

- 
- ゲームコントローラーの課題のメールでの提出は4月19日（金）の午前9時まで受け付けます。
  - 提出物に学籍番号や名前を忘れている人がいます。提出物を提出する際には気を付けて下さい。

# 情報メディア 基盤ユニット

---

4月16日

変数、SETUPとDRAW

情報メディア学科佐藤尚

---

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。

# 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

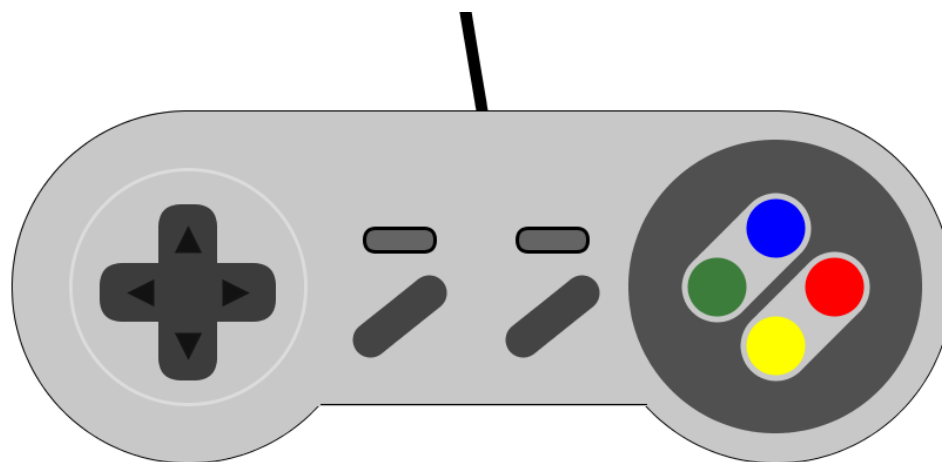
size

fill,stroke,

rect,ellipse,line,point

などの命令を使った

色の指定(RGB)



# コメント

---

//以降はコメント  
コメントの部分は無視される

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

```
// comment!!
```

```
ellipse(75,60,80,80); // Left
```

```
ellipse(175,60,80,80); // Middle
```

```
ellipse(275,60,80,80); // Right
```

# コメント

---

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

```
/*
```

```
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

# 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);
```

静的なプログラム

## 基本形その2

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

動的なプログラム

# 今日やること

---

変数

setupとdraw（基本形その2の基本要素）

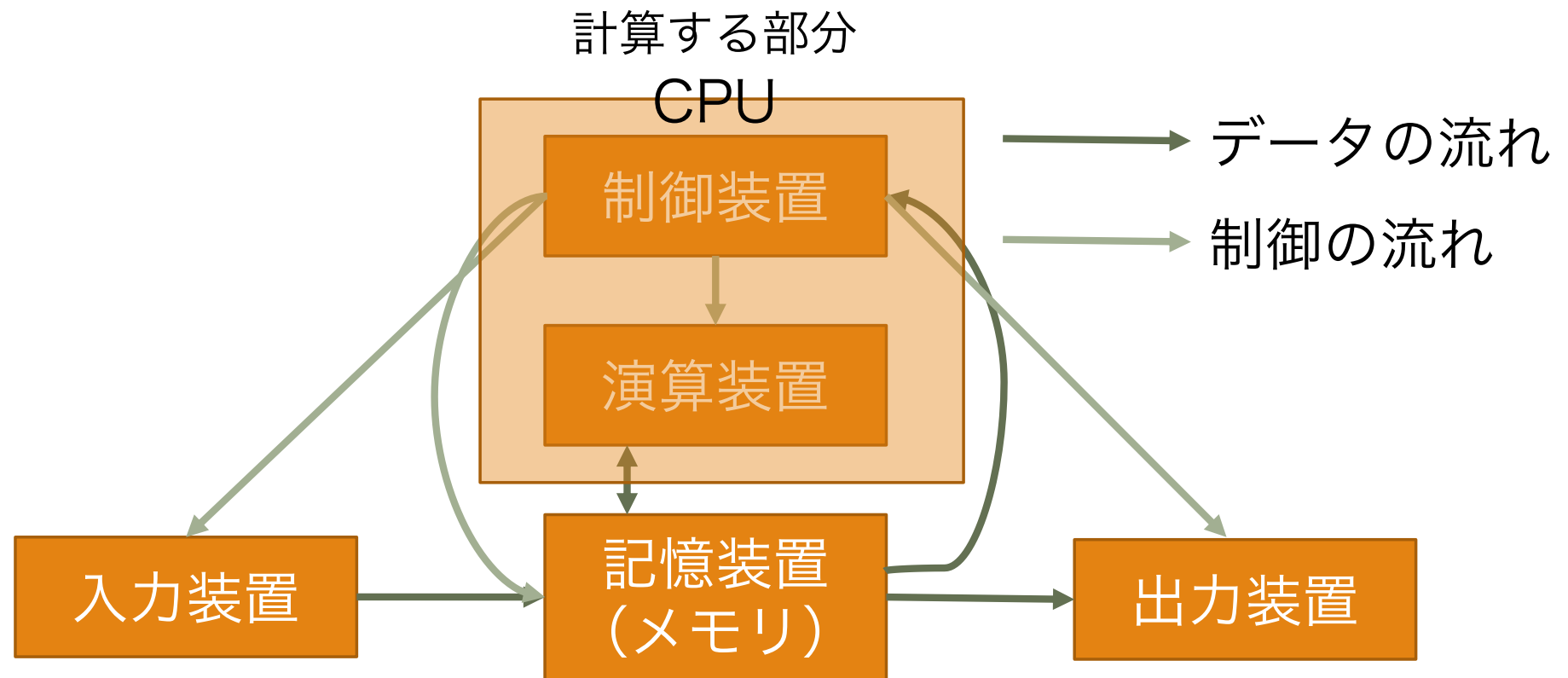
Processingで簡単なアニメーションが作れる

前回：単に図形を描くだけ



# コンピュータの 基本的な構造

---



データや命令を記憶するメモリ

# 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(variable)

- 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（整数、実数）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

# 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
  - 浮動小数点
  - 小数点付きの数

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、\_、\$を組み合わせて作られる単語です。

# 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 | hitPoint\$Mage  |
|   |    | 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

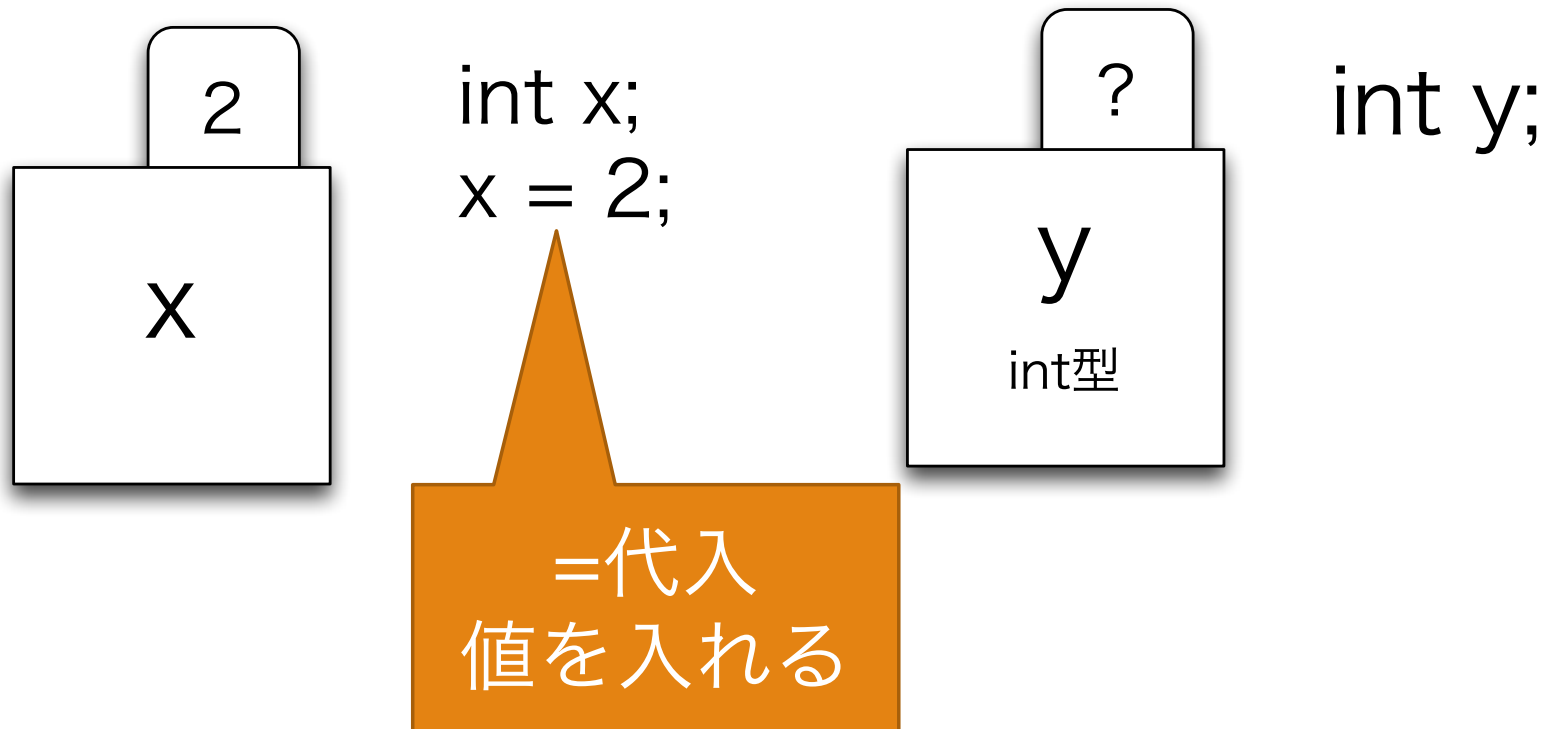
などなど

**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

---

値の入っている箱（メモリ）に名前をつけたもの



# 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；
- $x = 2$ ;
- 変数名=式;

同じ値を再利用

---

## 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

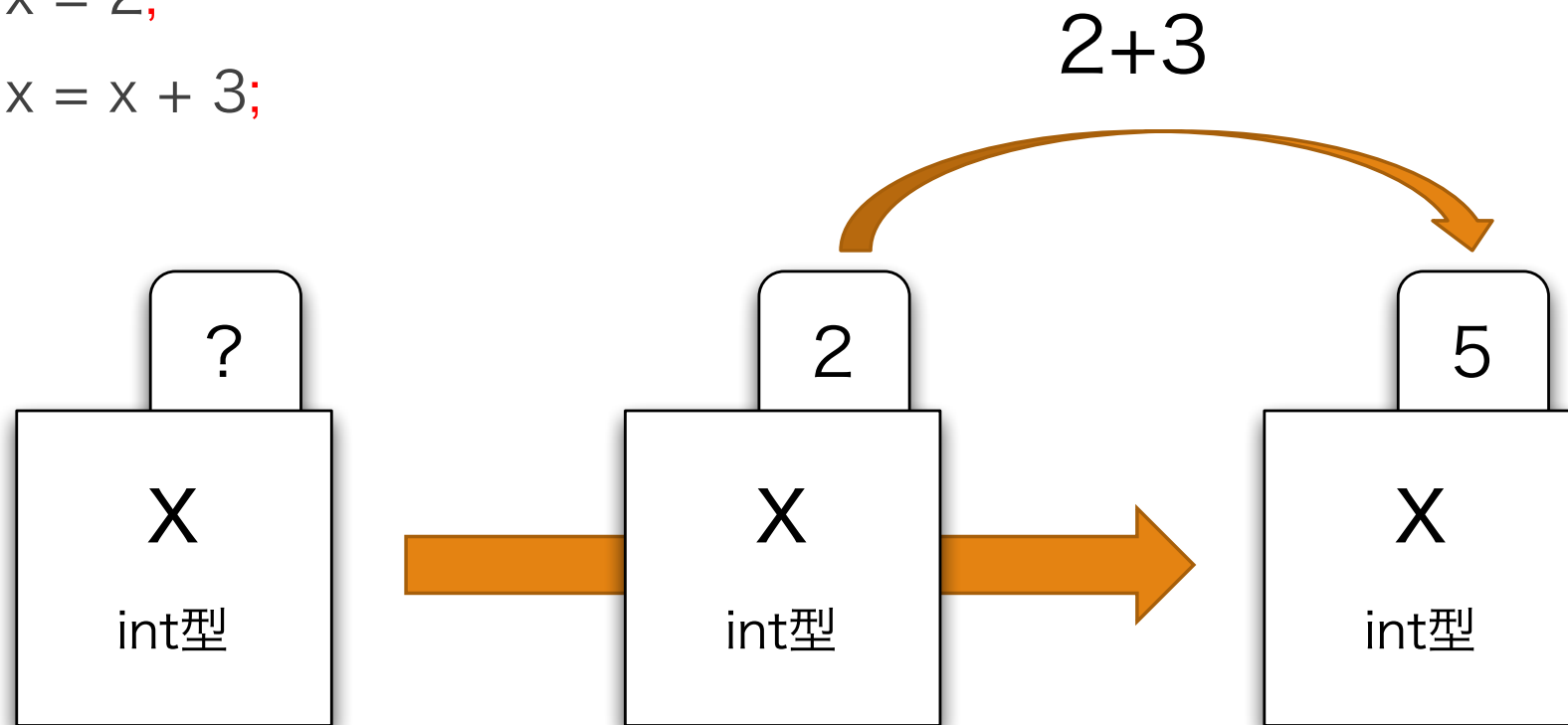
- + : 足し算 (+)       $x + 2$
- - : 引き算 (-)       $x - 2$
- \* : かけ算 (×)       $x * x$
- / : 割り算 (÷)       $x / 10$
- % : 余り       $x \% 10$

---

```
int x;
```

```
x = 2;
```

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



# {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);  
  fill(255,255,10);  
}
```

```
void setup(){  
  size(480,120);  
  fill(255,255,10);  
}
```

# 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);  
  fill(255,255,10);  
}
```

```
void setup(){  
    size(480,120);  
    fill(255,255,10);  
}
```

# インデントが面倒

---

自動フォーマットを利用する  
編集＞自動フォーマット

# 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(){
```

```
•  
•  
•
```

```
}
```

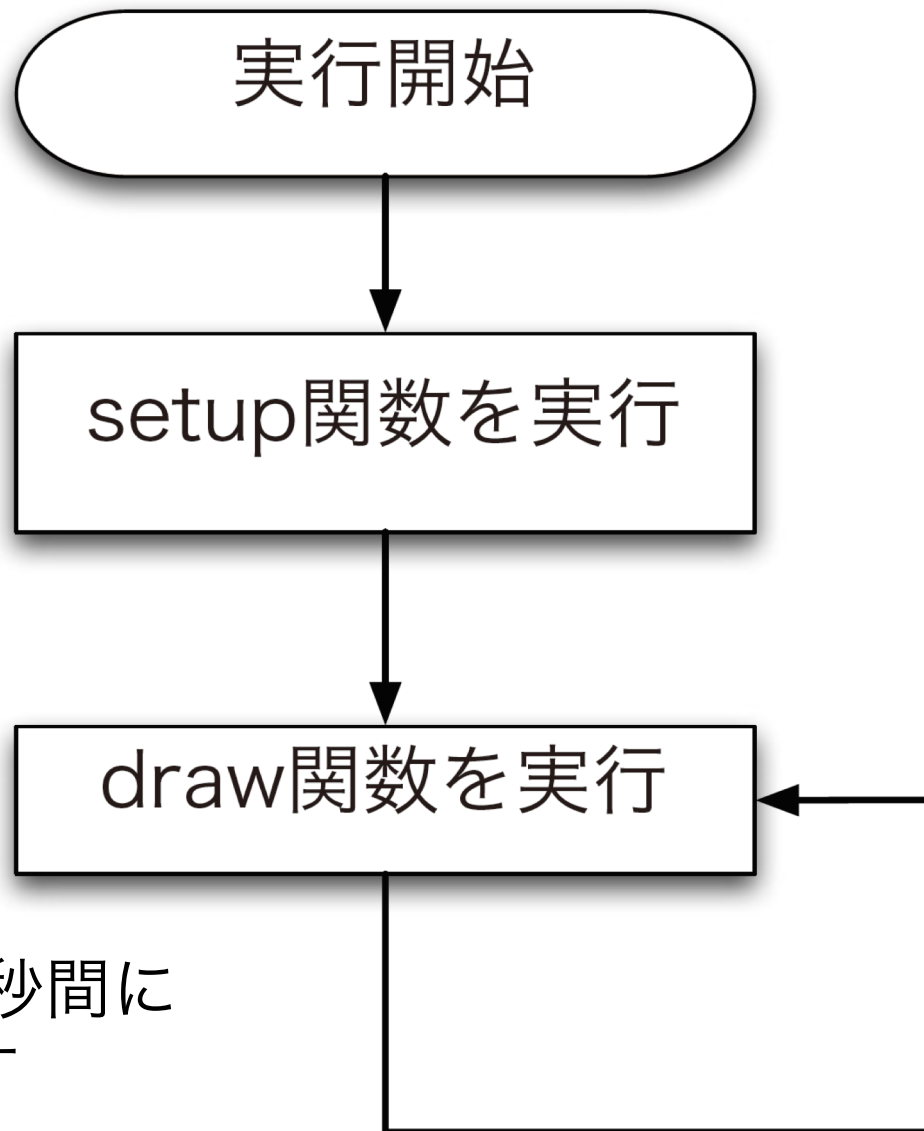
プログラムの開始後に定期的に  
実行される命令の集まり

```
void draw(){
```

```
•  
•  
•
```

```
}
```

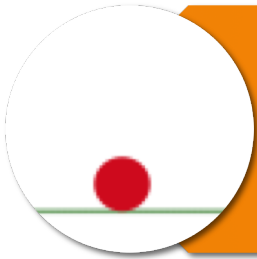
一般的には1秒間に60回繰り返す



一般的には1秒間に  
60回繰り返す

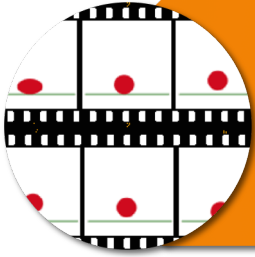


# アニメーションを作りたい



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



ちよっとずつ異なる位置や大きさに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+dy;  
  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);  
}
```

```
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);  
}
```

```
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);  
}
```

```
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 (24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

# 変数の値の確認

---

printlnを使う

変数に入っている値を調べる事が出来る

# println

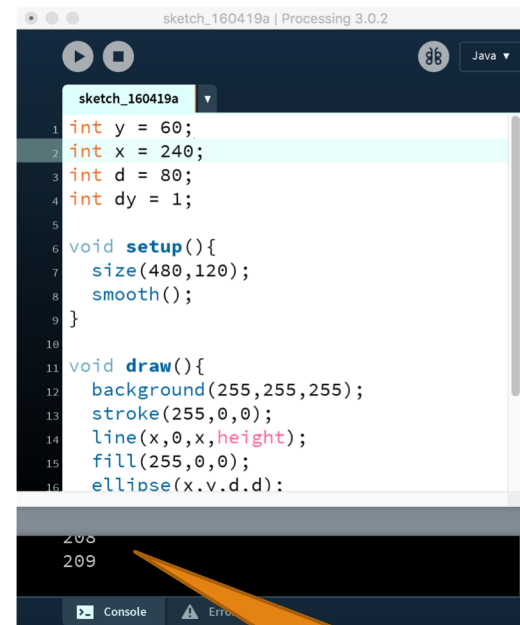
変数の値を表示する命令  
使い方の例

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);`



```
sketch_160419a | Processing 3.0.2
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):
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

ここに表示される

# デバッグ

---

バグ

作成したプログラムが意図通りに動作しないこと

デバッグ

バグを修正する作業



# 3種類のバグ

---

構文エラー

文法に関連したエラー

実行時エラー

プログラムの実行時に発生するエラー

論理エラー

構文や実行時のエラーではないが、意図通りに動かない