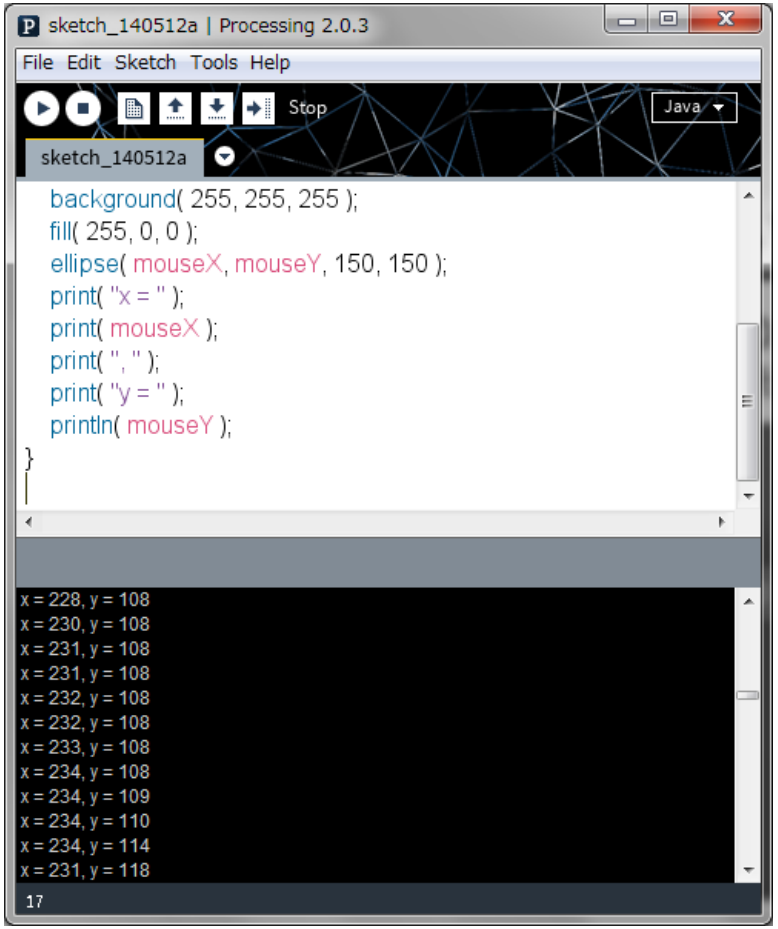

- printlnのかわりにprintを使うと改行されません

```
void setup()
{
  size(400, 300);
  println("background");
  background(255, 255, 255);
}

void draw()
{
  print("fill");
  fill(255, 0, 0);
  println("ellipse");
  ellipse(mouseX, mouseY, 150, 150);
}
```

# マウスの座標を表示したい

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


- 例えば (200, 300) にマウスの座標がある場合には， x = 200, y = 300 と表示したい！
  - print で 「x = 」 を書いて
  - print で mouseX を表示して
  - print で 「, 」 を書いて
  - print で 「y = 」 を書いて
  - println で mouseY を表示



# マウスの座標を表示したい

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

- 例えば (200, 300) にマウスの座標がある場合には， x = 200, y = 300 と表示したい！

```
void draw()
{
  background(255, 255, 255);
  fill(255, 0, 0);
  ellipse(mouseX, mouseY, 150, 150);
  print("x = ");
  print(mouseX);
  print(", ");
  print("y = ");
  println(mouseY);
}
```

# マウスの座標を表示したい

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

- ちなみに，下記のように書くことも可能
  - 「+」は文字と文字をくっつけるという意味

```
void draw()
{
  background(255, 255, 255);
  fill(255, 0, 0);
  ellipse(mouseX, mouseY, 150, 150);
  println("x=" + mouseX + ", y=" + mouseY);
}
```

小テスト

# 時分秒を表示してみる！

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

- 現在の時分秒をどうやって取得するか？
  - 必要なものは命令として用意されている
  - https://processing.org/reference/hour_.html
  - 積極的にリファレンスを探してみよう！！

```
void setup()
{
  print(hour() + "時");
  print(minute() + "分");
  print(second() + "秒");
}
```

# 重要なポイント

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

- プログラムはブロックの組み合わせからなる
- ブロックは{}で囲まれた領域がそれに該当
- ブロック内は上から順に連続して実行される
- ブロックの中身が同じ階層にあることをわかりやすくするために，「{」「}」の位置に注意することと，字下げ（インデント）をする

# [超重要] 括弧の位置

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

- 初学者は→を推奨

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

void draw(){
  background(255, 255, 255);
  if(mouseX < 200){
    fill(255, 0, 0);
  } else {
    fill(0, 0, 0);
  }
  ellipse(200, 200, 100, 100);
}
```

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

void draw()
{
  background(255, 255, 255);
  if(mouseX < 200)
  {
    fill(255, 0, 0);
  }
  else
  {
    fill(0, 0, 0);
  }
  ellipse(200, 200, 100, 100);
}
```

# [超重要] インデント

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

- プログラムのブロックを把握するために利用

```
void setup(){
size(400,400);
}
void draw(){
background(255,255,255);
if(mouseX<200){
fill(255,0,0);}
else {
fill(0,0,0);}
ellipse(200,200,100,100);
}
```

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

void draw()
{
  background(255, 255, 255);
  if(mouseX < 200)
  {
    fill(255, 0, 0);
  }
  else
  {
    fill(0, 0, 0);
  }
  ellipse(200, 200, 100, 100);
}
```

# [超重要] インデント

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

- インデント
  - 「{」と「}」との間を右に1段字下げして左を揃える
  - ゲシュタルトの心理学の良い連続！

```
void setup()
{
Tab size(400, 300);
Tab background(255, 255, 255);
}

void draw()
{
Tab fill(255, 0, 0);
Tab ellipse(200, 150, 150, 150);
}
```

Tab はキーボードの「Tabキー」の事

# BSD記法（オールマン記法）

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


- 右記の記述形式に慣れる！
- 「{」と「}」はそれぞれ1行に！
- 「{」と「}」の横の位置は揃える
- 括弧直前の命令の左端と揃える
- 「{」と「}」の中は2つ分スペースで字下げする（インデント）
- 同じ階層（ブロック）のものは，横の位置を揃える！！！！
- **左端をそろえるのだ！！！！**

```
hogehoge
{
  hogehogehoge;
  higehigehige;
  piyopiyopiyo;
}

hogehige
{
  hogehogehoge;
  hogehoge
  {
    higehige;
    higehige;
  }
  piyopiyopiyo;
}
```

# BSD記法（オールマン記法）

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

- 右記の記述形式に慣れる！
- 「{」と「}」はそれぞれ1行に！
- 「{」と「}」の横の位置は揃える
- 括弧直前の命令の左端と揃える
- 「{」と「}」の中は2つ分スペースで字下げする（インデント）
- 同じ階層（ブロック）のものは，横の位置を揃える！！！！
- **左端をそろえるのだ！！！！**

**Edit → Auto Format ですべて整形**
**Ctrl+T を押すくせをつけよう！**

```
hogehoge
{
  hogehogehoge;
  higehigehige;
  piyopiyopiyo;
}

hogehige
{
  hogehogehoge;
  hogehoge
  {
    higehige;
    higehige;
  }
  piyopiyopiyo;
}
```

# コメント

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

- プログラムとしては実行されない，人間用の説明文で，後で読むためにどんどん書く！
- コメントは「//」か「/*」と「*/」のペアを利用
  - 「//」は「//」以降行末までをコメントとして解釈
  - 「/*」と「*/」は，その中身を全てコメントとして解釈

```
// 背景を白色に設定
background(255, 255, 255);
fill(255, 0, 0); // 赤色で塗りつぶす
// fill(0, 0, 255); // 青色で塗りつぶす
/*
  ここから
  ここも
  ここまでもコメントですよー
*/
```

**動かない時に消さずコメントアウトしよう！**

# 最後に

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


- 短いプログラムは良いのだけれど，長くなるとすぐにワケが分からなくなります

- 下記の2つの点に注意しましょう！
  1. コメントをしっかり書く
  2. インデントでわかりやすく！

# 参考

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入門
  - http://www.cp.cmc.osaka-u.ac.jp/~kikuchi/kougi/simulation2009/processing0.html
- Processing リファレンス
  - http://processing.org/reference/