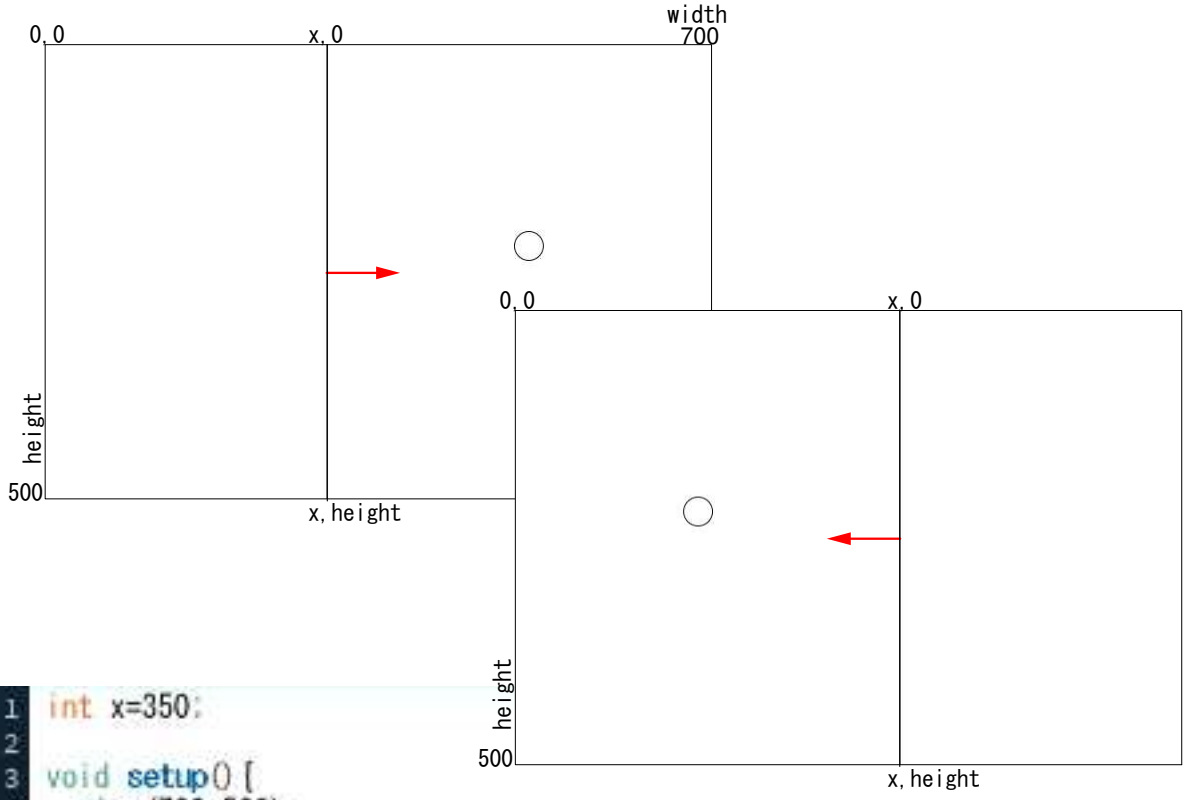
Processing

第 7 回

松田小学校／寄小学校

7ーステップ　0：前回の復習だよ

白丸が、線の右側にあるときは線が右に、左側にあるときは線が左に動くようにコードのマスを埋めてみよう。




```
1  int x=350;
2
3  void setup() {
4    size(700, 500);
5  }
6
7  void draw() {
8    background(255);
9    if(mouseX>x) {x=x+1;}
10   if(mouseX<x) {      ;}
11   line(x, 0, x, height);
12   ellipse(      ,      , 20, 20);
13 }
```

白い円が右側にあるとき、右に進むコード

白い円が左側にあるとき、左に進むコード

マウスに付いて動く白い円

発展的復習だよ

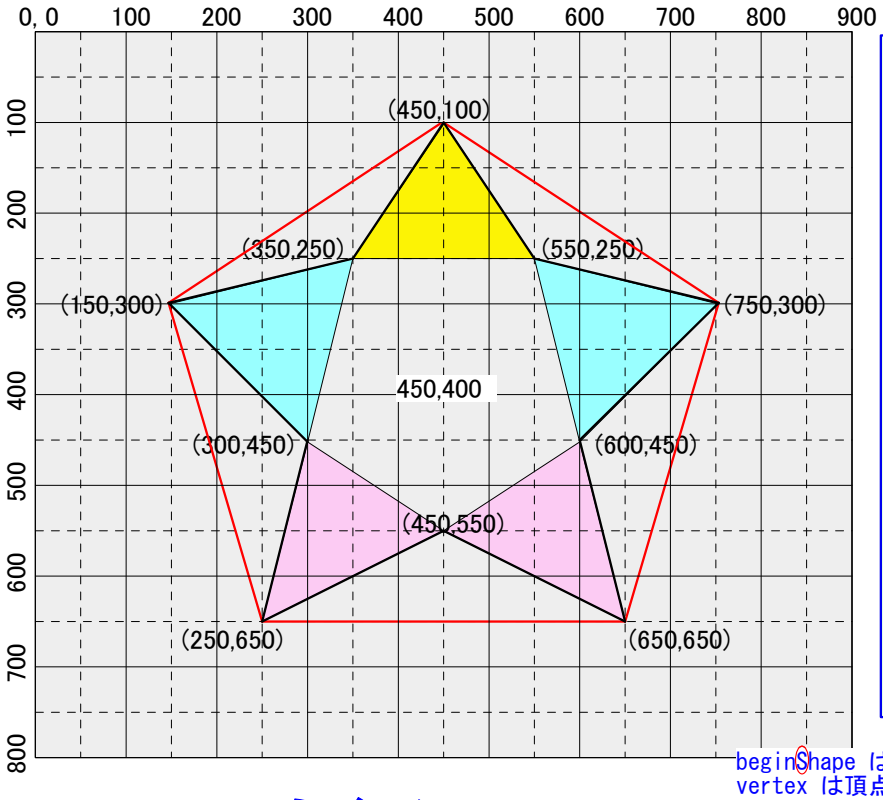
```
1  int x=350;
2
3  void setup() {
4    size(700, 500);
5  }
6
7  void draw() {
8    background(255);
9    if(mouseX>x) {x=x+1;}
10   else{x=x-1;}
11   line(x, 0, x, height);
12   ellipse(mouseX, mouseY, 20, 20);
13 }
```

10行目のif文を左のように書いても同じだよ。else はエルスと読んで、意味は if の条件が満たされないときは、{　}を実行せよ、となるんだ。

ここでは、mouseX>x ではない時、つまり、mouse<x の時に x=x-1; を実行せよということになるね。

７ーステップ　１：画像へ行く前に

ファイルから新規を開くんだけど、星形のコードは下の図を見て考えて欲しい。



新しい命令

この図形を描く命令はすでに学んだ。
1. lineを10本書いて線で囲む。
2. 色つきの部分の三角形を５つ描く。
どちらでも描けるね。
　しかし、それでは星形の内部の面積がない図形でしかない。第１回で確認したとおりだ。

多角形を描く命令が用意されている。

```
beginShape();
vertex(  ?  ,   ?   );//点の座標
vertex(  ?  ,   ?   );
vertex(  ?  ,   ?   );
      〃
      〃
vertex(  ?  ,   ?   );
vertex(  ?  ,   ?   );
endShape();
```

コピペだよ
座標値は後でね

() は大文字だよ

　これを使えば、複雑な多角形も内部面積をもった図形として描くことができるのだ。

beginShape は形を始める、endShape は形を終わる、
vertex は頂点とか尖った先という意味だ。

ミッション

　星形もしくは赤い線で囲まれた五角形のコードを書いてみよう。size はいくつかな？
次の行には、beginShape(); と書こう。そして、vertex(　　,　　); は上の図から読み取ってほしい。１０点もしくは５点が書けたらendShape(); で閉じて、実行してみよう。星形でも五角形でも好きなほうを選んでね。
　何かおかしくないかい？　最後が閉じてないね。閉じるには、最初に書いた vertex(　　,　　); をもう一度書くか、endShape(); を endShape(CLOSE); としてやるんだ。すると図形が閉じて面積を確定できるよ。

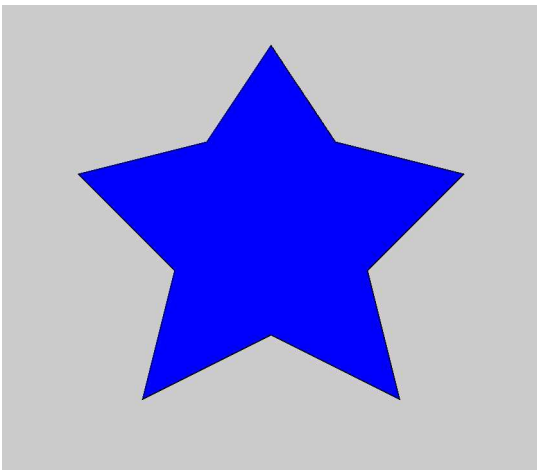
星形のコード

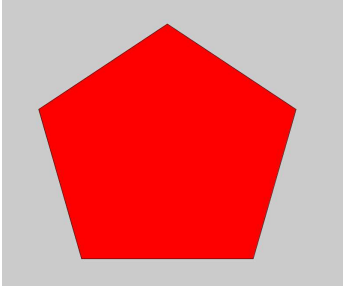
```
1  size(900,800);
2      コピペだよ！ 座標値は後からね
3  beginShape();
4  vertex(450,100);
5  vertex(550,250);
6  vertex(750,300);
7  vertex(600,450);
8  vertex(650,650);
9  vertex(450,550);
10 vertex(250,650);
11 vertex(300,450);
12 vertex(150,300);
13 vertex(350,250);
14 endShape(CLOSE);
15
```

五角形のコード

```
1  size(900,800);
2    コピペだよ！ 座標値は後からね
3  beginShape();
4  vertex(450,100);
5  vertex(750,300);
6  vertex(650,650);
7  vertex(250,650);
8  vertex(150,300);
9  vertex(450,100);
10 endShape();
11
```

()と(　)内は大文字だよ





　これで複雑な図形も描けるようになったね。線で描いたものとは違って、内部があるから色付けもできる。星形では beginShape(); の前に fill(0,0,255); を、五角形では fill(255,0,0); を書いただけだよ。重ねて描いたり、移動させたり他の図形とまったく同じように扱うことができるんだ。

star という名前で保存しよう。

7ーステップ　2：画像へ行く前に、Star の続き

複雑な星形のコードを書いたのだから、このコードを使い廻しできないだろうか。同じ図形をたくさん描くときには、for を使った。今回は、星形のコードを pushMatrix(); と popMatrix(); とで挟んでやれば1つの図形となって、それに名前を付けてセットとして扱えるようになるんだ。

```
15 void star(int x, int y){
16          ○ は大文字だよ
17  pushMatrix();
18   translate(x,y);
19   scale(0.1);
20   noStroke();
21   fill(0,0,255);
22   beginShape();
23     vertex(450,100);
24     vertex(550,250);
25     vertex(750,300);
26     vertex(600,450);
27     vertex(650,650);
28     vertex(450,550);
29     vertex(250,650);
30     vertex(300,450);
31     vertex(150,300);
32     vertex(350,250);
33   endShape(CLOSE);
34  popMatrix();
35 }
```

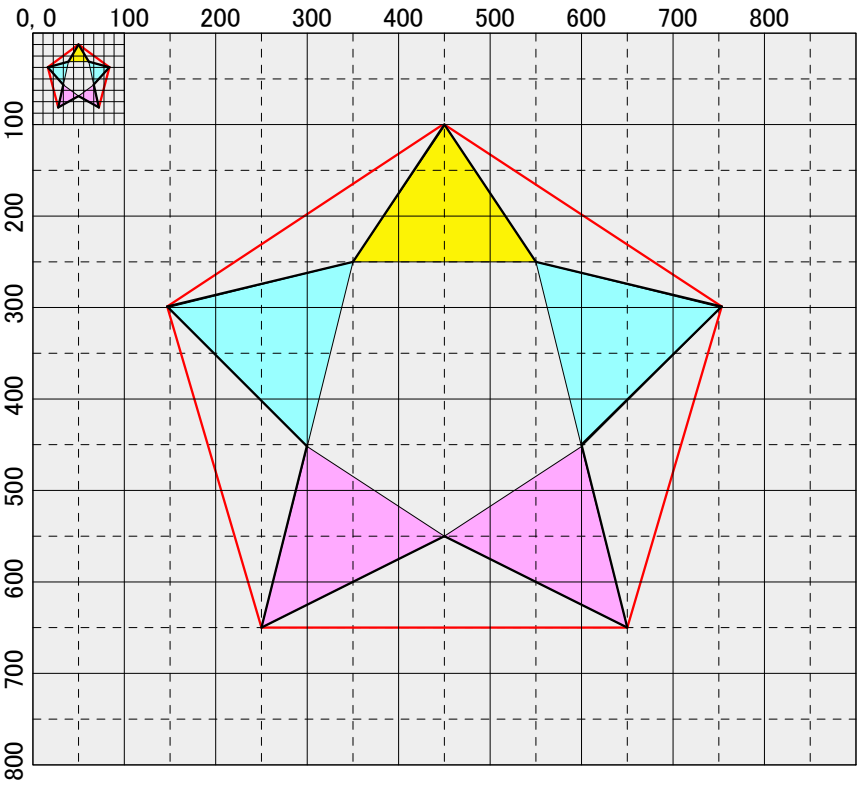
この間が、１セットとなる。

第5回ステップ２でやった座標系をずらす命令

座標系を 0.1 に縮小する命令。(　)の数字が小さいと小さく縮小される

この間が、星形のコード

pushMatrix(); は保持する
popMatrix(); は解放する
という意味かな。




```
1  void setup(){
2  size(900,800);
3  }
4
5  void draw(){
6    star(100,100);
7    star(200,300);
8    star(550,250);
9    star(400,500);
10   star(750,450);
11 }
12
13 void star(int x, int y){
14
15  pushMatrix();
16   translate(x,y);
17   scale(0.1);
18   noStroke();
19   fill(0,0,255);
20   beginShape();
21     vertex(450,100);
22     vertex(550,250);
23     vertex(750,300);
24     vertex(600,450);
25     vertex(650,650);
26     vertex(450,550);
27     vertex(250,650);
28     vertex(300,450);
29     vertex(150,300);
30     vertex(350,250);
31   endShape(CLOSE);
32  popMatrix();
33 }
```

自作の関数

最初に書いた星形は大きいので、使い回すには不便なんだ。そこで、scale(　);をつかって 0.1 を掛けて10%の大きさに縮めてしまったのが、上の図の(0, 0)と(100, 100)の間に描いてある星形だよ。

translate(　); は値を、変数 x,y にしておいて、使うときに位置を指定しているようにしたんだね。

この１セットに star って名前を付けたのが、void star(int x, int y); だ。この１セットを自分で作ったので、自作関数と呼ぶんだよ。

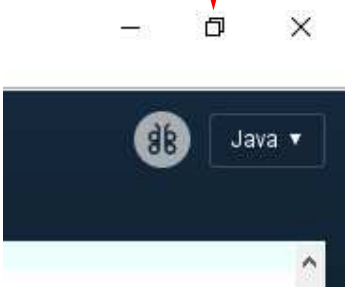
使い方は簡単。5～10行目を見て欲しい。star( ); って書いて、カッコの中に位置だけ指定すれば良いんだ。これで５つの青色の星が描けたよ。自作関数を使うと、たった１行で複雑な星が描けるようになったね。

star-1　という名前で保存しよう。

８ーステップ　３：画像を取り入れよう

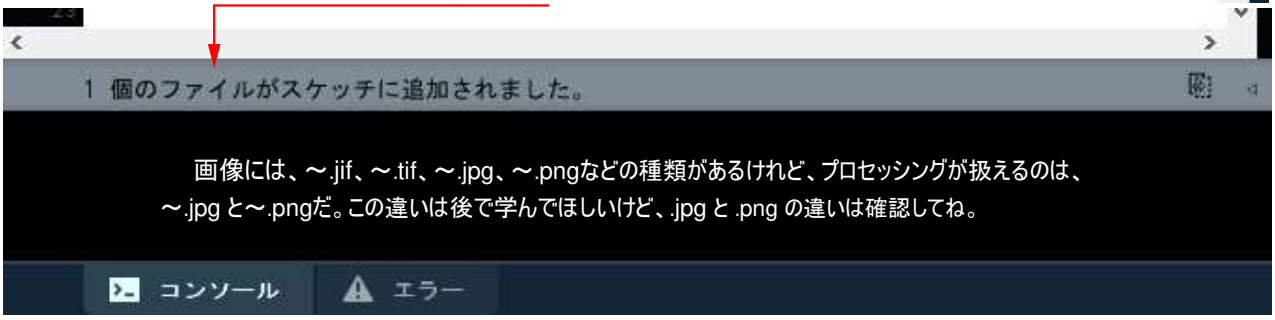
ファイルから新規を開いてみよう。

ここをクリックして画面を小さくする



新規を開いて全画面になっている場合は、右端上の二重重ねの小さな四角をクリックして、画面を小さくしてね。
デスクトップに DATA というフォルダーがあるから、それをダブル･クリックしてみよう。フォルダーの中には画像がたくさんあるので、好きなのを選んで画像の上で左クリックしたまま、コードを書く白いテキスト･エディターへと引きずって欲しい。そこでクリックを放すと、画像が取り込まれるんだ。　ではやってみよう。

ここに下の文がでれば、成功だ！



○ は大文字だよ

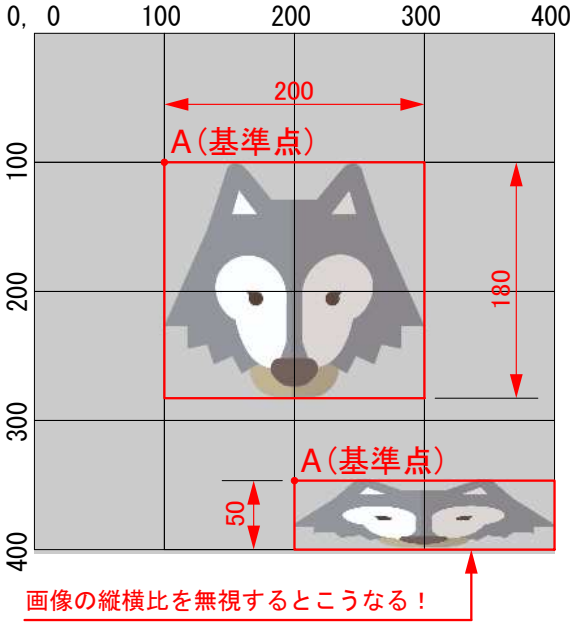
PImage wolf; と画像のための変数を宣言する

```
1 PImage wolf;
2
3 size(400, 400);
4 wolf = loadImage("wolf.png");
5 image(wolf, 100, 100, 200, 180);
```

変数 wolf に画像名を代入

.png を確認

画像の名前　画像の位置　画像の大きさ



画像の縦横比を無視するとこうなる！

PImage は画像を扱いますよと言う宣言で、P と I が大文字だ。続いて画像の名前を変数として宣言する。たくさん画像を扱う場合には、それぞれに名前を変える必要があるからね。
宣言した変数に、wolf.png を代入するんだけど、画像の場合は、特に loadImage(″　″); と言う形にするんだよ。
イメージを取り込むって意味かな。
取り込んだ画像を表示するのが、image( ); だ。wolf は画像の名前、次の２つの数字は画像の基準点のX座標とY座標を最後の２つの数字は画像の大きさで、横幅と縦の長さを表している。横幅と縦の長さの比率を無視すると、右下の図のようになってしまうよ。縦横比は次のステップで詳しく説明するね。
では、他の画像を取り込んだりして、数字を変化させて大きさや位置を変えて楽しんでみよう。

どこを変えたかな？　実行してみよう。

```
1 PImage wolf;
2
3 size(400, 400);
4 wolf = loadImage("wolf.png");
5 image(wolf, 50, 50, 300, 270);
```

.png を確認

画像にも色付けができる？　tint(r, g, b); だ。

```
1 PImage wolf;
2
3 size(400, 400);
4 wolf = loadImage("wolf.png");
5 tint(43,185,108);
6 image(wolf, 50, 75, 100, 100);
7 tint(185,43,88);
8 image(wolf, 200, 75, 150, 150);
```

fill じゃなくてtint だよ。

tint は染めるという意味だ。

どこを変えたかな？　実行してみよう。

```
1 PImage wolf;
2
3 size(400, 400);
4 imageMode(CENTER);
5 wolf = loadImage("wolf.png");
6 image(wolf, width / 2, height / 2, 100, 100);
```

○ は大文字だよ

画像の基準点を画像の中心にする

for 文を組み込んでみよう。

```
1 PImage wolf;
2
3 size(400, 400);
4 wolf = loadImage("wolf.png");
5 for(int a=1; a<400; a=a+100)
6 image(wolf, a, 75, 100, 100);
```

○ は付けてないけど、
どれが大文字か確認してね。

7ーステップ ４：画像を動かそう

ファイルから新規を開いてみよう。画像を自分で動かすことは出来ないだろうか？
そこで if 文と組み合わせてみるよ。

```
1  PImage   ufo1;
2  int x=0;   //ｘ方向に動かすための変数
3  int y=0;   //Ｙ方向に動かすための変数
4
5  void setup(){
6    size(800, 800);
7    imageMode(CENTER);
8  }
9
10 void draw(){
11   background(255);
12   ufo1 = loadImage("ufo1.png");
13   image(ufo1, x, y, 150, 100);
14 }
15
16 void keyPressed(){
17   if(keyCode ==LEFT){x=x-1;}
18   if(keyCode ==UP){y=y-1;}
19   if(keyCode ==RIGHT){x=x+1;}
20   if(keyCode ==DOWN){y=y+1;}
21 }
22
```

ufo1 という画像を使ってみるよ。

()は大文字だよ

基準点は画像の中心だね。

.png を確認

()は大文字だよ

コピペだよ、手直しはその後でね。

PImage は ufo1 という画像を扱いますよと宣言した。動かすのだから、変数が必要だね。しかも、縦横に動かすのは、X座標とY座標の両方向に動かすということだから、変数は x　y の２つ必要だ。

動かすためには、void setup() と void draw() の命令を使うんだったね。void setup() には最初の一度だけ設定する命令を、void draw() には繰り返し使う命令を書くんだよ。

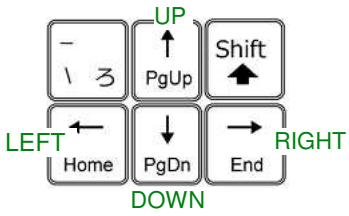
background(255); で背景を白にして、何度も表示させているから、次の行の ufo1.png が何度表示されても、そのたびに白で覆ってしまうから動いているんだ。

16行からのコードは下の説明を読んで欲しい。

画像の基準点は左上だけど、
imageMode(CENTER); を加えると、中心になるんだよ。

void は２つに限らず、16行目のような使い方もできるんだ。keyPressed は mousePressed と同じで、キイが押されたと言う意味だけれど、キーはたくさんあるから if 文を使って命令を細かく指定してやるんだね。

17行目は、もし、keyCode が LEFT なら x に x-1 を代入しなさい、というものだから、画像は左へと動いていくわけだ。右の図はキーボード右下を切り取ったんだけど、上のコードを実行したらキーを押してみて欲しい。



ミッション

矢印に従って動くけど、メチャクチャ遅いね。
下のヒントを使って、早く動くようになおしてみよう。

ヒント

速さを制御しているのは、x=x-1 の -1 だった。この数字を大きくすれば早くなった。でも４つもあるから、一つ一つ変更するのは大変だ。しかも、変更した数字が最適とは限らないから、また４つ直さなければならない。そこで、速さの定数 int go; を作ろう。int go=10; で試してみよう。上のコードの 17～20行目までの 1 は go に変えるんだよ。

早く変えたコード

```
1  PImage   ufo1;
2  int x=0;
3  int y=0;
4  int go=10;
5
6  void setup(){
7    size(800, 800);
8    imageMode(CENTER);
9    x =width/2;
10   y =height-900;
11 }
12
13 void draw(){
14   background(255,72,139);
15   ufo1 = loadImage("ufo1.png");
16   image(ufo1, x, y, 150, 100);
17 }
18
19 void keyPressed(){
20   if(keyCode ==LEFT){x=x-go;}
21   if(keyCode ==UP){y=y-go;}
22   if(keyCode ==RIGHT){x=x+go;}
23   if(keyCode ==DOWN){y=y+go;}
24 }
```

加えたコード

実行されたときにUFOのいる座標

.png を確認

()は大文字だよ

コピペだよ、手直しはその後でね。

実行すると、ピンクのウィンドウしか出てこないけど、下向きの ↓ を押すと、UFO が降りてくるよ。

ufo-1 で保存しよう。

7ーステップ ５：写真を取り込んでみよう

ファイルから新規を開いてみよう。画像と同じように写真も取り込むことができる。
ふつう写真は背景に取り込むよね。そこで大きさや縦横比が大きな影響を与えるんだ。

① DATA のホルダーをダブル･クリックして開くよ。たくさんある画像や写真の中から、使いたい写真の上にマウスを置くと、左のようにファイルの種類や大きさが表示されるんだ。p700.jpg は 800×600 だね。これを書き留めて、コードの中に書くんだよ。
　写真を左クリックしたまま引きずってくるのは、画像と同じだよ。

② 写真を背景に使う場合は、もう１つ注意する点がある。背景はウィンドウ全体に広がっていないとおかしいよね。
　背景のサイズと、size(,); の数字をそろえるないと、左のような状態になってしまうんだ。p700.jpg の写真を背景に使うなら、size(800,600); にすると完璧な納まりになって、キレイだよ。写真の基準点は、左上だから写真の位置は 0, 0, だ。
　ではやってみよう。

```
PImage p700;

size(800, 600);                    .jpg を確認
p700 = loadImage("p700.jpg");
image(p700, 0,0,800,600);
```

キレイなウィンドウが現れたと思う。でも、手抜きの方法もあるんだ。写真のサイズと size をそろえた場合は、下のように、写真の大きさを書かないんだ。
　ただし、写真のサイズと size が違う場合は、写真のサイズを調整しないと右上のようになってしまうからね。

```
PImage p700;

size(800, 600);                    .jpg を確認
p700 = loadImage("p700.jpg");
image(p700, 0,0);
```

画像の縦横比の話し＝相似(そうじ)ということ

　画像も、縦横比を保ったまま拡大･縮小しないと歪んでしまうね。例えば、wolf.png は 513*462 だった。これを 200*50 の大きさにしたら、平べったくなってしまったね。
　そこで横幅が 200 になるようにしたかったので、200*( ) になるように計算したら、約180 だった。だから、大きさを200,180としたんだ。
　計算方法は、513*462=200*( ) だから、左辺の 462/513 の割算をして、右辺の 200 を掛ければ良いんだ。下の図を見ながら、画像や写真の大きさを計画して欲しい。

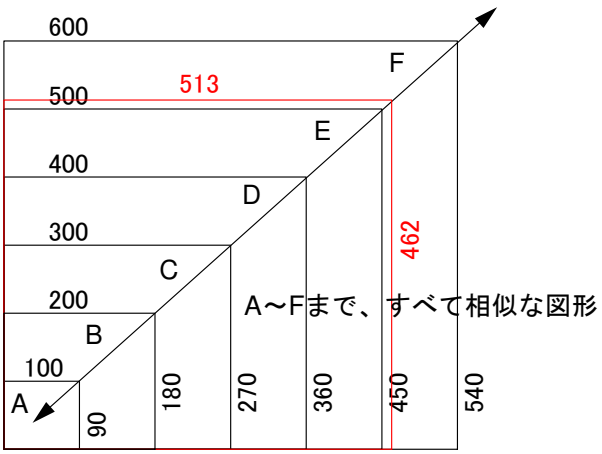


foto と名付けて保存しよう。

　画像も写真も横幅と縦の長さに注意して扱わないと、おかしなことなってしまうよ。
　縦横比が等しい図形のことを＜相似＞といって、左の図のような関係があるんだ。図形が大きくなっても小さくなっても、横／縦または縦／横の値が一定なのだ。
　この図だと、600／540 も 100／90 も1.11 だし、513／462 も 1.11 だね。 540／600 も 360／400 も 0.9 だし、462／513 も 0.9 だね。

背景にキレイな写真が入ったので、
次にはこの背景に画像を加えてみよう。

７ーステップ　６：写真の背景に画像をいれてみよう

foto の続きだよ。

```
PImage  p700;

size(800, 600);
p700 = loadImage("p700.jpg");
image(p700, 0,0);
```

やってみよう。

海の背景に鯨を入れてみよう。DATA から whale.png をウィンドウへドラッグしてね。ドラッグとは左クリックしたまま右や左に引きずることだよ。「１個のファイルがスケッチに追加されました。」ってでた？
次に、２行目に PImage　whale; って書こう。
これで鯨を読み込む準備ができた。そして、鯨の読み込みだ。
一番下に、whale = 読み込む(" 鯨.png "); と
image(鯨, X座標, Y座標, 鯨の横方向の大きさ, 鯨の縦方向の大きさ);
を書き加えてみよう。日本語はコードにしてね。これで鯨が海にでたぞ。

鯨を左右に泳がせてみよう。左右に移動させるには、変数が２つ必要だね。
PImage whale; の下に int a = 1;　int b = 1; と書き加えよう。そして、size(800,600);をvoid setup(){　}ではさみ、whale = 読み込む(" 鯨.png "); から最後の行までを、void draw(){　　} で包んでやる。
一番最後の } の前に、a=a+b;　if (a >= 400) b = -1;　　if (a <= 0) b = 1; を書き加えて、image( ); 内のX座標を a に書き換えれば、鯨は泳ぎ始めるぞ。

できたら whale で保存しよう。

ミッション

①と③で新規のファイルだよ。写真のサイズとsize をそろえるのを忘れないでね！

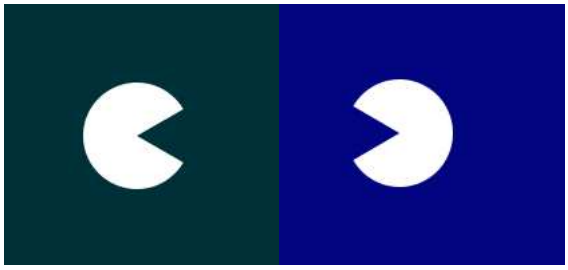
① 星空(p102.png)を背景にして、画面中央下部にロケットを描いてみよう。できたら rocket-1 で保存。
② ロケットを上昇させてみよう。できたら rocket-2 で保存しよう。
③ 同じ星空を背景にして、月(moon.png)を横切らせてみよう。できたら moon-1 で保存。
④ 同じ星空を背景にして、月を左下から登らせて、真ん中まで来たら、右下へと沈んで行かせよう。
できたら moon-2 で保存しよう。if 文を使うのがヒントだ。

画像や写真は下記のサイトから借用させて頂きました。
FLAT ICON DESIGN　http://flat-icon-design.com/
photoAC　https://www.photo-ac.com/

次回の予告

円弧を知っているか？ 円の一部を円弧と言うんだね。半円は中心角が180°の円弧だし、上弦の月も下弦の月も円弧だよ。次回は、円弧の描き方から、方向変換などをやろう。

行き　帰り



画像や図形の往復はできるようになったね。今までは行ったままの姿勢で戻ってきていた。今度はキチンと進行方向を向いて往復させたいね。

Bye-bye!