

情報メディア 基盤ユニット

4月19日

変数、SETUPとDRAW

情報メディア学科佐藤尚

先週やったこと

ウィンドウを開いて、

色を決めて、

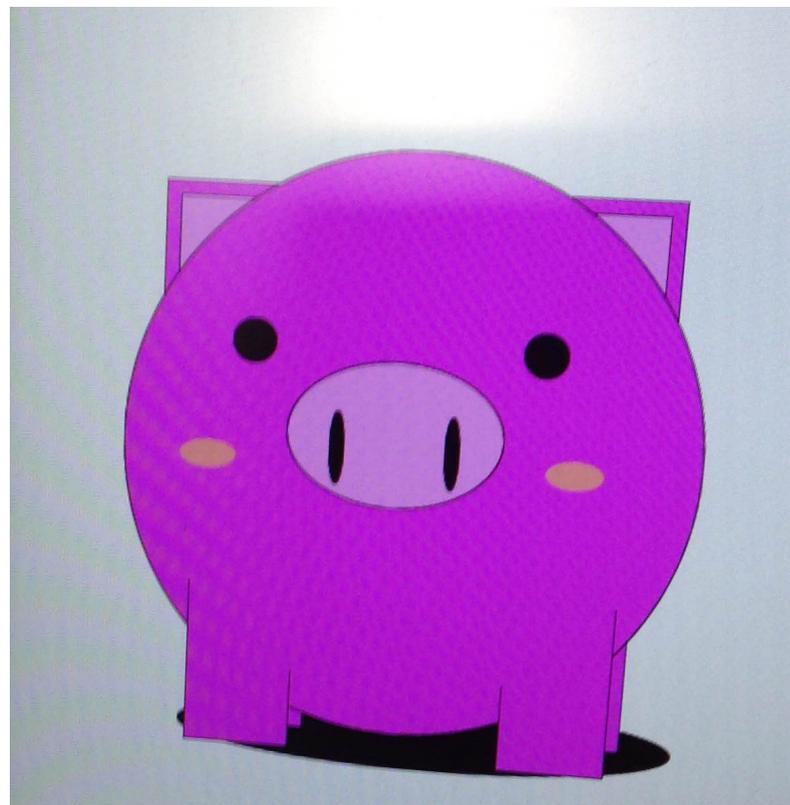
図形を描画する

size

fill,stroke,

rect,ellipse,line,point

などの命令を使った



プリント1枚（クイズ）があります。

プリントを持って行って下さい。

授業で配布したプリントなどは

<http://www.sato-lab.jp/imfu>

にアップされています。



Processingのスケッチの 基本形

Processingで作られたプログラムのことをSketch（スケッチ）と呼ぶ。

基本形その1

```
size(400,400);  
ellipse(200,200,80,80);  
ellipse(50,50,50,50);  
ellipse(300,350,80,80);
```

基本形その2

```
void setup(){  
  size(640,480);  
  smooth();  
}  
void draw(){  
  if(mousePressed){  
    fill(0);  
  }else{  
    fill(255);  
  }  
  ellipse(mouseX,mouseY,80,80);  
}
```

今日やること

変数

setupとdraw（基本形その2の基本要素）

Processingで簡単なアニメーションが作れる

前回：単に図形を描くだけ

変数

変数：数学で出てきた x みたいなやつ

$$f(x) = x^2 + 5x + 6$$

Processingでの変数

- コンピュータのメモリに名前をつけて、値を保存したり、読み出して利用するもの
- メモリに名前：変数名
- 変数には数値以外の情報を保存することが出来ます。
- どんな種類の情報（データ）か：データ型

コンピュータの 基本的な構造

計算する部分 (CPU)

データや命令を記憶するメモリ

変数の使用例その1

変数を利用するためには：**変数宣言**

どんな種類のデータか？

- 整数：int integer
- 実数：float floating point number
 - 小数点つきの数

変数名は

- アルファベットまたは_（アンダースコア）またはで始まり、アルファベット、数字、_、\$を組み合わせで作られる単語です。

変数名の例

x	x1	price	_price
y	y0	hitPoint	hit_point
i	i1	hitPoint1	hit\$point1
j	j0	hitPointOfMage	hitPointOfMage
k	k3	hitPoint123Mage	hitPoint123Mage
		kait	kait
		Qwerty	Qwerty
		birthday	birthday

変数名としてダメな例

1x 1x1 Hit^point size

2y 5y0 Hit^point1 fill

3i @i1 hitPointOfMage rect

4j *j0 Mage? if

5k 00k3 for

などなど

予約語（Processingが使うことになっている単語）は変数名として利用できません。

変数の利用

変数に値を保存する。代入する(assign)。

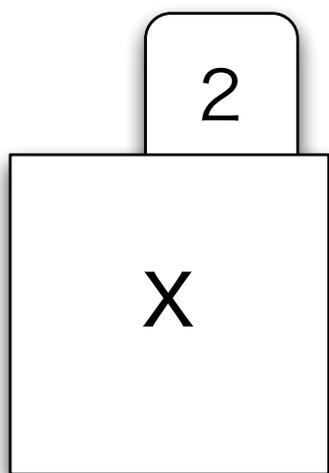
変数に値を代入する。

- 変数名=値；

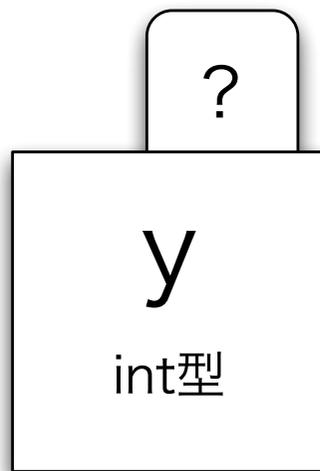
同じ値を再利用

変数

値の入っている箱に名前をつけたもの



```
int x;  
x = 2;
```



```
int y;
```

変数名として使える

- kait
- KAIT
- shimoogino1030
- shimoogino1030Kait

変数名として使えない

- 1030shimoogino
 - Kait.1030
- 

変数の使用例その2

= : 値が等しいという意味ではない。

=: 値を代入するという意味です。

簡単な式を利用する

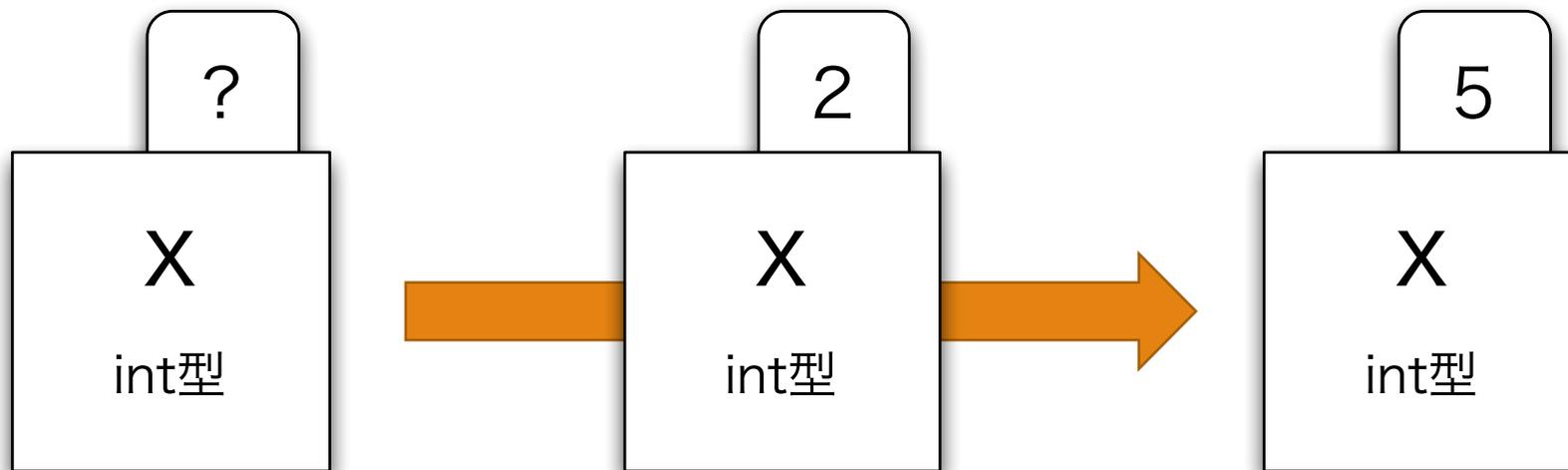
式で利用出る**演算子**の例

- + : 足し算
- - : 引き算
- * : かけ算
- / : 割り算
- % : 余り

```
int x;
```

```
x = 2;
```

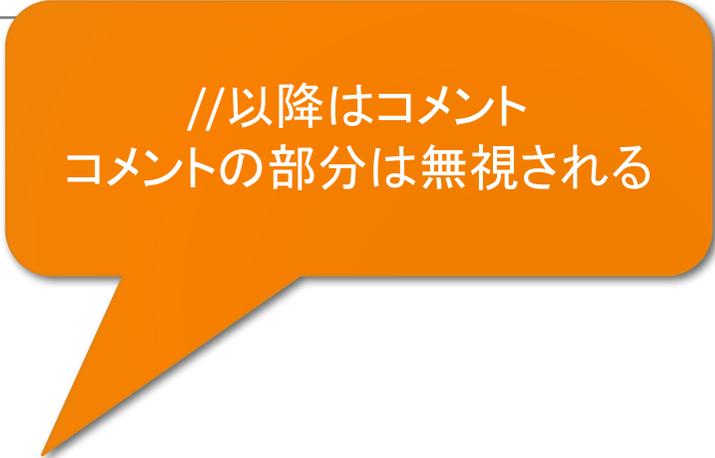
```
x = x + 3;
```



コメント

```
size(480,120);  
smooth();
```

```
ellipse(75,60,80,80); // Left  
ellipse(175,60,80,80); // Middle  
ellipse(275,60,80,80); // Right
```



//以降はコメント
コメントの部分は無視される

コメント

```
size(480,120);  
smooth();  
/*  
ellipse(75,60,80,80);  
ellipse(175,60,80,80);  
ellipse(275,60,80,80);  
*/
```

/*~*/の間はコメント
コメントの部分は無視される

{と}

{…}は命令の集まりを一塊にする

通常は

{

命令の集まり

}

と使う

{と}

通常は、{と}で作った塊の前に何かの指示が来ることが多い。

例えば、

```
void setup(){  
  size(480,120);  
  smooth();  
}
```

```
void setup(){  
  size(480,120);  
  smooth();  
}
```

インデント、字下げ

{と}で作る塊の中身をハッキリさせるためにインデント (indent, 字下げ) を行うことが一般的。

```
void setup(){  
  size(480,120);  
  smooth();  
}
```

```
void setup(){  
    size(480,120);  
    smooth();  
}
```

基本形その2の構造

プログラムの開始時に1回だけ
実行される命令の集まり

```
void setup(){  
.  
.  
.  
}
```

プログラムの開始後に定期的に
実行される命令の集まり

```
void draw(){  
.  
.  
.  
}
```

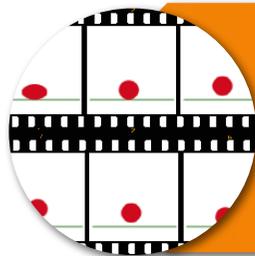
一般的には1秒間に60回繰り返す

アニメーションを作りたい



アニメーションとは？

連続して変化する絵や物により発生する仮現運動を利用した映像手法



ちよつとずつ異なる位置や大きさを図形を
描画することで動きを再現する。

```
void setup() {  
  size(480,120);  
  smooth();  
}  
  
void draw() {  
  background(255,255,255);  
  stroke(255,0,0);  
  line(x,0,x,height);  
  fill(255,0,0);  
  ellipse(x,y,d,d);  
  x = x + dx;  
  y = y + dy;  
  println(y);  
}
```

図形を表示する位置や大きさなどを変数によって指定し、
その変数の値を変化させることで、表示位置や大きさなどを変え、
アニメーションを実現する。

変数を宣言する場所

当面は、プログラムの先頭で変数を宣言することにする
詳しいことは、後ほどやります。

```
int y = 60;  
int x = 0;  
int d = 80;  
int dx = 1;
```



プログラムの先頭で
変数宣言を行う

```
void setup(){  
  命令の集まり  
}
```

```
void draw(){  
  命令の集まり  
}
```

```
int y = 60;
int x = 0;
int d = 80;
int dx = 1;
```

```
void setup(){
  size(480,120);
  smooth();
}
```

```
void draw(){
  background(255,255,255);
  stroke(255,0,0);
  line(x,0,x,height);
  fill(255,0,0);
  ellipse(x,y,d,d);
  x = x+dx;
  println(x);
}
```

```
int y = 60;
float x = 0;
int d = 80;
float dx = 0.5;
```

```
void setup(){
  size(480,120);
  smooth();
}
```

```
void draw(){
  background(255,255,255);
  stroke(255,0,0);
  line(x,0,x,height);
  fill(255,0,0);
  ellipse(x,y,d,d);
  x = x+dx;
  println(x);
}
```

```
int y = 120;  
int x = 240;  
int d = 80;  
int dy = -1;
```

```
void setup(){  
  size(480,120);  
  smooth();  
}
```

```
void draw(){  
  background(255,255,255);  
  stroke(255,0,0);  
  line(x,0,x,height);  
  fill(255,0,0);  
  ellipse(x,y,d,d);  
  y = y + dy;  
  println(y,dy);  
}
```

システム変数 (8ページ)

mouseX, mouseY

- マウスカーソルのX座標とY座標の値を記憶している変数

width, height

- 表示しているウィンドの横幅と高さの値を記憶している変数

などなど

システム変数は変数宣言する必要がない

println

変数の値を表示する命令

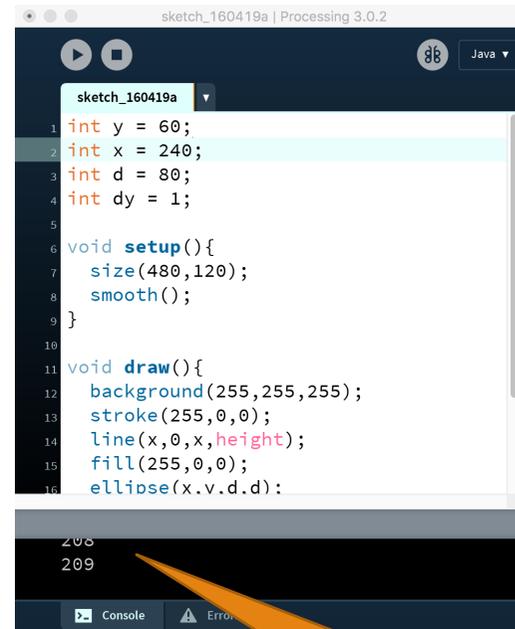
使い方の例

変数xの値を表示する

- `println(x);`

変数xとyの値を表示する

- `println(x,y);`

A screenshot of a Processing IDE window titled "sketch_160419a | Processing 3.0.2". The code editor shows the following code:

```
1 int y = 60;  
2 int x = 240;  
3 int d = 80;  
4 int dy = 1;  
5  
6 void setup(){  
7   size(480,120);  
8   smooth();  
9 }  
10  
11 void draw(){  
12   background(255,255,255);  
13   stroke(255,0,0);  
14   line(x,0,x,height);  
15   fill(255,0,0);  
16   ellipse(x.v.d.d);
```

The console window at the bottom shows the output:

```
208  
209
```

An orange callout bubble points to the console output.

ここに表示される