移動するボール　変数に慣れよう

(a) 変数宣言：変数をつくる。　　**型　変数の名前;**

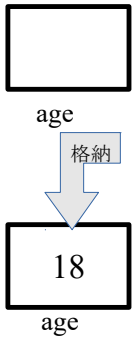
型は、データの種類　**int:整数**　　**float:小数**　　**String:文字列**

名前は、変数の用途にあわせて分かりやすい名前を自分でつける。

変数の名前をカンマ(,)で区切って複数まとめて宣言することもできる。

例　int age;

　　int x, y;　　複数の整数型の変数をまとめて宣言



(b) 代入：変数に値を格納すること。　**変数の名前 = 値;**

例　age = 18;

(c) 演算　+ - * / %　(足し算、引き算、掛け算、割り算、割り算の余り)

ある値増やす

変数の名前 = 変数の名前 + 値;	x = x + 10;	代入を使う。
変数の名前 += 値;	x += 5;	複合代入演算子
1増やす　**変数の名前++;**	x++;	インクリメント演算子
1減らす　**変数の名前--;**	x--;	デクリメント演算子

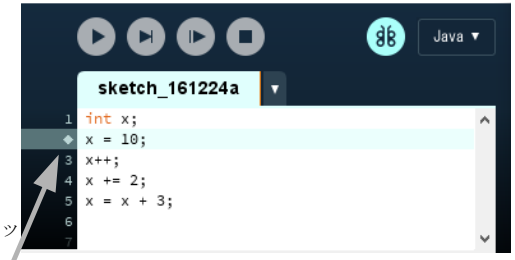
(d) 変数の宣言と初期化：宣言と代入をまとめて記述　　**型 変数の名前 = 値;**

例　int score = 0;

1. 変数の概要（宣言、代入、演算）

以下のプログラムを入力

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



Java▽の左のデバッグボタンをクリック（ステップボタンなどが増える）

2行目の2の数字をクリック(♦の表示に)

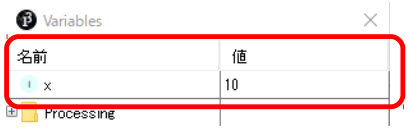
実行(4つのボタンの一番左)すると、2行目の♦が▸になり実行が止まる。

ステップ(4つのボタンの左から2つ目)を押すと

1行実行が進み、x = 10;が実行されて、

Variables のウインドウで変数 x の値(10)を確認。

ステップを繰り返し、命令と x の値の変化を確認。



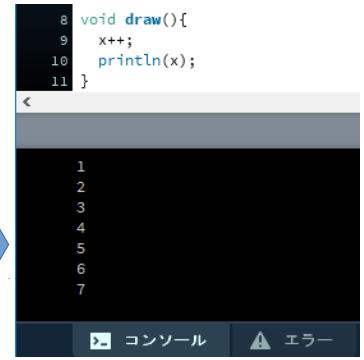
2. 変数でカウントアップ

新たにウインドウを開く。([ファイル]メニューから[新規])

```
int x;

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

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



setup の{から}までの部分が、初期設定の処理で実行時に一度実行される。draw の{から}の部分が、描画の部分で、1秒間に frameRate の命令で設定した回数繰り返し実行される。

x++; を書き換えてみよう。2ずつ増やすには？

3. 変数

新たにウインドウを開く。(ファイルメニューから新規)

左上の隅に変数を使って丸を描くプログラムを入力する。

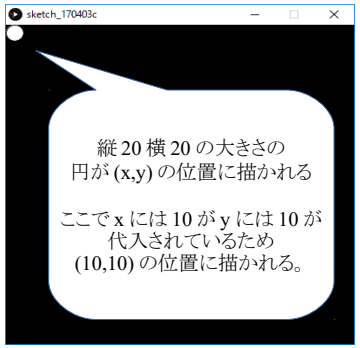
変数を宣言し、setup の中で位置を指定する。　draw の中でその変数の座標に丸を描く。

```
int x, y;

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

void draw(){
  background(0);

  ellipse(x, y, 20, 20);
}
```



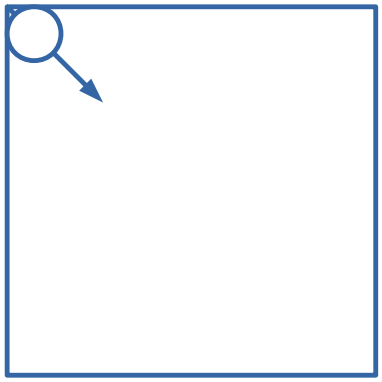
4. 変数で移動

左上から右下に図形を動かしてみよう。座標の変数の値つまり x や y を増やす命令を追加すればよい。

```
int x, y;

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

void draw(){
  background(0);
  ここに座標を増やす処理を追加
  ellipse(x, y, 20, 20);
}
```



タイピングに余裕がある人は、println(x + ",," + y); を draw の中に追加してみよう。
値が変化していくのをコンソール（下の黒いところ）で確認できる。
右下まで行くと変化がなくなり面白くない！ → 次回、条件分岐で対応

5. 変数の追加：ボール（丸）を違う角から動かしてみよう。余裕があれば四隅から。

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、draw で増減と描画
setup での最初の値の代入の際に、右上は x 座標が 400 で y 座標が 0 とすると良い。
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よい。

width ：ウインドウの横幅　　height ：ウインドウの高さ

6. 応用（これまでの復習、自由にアレンジしよう）

図形を消さずに描き続けると残骸が残る。（draw メソッド中の background(0);を削除）

枠なしにする。setup メソッド内に noStroke(); を追加する。

ボールの色を設定しよう。

乱数をつかっても面白い random(256)で 0 以上 256 未満の乱数

緑で乱数　　fill(0,random(256),0);

赤緑青すべて乱数　　fill(random(256),random(256),random(256));

HSB モード colorMode(HSB, 359, 99, 99);を setup 中に書く。

fill(H,S,B); で、H:色相,S:彩度,B:輝度を指定する。H の値は 0 で赤、120 で緑、240 で青

デフォルトだと RGB モード　colorMode(RGB,255,255,255);

形を変えてみよう。

徐々に大きく（x,y を大きさでも使う）　ellipse(x, y, x, y);

大きさを乱数に　ellipse(x, y, random(20), random(20));

徐々に大きくかつ乱数　ellipse(x, y, random(x), random(y));

図形そのものの変更　rect(x, y, random(x), random(y));

点滅

int c;を先頭に追加し、draw の中の fill を以下のように変更すると赤く点滅する。

```
c += 4;
fill(c % 256, 0, 0);
```

緑の点滅にしたければ

```
fill(0, c % 256, 0);
```

青なら

```
fill(0, 0, c % 256);
```

HSB モードで色を指定したい場合は、

```
colorMode(HSB, 359, 99, 99);にして
fill(c % 360,  99, 99);
```

ここで % は割り算の余り、z % 360 とすると z がいくら増えても 0 から 359 までの数値になる。

プログラムの提出（5 週）　K’s Life のレポート機能にて提出

・以下をコメントとして入力

（文字数の制限で入らない場合は３つ目の提出ファイルとして追加してもよい）

今回の内容の概要、工夫した点

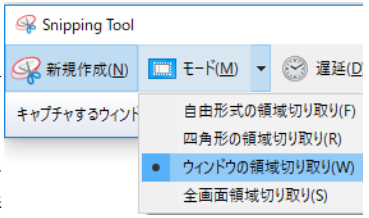
質問（何かあれば）・感想　難しかった。簡単だった。進め方が早い。遅い。など

・提出ファイルとして 1 つ目のプログラム(*.pde)と実行画像(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

実行画像の作り方

（frameRate(30); を frameRate(5);ぐらいにして動作を遅くしておくと良い。）



1. Snipping Tool を起動する。
2. モードの横の三角形をクリックし、「ウィンドウの領域切り取り」に切り替える。（もしくは「四角形…」）
3. キャプチャしたいウインドウをクリックする。（または範囲をドラッグ。）
4. ファイルメニューから名前を付けて保存する。（または、フロッピーディスクのアイコン）