

後ろ5列は着席禁止
3人掛けの中央は着席禁止

プログラミング入門 Processingプログラミング 第5回

九州産業大学 理工学部情報科学科
神屋郁子
(pp@is.kyusan-u.ac.jp)

時限	クラス
水1	機械 (クラス3)
水2	機械 (クラス1)
水4	電気 (B1、B2)

第5回の内容

- 変数
- 丸を動かす

2

プログラムを構造にする

- 命令を並べて命令を作る

```
void setup(){  
  最初に1度  
  実行したい処理  
  (初期設定)  
}  
  
void draw(){  
  繰り返し  
  実行したい処理  
  (描画)  
}
```

setupとdrawという命令の
名前は決まっている。

setupとdrawという2つの
命令を作ることで動きの
あるプログラムを作ること
ができる。

3

マウスポインタの位置

ellipse(0, 0, 50, 50);

- 数値はすべて0や50という数値
- 座標は変化しない

ellipse(mouseX, mouseY, 50, 50);

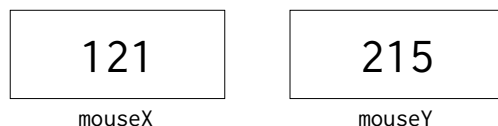
- mouseXやmouseYは、現在のマウスの座標が数値として格納されている変数
- 値が変化するので座標も変化する
- 実は、mouseXやmouseYは、変化する値を扱う**変数**というもの

4

mouseXやmouseYという変数

`ellipse(mouseX, mouseY, 50, 50);`

- `mouseX`や`mouseY`という名前の箱があり、現在のマウスの位置が数値として入っている



5

変数

- データを格納する入れ物
- 変化するデータを扱うのに使う
- processingで用意されている変数
 - `mouseX`, `mouseY`, `width`, `height`
- 自分で変数を作ることもできる

6

変数を作る（変数宣言）

- 変数宣言の構文

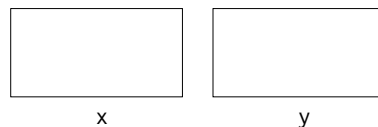
型 **変数の名前**;

- **型** 格納するデータの種類

- `int`: 整数
- `float`: 小数

- **変数の名前** 自分でつける

- 例 `int score;`
`int x, y;` (複数まとめて)



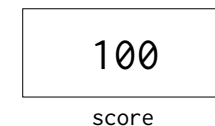
7

値を格納する（代入）

- 代入の構文

変数の名前 = データ;

- データは具体的な数値であったり式であったり
 - 例 `score = 100;`



8

```
int score;
```

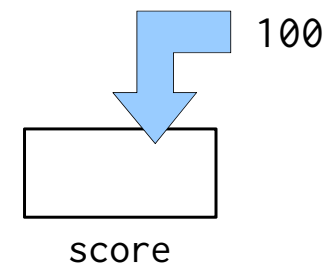
- 整数を扱うscoreという名前の変数ができる



9

```
score = 100;
```

- scoreという名前の変数に100を格納する。



10

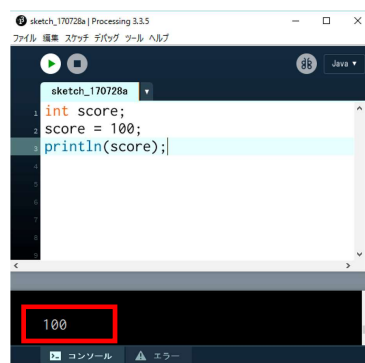
画面にデータを表示する

- printlnメソッドを使用し、画面にデータを表示

```
println(データ);
```

- scoreの値を表示

```
println(score);
```



11

演算

- + 足し算
- - 引き算
- * 掛け算
- / 割り算
- % 割った余り

12

変数の値を変化させる

- ある値増やす
変数の名前 = 変数の名前 + 値;
x = x + 10;
- 複合代入演算子(+=)である値、増やす
変数の名前 += 値;
x += 5;
-,*,/,%についても同様
- インクリメント演算子(++で1増やす) x++;
- デクリメント演算子(--で1減らす) x--;

13

変数の宣言と初期化

- 宣言 代入
- int score; score = 100;
- 宣言と代入をまとめて記述
- int score = 100;

14

作業

1. 変数の概要 (宣言、代入、演算)
2. 変数でカウントアップ
3. 変数
4. 変数で移動
5. 変数の追加

15

```
int x;  
x = 10;  
x++;  
x += 2;  
x = x+3;
```

演習1

```
int x;
```

演習2

```
void setup(){  
  frameRate(5);  
  x = 0;  
}
```

```
void draw(){  
  x++;  
  println(x);  
}
```

```
int x,y;
```

演習3

```
void setup(){  
  size(400,400);  
  frameRate(30);  
  x = 10;  
  y = 10;  
}
```

```
void draw(){  
  background(0);  
  ellipse(x,y,20,20);  
}
```

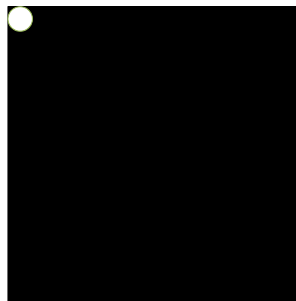


```
void draw(){  
  background(0);
```

演習4
(drawメソッドのみ)

```
x += 10;  
y += 10;
```

```
ellipse(x,y,20,20);  
println(x + ", " + y);  
}
```

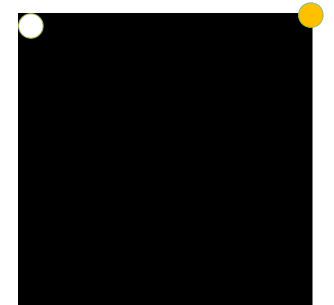


```
int x,y;
```

```
int x1,y1;
```

```
void setup(){  
  size(400,400);  
  frameRate(30);  
  x = 10;  
  y = 10;  
  x1 = 400;  
  y1 = 0;  
}
```

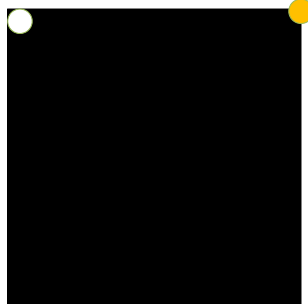
- 変数をもう1セット用意
(例えば、x1とy1の変数宣言)
- **setup**で初期値
(右上はx座標が400でy座標が0とするとよい。または400と書く代わりに右や下の座標は以下のシステム変数を利用すると良い。width:横幅 height:高さ)



```
void draw(){
  background(0);
  x += 10;
  y += 10;
  x1 -= 10;
  y1 += 10;
  ellipse(x,y,20,20);
  ellipse(x1,y1,20,20);
}
```

演習5
(drawメソッドのみ)

• drawで増減と描画



色空間

- 各色を数値の組み合わせで表現する方法または表現可能な色域
- RGBとHSB
 - RGB：赤(Red)緑(Green)青(Blue)
 - HSB：色相(Hue)彩度(Saturation) 明度(Brightness)

23

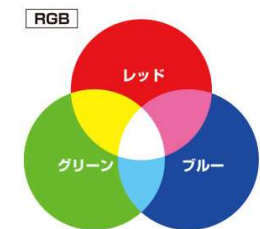
作業

- 6. 応用
- 7. K's Lifeで提出

22

RGB

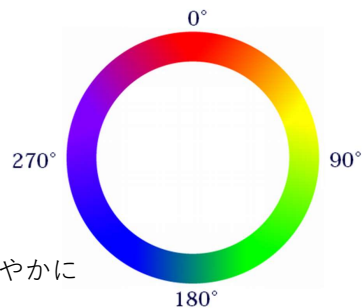
- 赤(R)緑(G)青(B)の光の三原色で色を指定する方法
 - 赤、青、緑の色の量を0~255の数値で表現
 - 赤なら(255, 0, 0)
 - 緑なら(0, 255, 0)
 - 青なら(0, 0, 255)
 - 黄色は赤と緑の両方なので(255, 255, 0)
 - すべて混ぜると白(255, 255, 255)
 - なにもないと黒(0, 0, 0)



24

HSB

- 色相(Hue)彩度(Saturation)明度(Brightness)を数値で表現
- 色相：色合い
この値を変えると色合いが変化する
 - 赤は0
 - 緑は120
 - 青は240
- 彩度：鮮やかさ
 - 0~99(%) 白から始まり値が増えるほど鮮やかに
- 明度：明るさ
 - 0~99(%) 黒から始まり値が増えるほど明るく



25

プログラムの提出（演習点）

- 動きのあるプログラムを作ってみよう！
- K's Lifeのレポート機能から
- 締め切りは6日後の23:59
 - それ以降も受け取るが**減点**する
 - もっと改良したくてもこの時間に一度提出しよう
 - （削除して提出しなおし可能）
- 評価はK's Life上ではしない

26

提出する内容

- 以下をコメントとして入力
 - 今回の内容の概要、工夫した点
 - 質問（何かあれば）・感想（簡単だった・難しかったなど）
 - 入らない場合は3つ目の提出ファイルとして追加してもよい
- 提出ファイルとして以下の2つ
 - プログラム(*.pde)
 - 実行画像(PNGまたはJPG)
- 提出ファイルの名称にはそれぞれ学籍番号を入力

27