プログラミング入門体験

コンピュータリテラシー

プログラム

- 計算機(コンピュータ Computer)
  - プログラムにしたがって動作する機械
- 計算機が理解できるプログラム
  - 機械語
- 機械語
  - 人間が理解するのは困難
- プログラム言語
  - 人間が理解できる言葉で記述したプログラムを，機械語に翻訳して利用

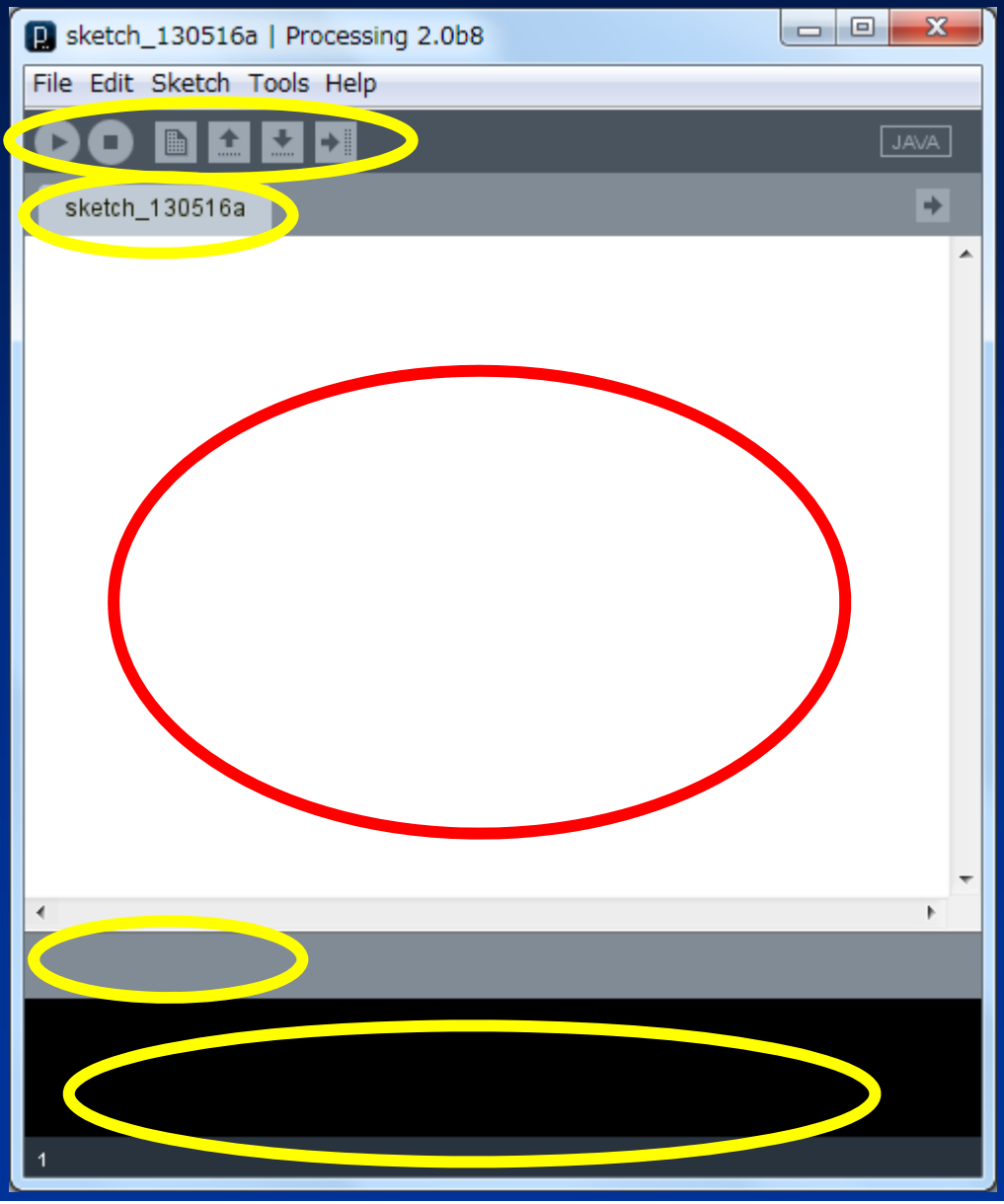
色々なプログラム言語

- C
- Visual BASIC (「情報の科学」で利用？)
- FORTRAN，BASIC，COBOL，PL/I
- Java，C#，C++
- Lisp，Prolog，perl，awk，sed
- Java script，Action Script，AIR
- Pascal，Ada，Algol，APL，Sql
- Processing，ドリトル，PEN，等々

Processing

- プログラミング入門用に開発された言語
- 少ないキー入力で容易にプログラミング可能
- インターネットから入手可能
- 文法がJavaに似ている
- Javaで作られている
  - Windows，Mac，Unixで利用可能
- 「プログラミング入門体験」に最適と判断

Processingの起動



ツールバー

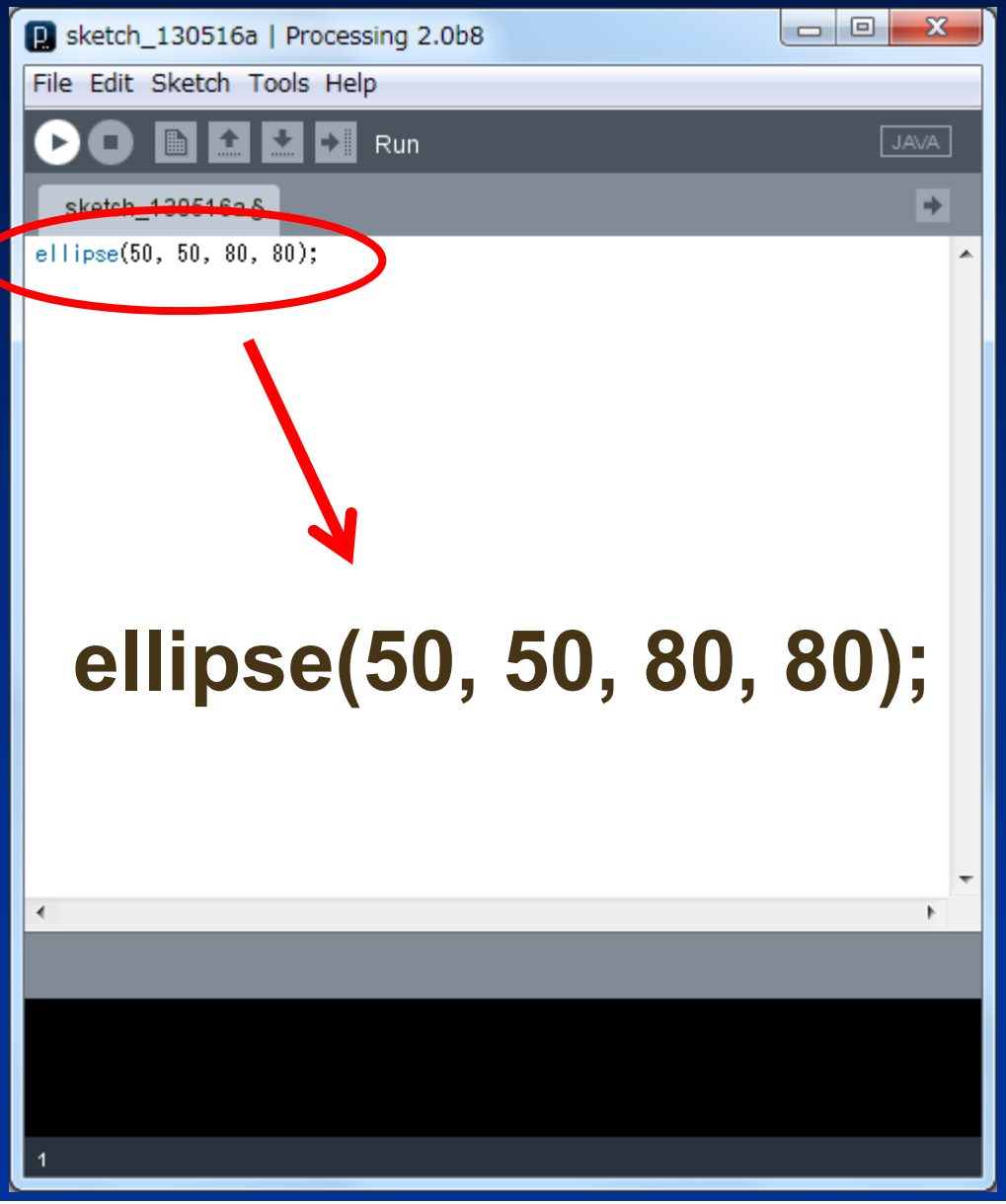
タブ

テキストエディタ

メッセージエリア

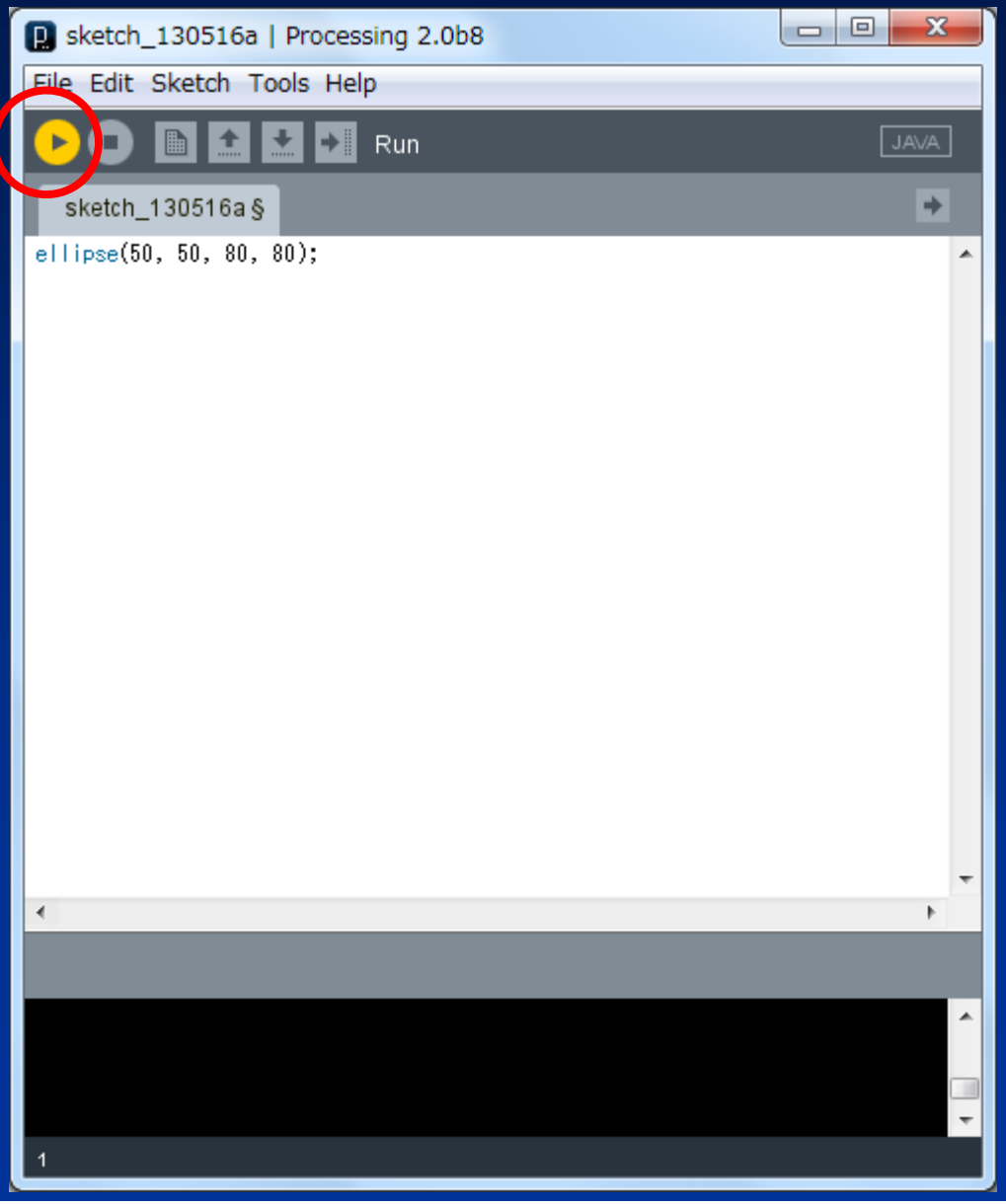
コンソール

プログラムの入力(p01)

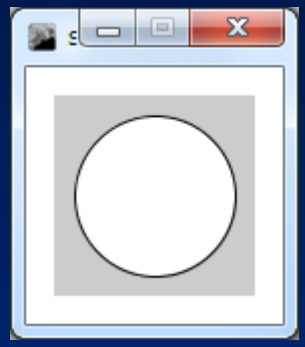


ellipse(50, 50, 80, 80);

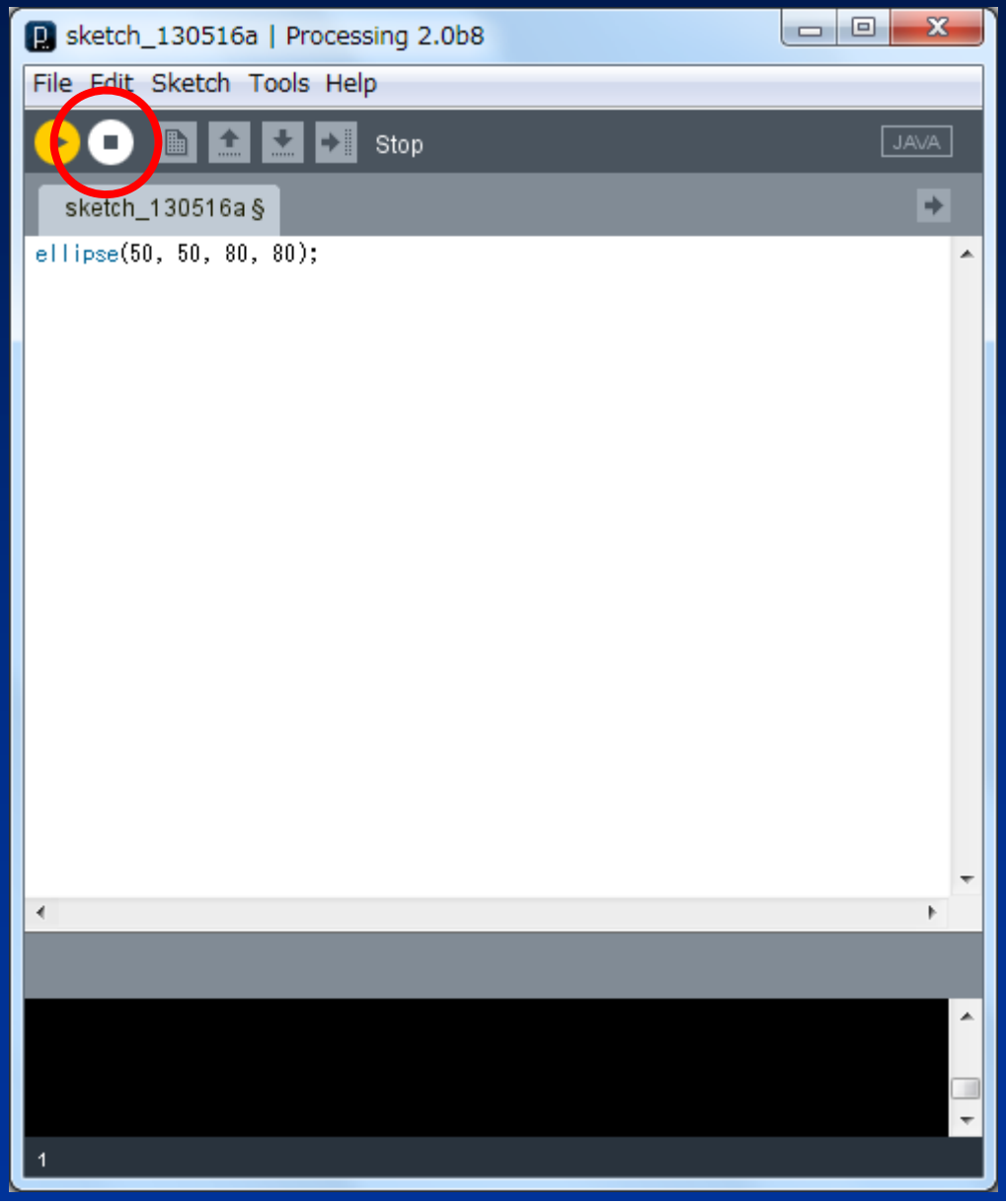
プログラムの実行


sketch_130516a | Processing 2.0b8
File Edit Sketch Tools Help
Run
JAVA
sketch_130516a §
ellipse(50, 50, 80, 80);
1

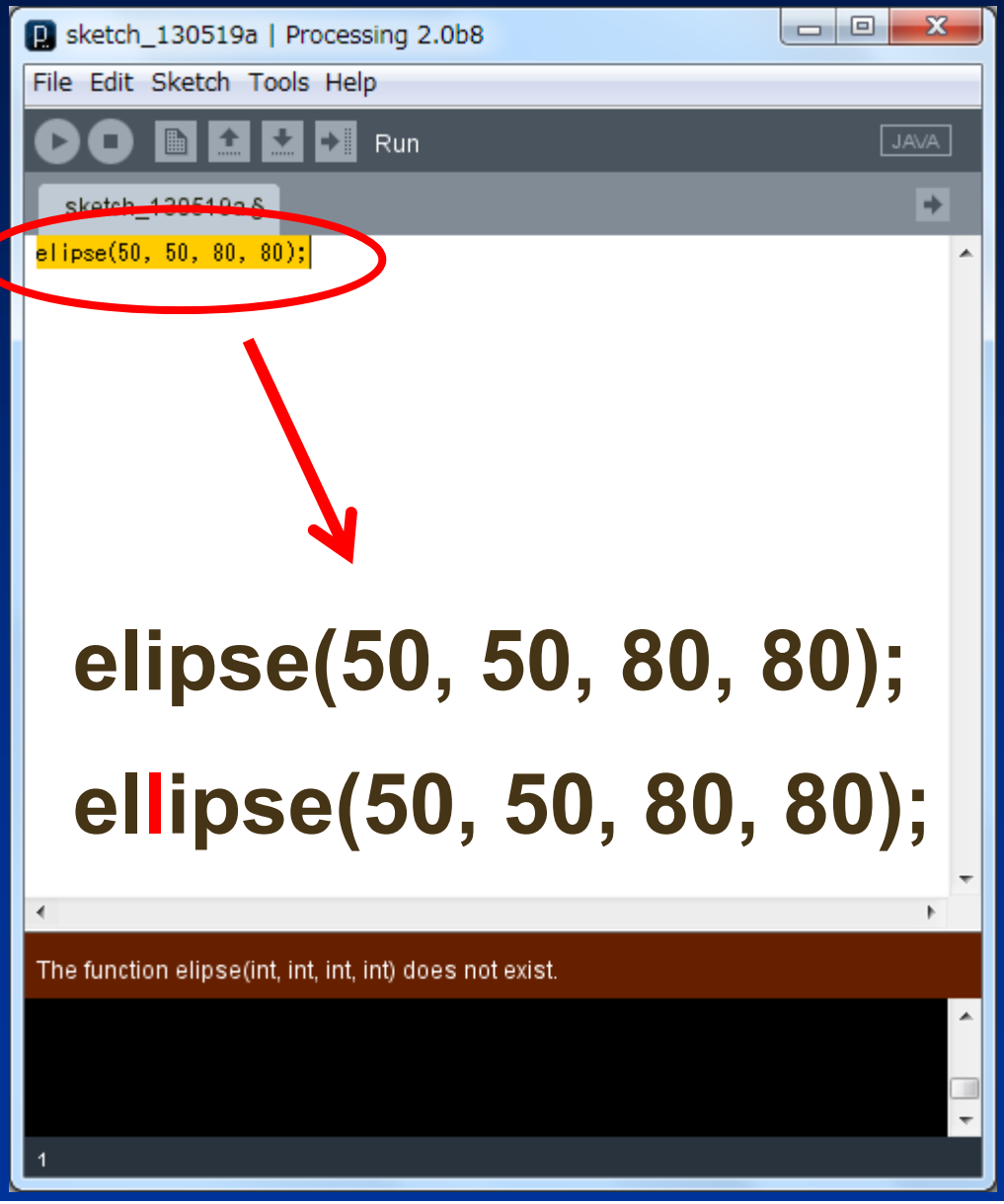
プログラムの停止



実行ウインドウ

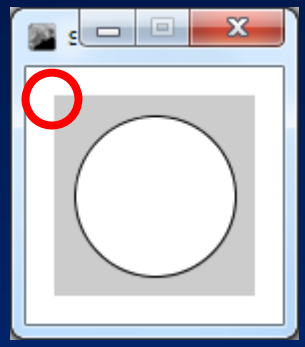


エラー（誤り）の発生



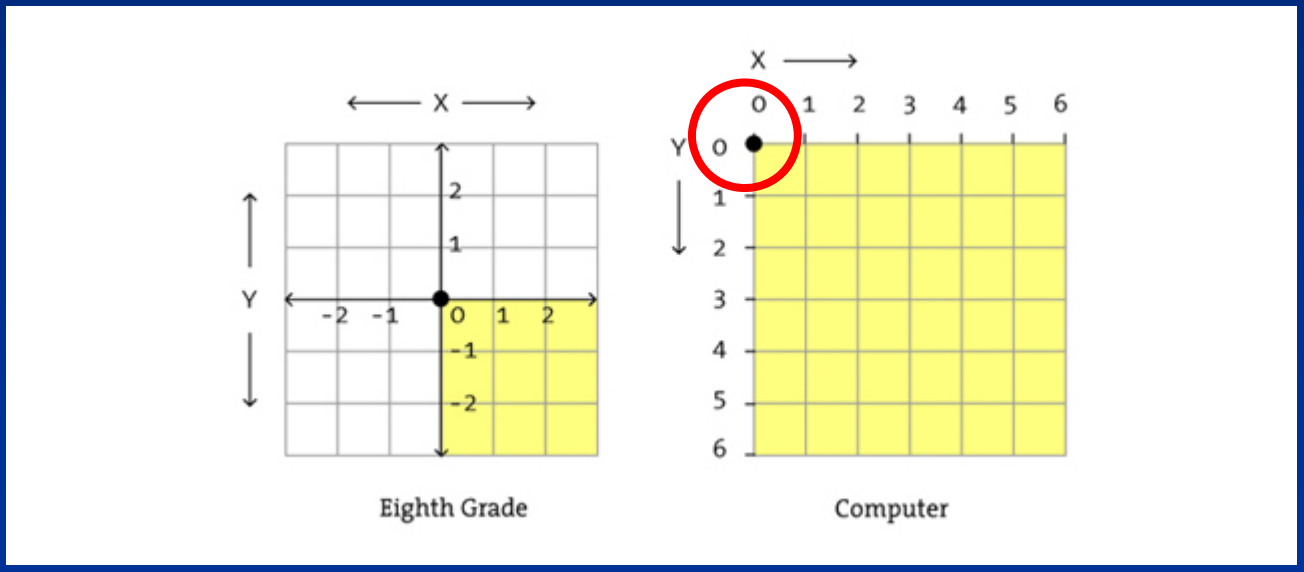
誤りを訂正します

座標系について



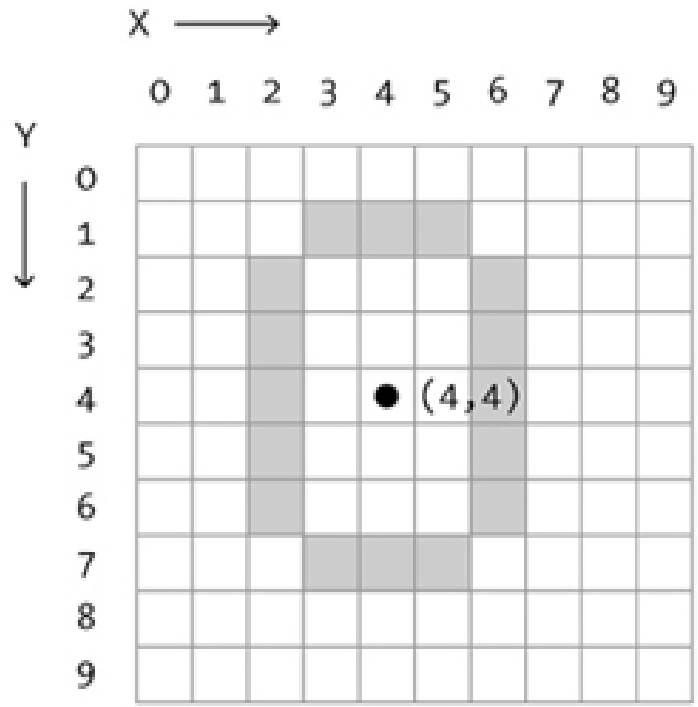
実行ウインドウ

実行ウインドウでは、
左上が原点になります



http://www.processing.org/learning/drawing/より掲載

ellipse（楕円）




```
ellipseMode(CENTER);
ellipse(x,y,width,height);
```

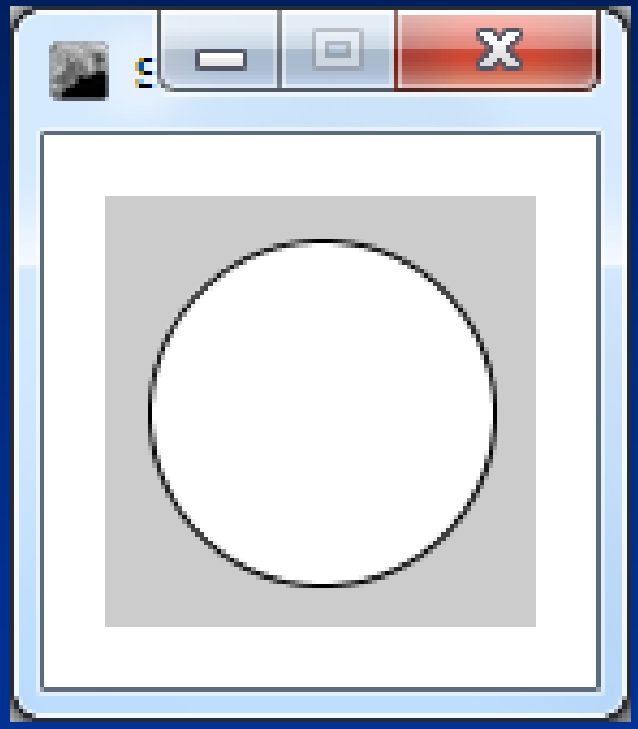
Example:

```
ellipseMode(CENTER);
ellipse(4,4,5,7);
```

http://www.processing.org/learning/drawing/より掲載

プログラムの意味

中心　50, 50



高さ　80

幅　80

ellipse(50, 50, 80, 80);

↑　↑　↑

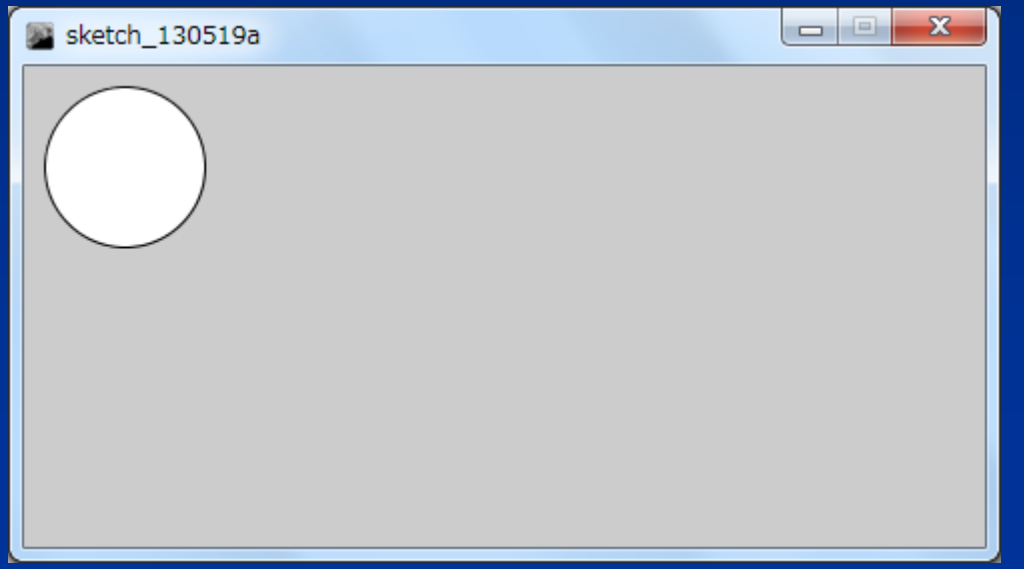
中心　幅　高さ

楕円を表示する関数
幅と高さが同じ場合は
真円になる.

実行ウインドウのサイズ変更(p02)

プログラム

```
size(480, 240);
ellipse(50, 50, 80, 80);
```

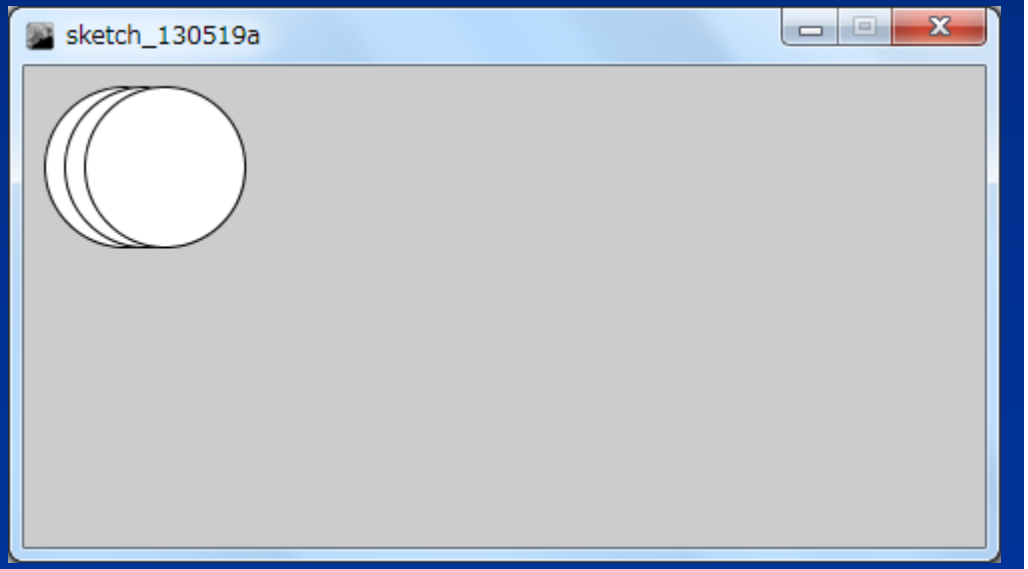


複数の円の表示(p03)

プログラム

```
size(480, 240);
ellipse(50, 50, 80, 80);
ellipse(60, 50, 80, 80);
ellipse(70, 50, 80, 80);
```

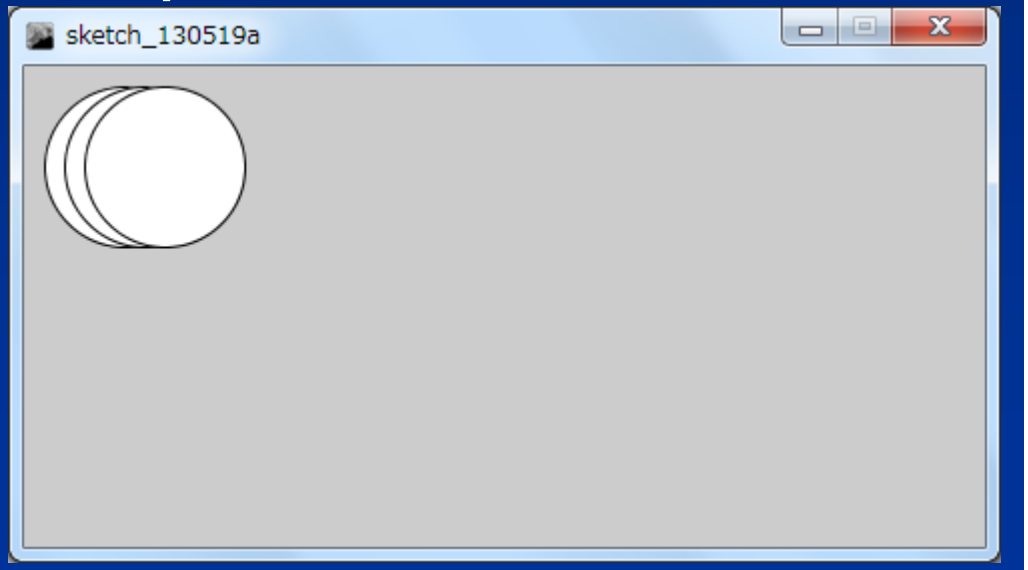
この調子で円をたくさん
表示するのは大変です.



繰り返し(for文)を使った円の表示(p04)

プログラム

```
size(480, 240);
for(int i = 0; i < 3; i++) {
  ellipse(50 + i * 10, 50, 80, 80);
}
```



iは**変数**と言います.
int i のように宣言します.
変数 i は繰り返しの
回数を数えます.

for文の仕組み

```
for ( int i = 0;  i < 3;  i++ ) {
        ①          ②       ③
④ ellipse(50 + i * 10, 50, 80, 80);

}
```

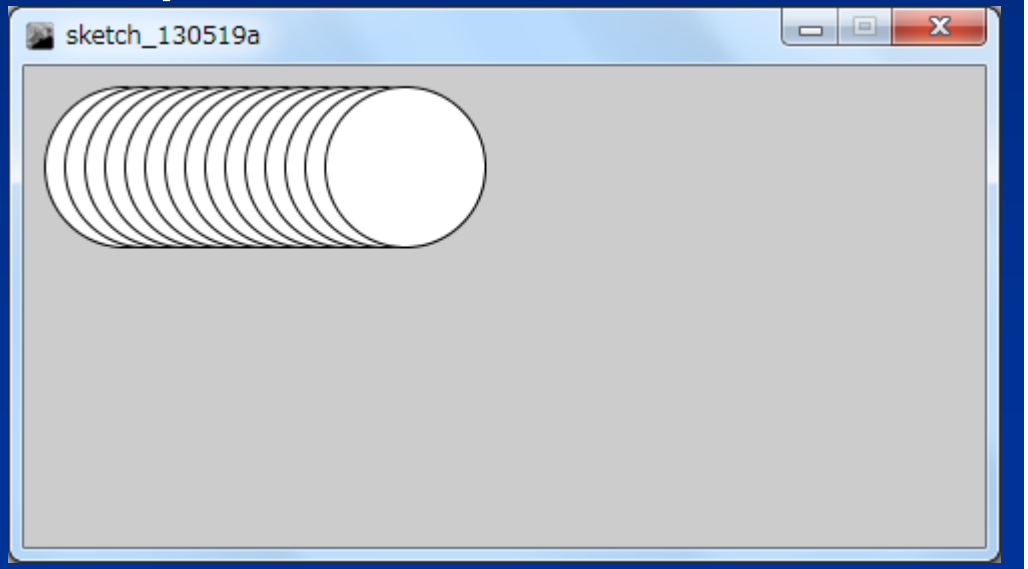
①が一回だけ実行されます（変数　i　を0にする）.

②のi < 3が成り立つ場合に④のブロックが実行されます.
③が実行されます（i++は，変数　i　に1を加えます）.
　　:　　:
②のi < 3が成り立つ場合に④のブロックが実行されます.
③が実行されます（i++は，変数　i　に1を加えます）.

表示個数の変更は容易(p05)

プログラム

```
size(480, 240);
for(int i = 0; i < 15; i++) {
  ellipse(50 + i * 10, 50, 80, 80);
}
```

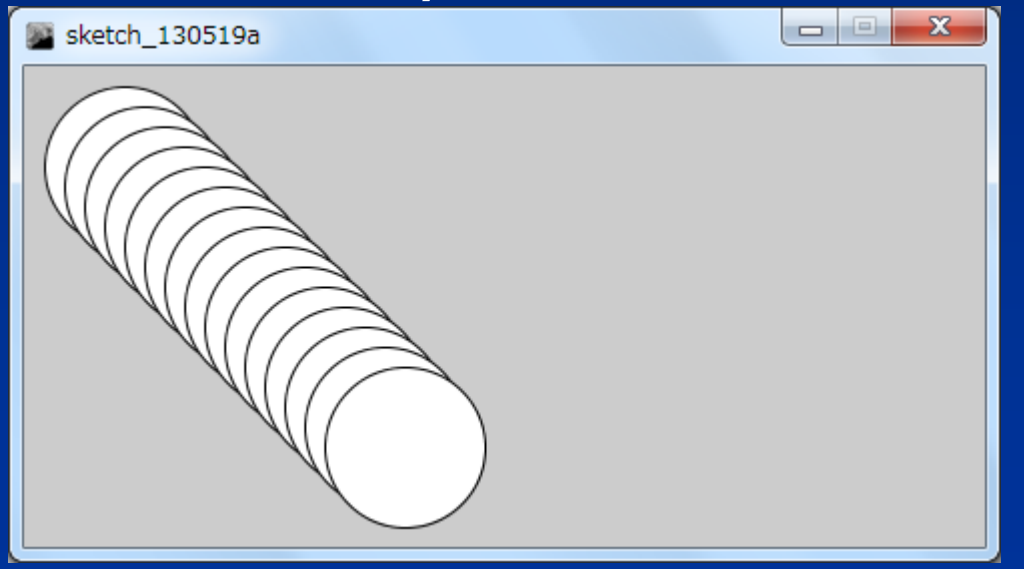


計算機は繰り返しが
大変得意です.

y座標も増加させると(p06)

プログラム

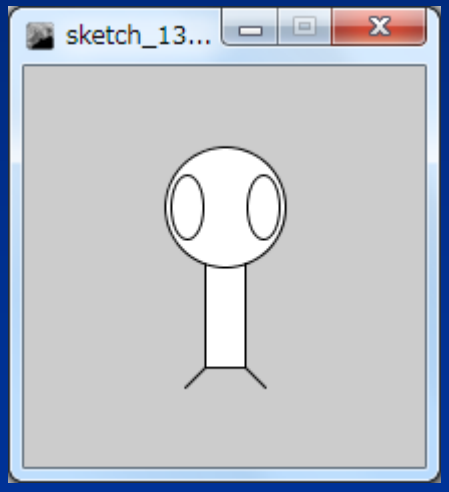
```
size(480, 240);
for(int i = 0; i < 15; i++) {
  ellipse(50 + i * 10, 50 + i * 10, 80, 80);
}
```



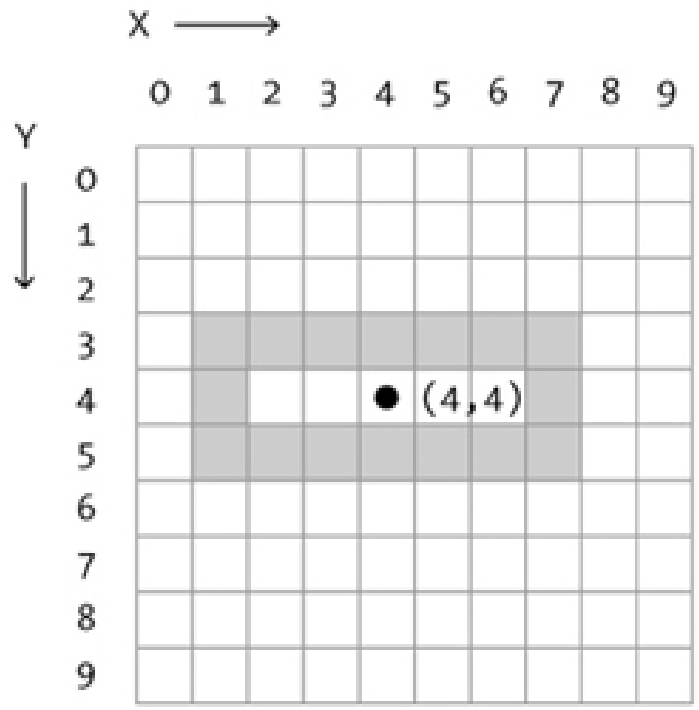
色々な形の組み合わせ(p07)

プログラム

```
size(200, 200);
rectMode(CENTER);
rect(100, 100, 20, 100);
ellipse(100, 70, 60, 60);
ellipse(81, 70, 16, 32);
ellipse(119, 70, 16, 32);
line(90, 150, 80, 160);
line(110, 150, 120, 160);
```



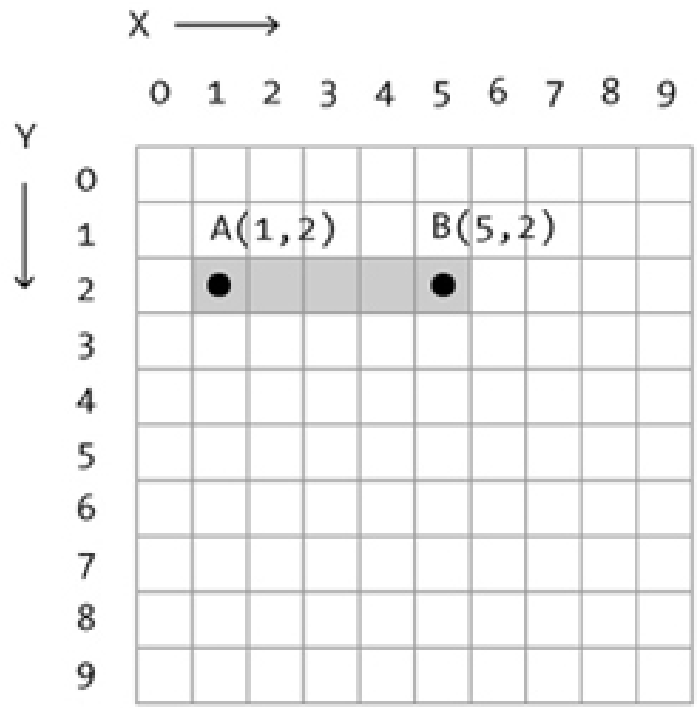
rect（長方形）




```
rectMode(CENTER);
rect(x,y,width,height);

Example:
rectMode(CENTER);
rect(4,4,7,3);
```

http://www.processing.org/learning/drawing/より掲載

line（線）




```
line(x1,y1,x2,y2);
```

Point A　Point B

Example:

```
line(1,2,5,2);
```

http://www.processing.org/learning/drawing/より掲載

マウスの座標を利用(p10)

プログラム

```
void setup() {
  size(480, 240);
}

void draw() {
  ellipse(mouseX, mouseY, 80, 80);
}
```

setup()関数は，プログラムの実行時に，一回だけ実行されます．

draw()関数は毎秒60回，実行されます．
mouseXとmouseYは，マウスの座標が入る変数です．

実行

マウスの移動に合わせて
円が表示されます.

マウス座標の表示

プログラム

```
void setup() {
  size(480, 240);
}

void draw(){
  ellipse(mouseX, mouseY, 80, 80);
  println(mouseX + " " + mouseY);
}
```

マウス座標をコンソールに
表示するプログラムの追加

マウス座標の表示（実行例）

sketch_130519a | Processing 2.0b8

File Edit Sketch Tools Help

JAVA

sketch_130519a §

```
void setup() {
  size(480, 240);
}

void draw(){
  ellipse(mouseX, mouseY, 80, 80);
  println(mouseX + " " + mouseY);
}
```

```
211 233
211 233
211 233
```

7

マウス座標が
コンソールに
表示されます．

ellipseをlineに変更(p12)

プログラム

```
void setup() {
  size(480, 240);
}

void draw(){
  line(0, 0, mouseX, mouseY);
  println(mouseX + " " + mouseY);
}
```

原点から直線が表示

sketch_130519a | Processing 2.0b8

File Edit Sketch Tools Help

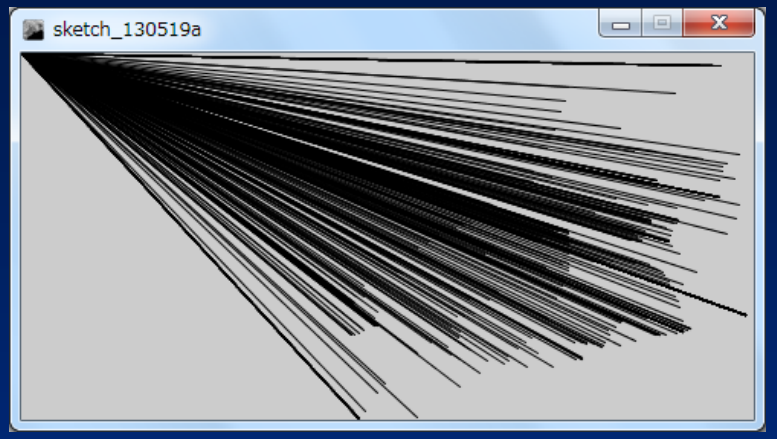
JAVA

sketch_130519a §

```
void setup() {
  size(480, 240);
}

void draw(){
  line(0, 0, mouseX, mouseY);
  println(mouseX + " " + mouseY);
}
```

```
479 2
479 2
479 2
```

7



分岐(if文)を使ったプログラム(p13)

プログラム

```
void setup() {
  size(480, 240);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

マウスのボタンが
押されると黒で
塗りつぶされます.

if文の仕組み

```
if ( mousePressed ) {
  fill(0);
} else {
  fill(255);
}
```

条件式

条件式が成立した時に
実行されるブロック

条件式が成立しない
時に実行されるブロック

マウスボタンが押されると黒

sketch_130519a | Processing 2.0b8

File Edit Sketch Tools Help

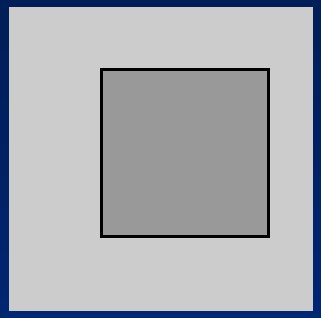
JAVA

sketch_130519a §

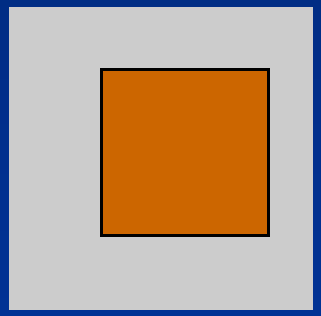
```
void setup() {
  size(480, 240);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

13

fill（塗りつぶし）



モノクロ
色の濃さを0から255の数字で指定
fill(153);
rect(30, 20, 55, 55);



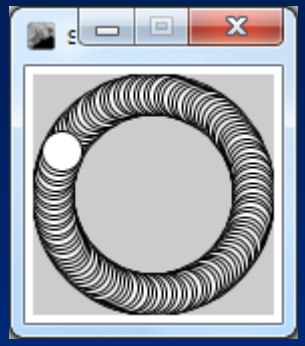
カラー
赤，緑，青の順に0から255の数字で指定
fill(204, 102, 0);
rect(30, 20, 55, 55);

http://www.processing.org/learning/drawing/より掲載

円の回転(p14)

プログラム

```
float rad = 0;

void setup(){
  size(120, 120);
}

void draw(){
  float x = 50 * cos(rad) + 60;
  float y = 50 * sin(rad) + 60;
  ellipse(x, y, 20, 20);
  rad = rad + 0.05;
}
```



プログラミングのおもしろさ

- プログラムの指示通りに計算機が動作
  - 指示されない事は何もしない（何もできない）
  - プログラムが動作した時の感動は格別
- 計算機の実行速度は非常に速い
  - 繰り返しの活用が威力を発揮
- アプリケーションプログラム(アプリ)の開発
  - 他人が利用し喜んでくれる事への満足感
  - 独創性や創造性を発揮

まとめ

- プログラミングを学ぶためには
  - 数学や理科の基礎知識（法則や理論の活用）
  - 英語の基礎知識（多くの言語は英語圏で開発）
- 本格的に学ぶためには
  - 「コンピュータプログラミングⅠ」で学びましょう

参考資料

- Processing
  - http://www.processing.org/
- Processingをはじめよう
  - Casey Reas、Ben Fry著　　船田　巧訳
  - オライリー・ジャパン
- Processingアニメーションプログラミング入門
  - 田中孝太郎著
  - 技術評論社
- Processingプログラミング入門
  - 田原淳一郎著
  - カットシステム