

Processing

第8回

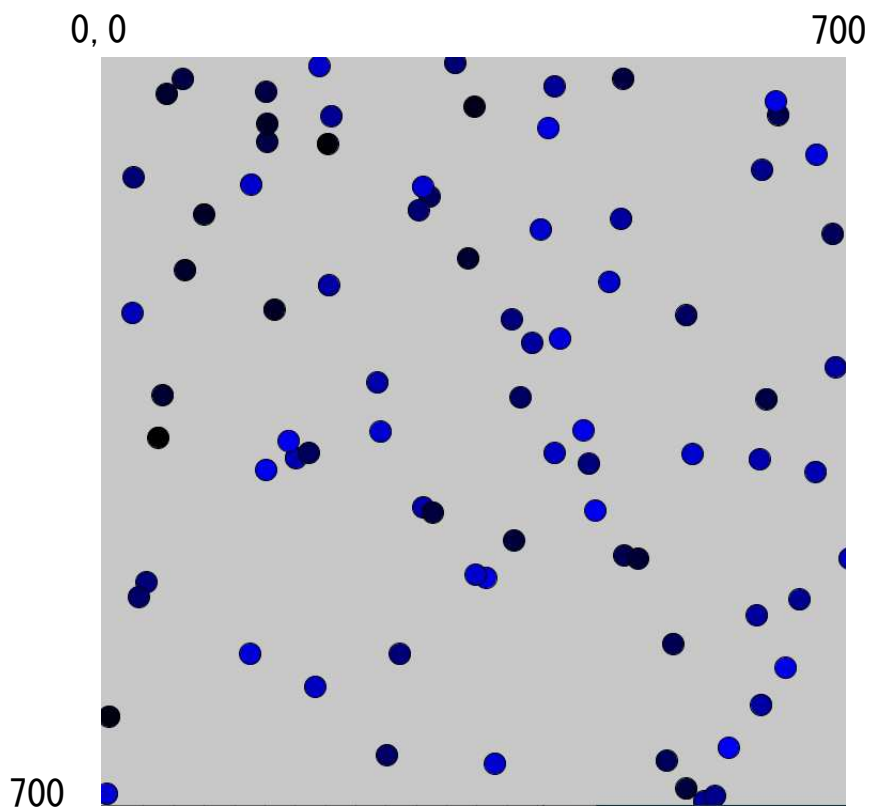


松田小学校／寄小学校

8 ステップ 0 : 前回の復習だよ

新規のファイルを開いて、下のコードを打って実行してみよう。

```
1 void setup() {  
2   size(700, 700);  
3   frameRate(10); ← 実行回数のコントロールで、1秒間に10回を表す  
4 }  
5  
6 void draw() {  
7   background(200);  
8   for(float a=100; a<700; a=a+0.5) {  
9     fill(0, 0, random(256));  
10    ellipse(random(10)*a, random(10)*a, 20, 20);  
11  }}
```



ミッション

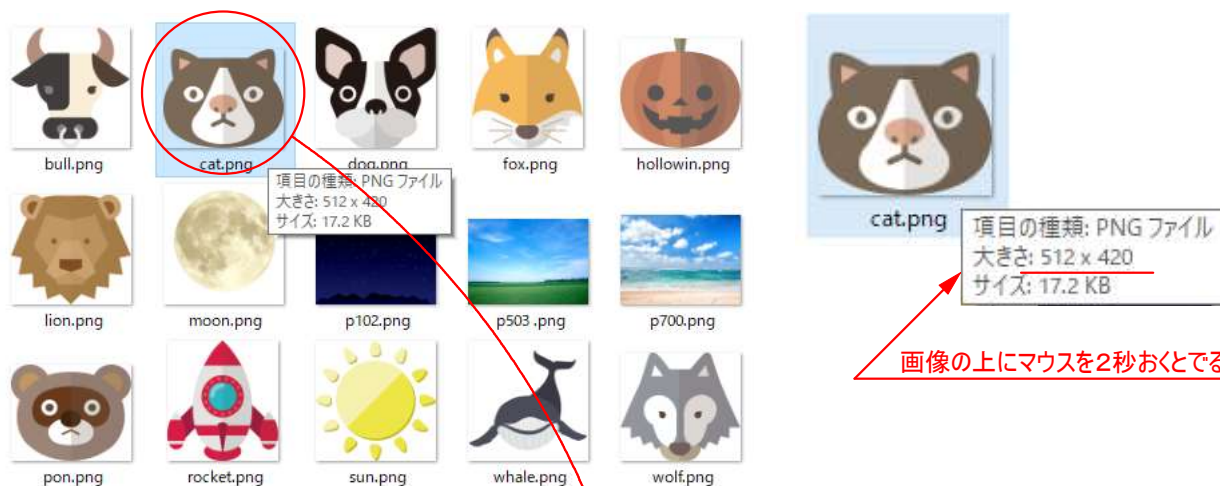
① 丸の色や大きさを変えてみよう。

保存は不要だよ。できたらステップ1に進もう。

8 - ステップ 1 : 画像 (image=イメージ) を取り入れよう

画像の取り入れ方

- ① デスクトップにあるDATA(データ)ファイルを、ダブルクリックして開く。
- ② 取り入れる画像を選んで、マウスでプロセッシングに引きずってくる。
- ③ 「1個のファイルがスケッチに追加されました」と出たら、成功だよ。



著作権について

絵やカット、写真、音楽などの作品を創作した者がもつ権利を著作権といいます。お金をかせぐことを目的として、著作権者に無断で作品を使うことはできません。ただし、著作権者が権利を放棄している場合は無断で使ってもかまいません。

8 - ステップ 2 : 画像 (image=イメージ) を表そう

画像を表すための準備

画像を扱うには、PImage (ピーイメージ) という変数が必要だ。

Picture Image (ピクチャー イメージ)

画像を読み込むには、画像名=loadImage ("画像名.png");

load Image(ロードイメージ)

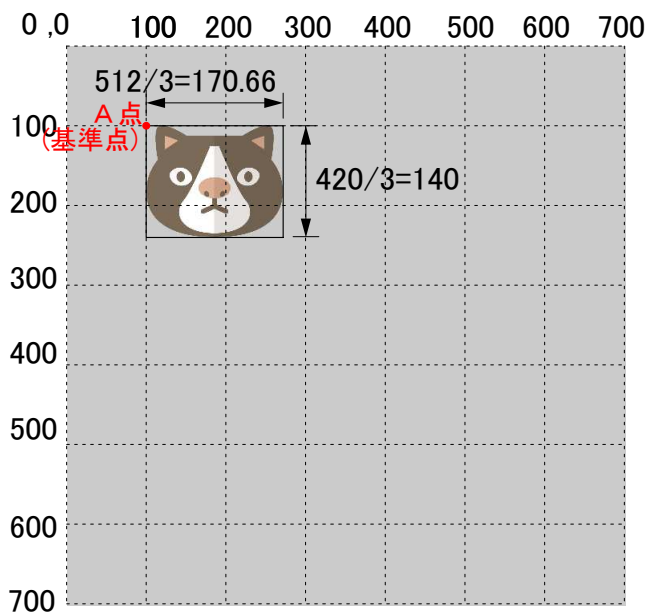
画像を表示するには、image (画像名、X座標、Y座標、横幅、縦の高さ)

```
1 PImage cat; ← cat という変数を使うと宣言する
2
3 size(700, 700); ← () は大文字だよ
4 cat = loadImage("cat.png"); ← 変数 cat に、画像 cat.png を読み込む
5 image(cat, 100, 100, 512/3, 420/3);
```

A 点の X 座標

A 点の Y 座標

大きすぎるので 1/3 にした。



cat.png
項目の種類: PNG ファイル
大きさ: 512 x 420
サイズ: 17.2 KB

ミッション

- ① 5行目の /3 をとって、実行してみよう。
- ② 画像の大きさを変えてみよう。/3 を、/2 とか /5 にすると変わるよ。
- ③ /3 に戻して、画像の位置を変えてみよう。中央に位置できるかな？
- ④ 他の画像も取り入れてみよう。2行目に PImage なまえ の宣言を追加してね。次に 4、5行目を 6、7行目にコピーして、取り入れる名前に変えよう。

8 - ステップ 3 : 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。



dog.png



fox.png



halloween.png



p102.png



p503.png



p700.png



sun.png



whale.png

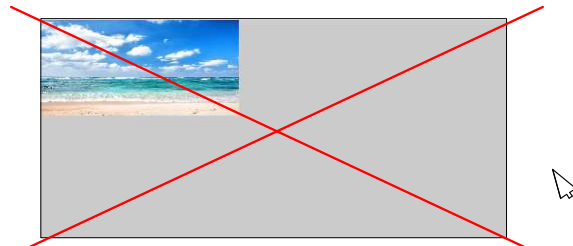


wolf.png

項目の種類: PNG ファ

大きさ: 800 x 600

サイズ: 66.4 KB



ウィンドウと写真の大きさをそろえないと、
上のようになってしまうよ。

```
1 PImage p700;  
2  
3 size(800, 600);  
4 p700=loadImage("p700.png");  
5 image(p700, 0, 0, 800, 600);
```

ウィンドウの大きさにそろえる。

```
1 PImage p700;  
2  
3 size(800, 600);  
4 p700=loadImage("p700.png");  
5 image(p700, 0, 0);
```

size(); と写真の大きさが同じ時は書かなくてもOK。

size(); の数字と写真の大きさが同じ時は、大きさを書かなくても、プロセッシングが自動的に判断して、ウィンドウ一杯に表してくれる。だから写真を背景とする場合は大きさを書かない方が良くもね。



できれば foto で保存しよう。

8 ステップ 4 : 写真の背景に画像をいれてみよう

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。
whale(くじら)を、テキストエリアに引きずって(ドラッグ&ドロップという)こよう。

```
1 Image p700, whale;
2 int a, b=1;
3
4 void setup() {
5   size(800, 600);
6   p700=loadImage("p700.png");
7   whale=loadImage("whale.png");
8 }
9
10 void draw() {
11   image(p700, 0, 0);
12   image(whale, a, 250, 512/3, 512/3);
13
14   a=a+b;
15   if( a<=0 ) { b=1;}
16   if(a>width-512/3) { };
17 }
```

背景にp700、鯨に whale という変数をつかう宣言

○ は大文字だよ

背景を読み込む

鯨を読み込む

鯨のX座標に変数 a をつかう

鯨が大きすぎるので、1/3にした。

右に行く

左に行く

鯨が左右に動く if 条件文



できたら whale で保存しよう。

8 ステップ 5 : チャレンジだ、写真と画像の続き

新規のファイルを開いて、右の図になるコードを書こう。

画像や写真はDATAファイルから引きずってくるんだ。

```
1 PImage p102, rocket;
2 [ ] ← int a = 最初のロケットの位置;
3
4 void setup() {
5   size(800, 600);
6   p102 = loadImage("p102.png");
7   rocket = [ ]("rocket.png");
8 }
9
10 void draw() {
11   image(p102, 0, 0);
12   image([ ], 400, a, 100, 200);
13   a=a-1;
14 }
```

0,0 800



600

ミッション

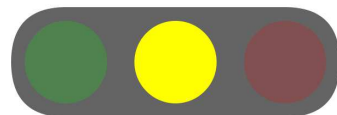
- ① ロケットを上昇させてみよう。2行目に動かすための変数 `a` を宣言してから、`void setup()`; と `void draw()`; を使うんだよ。できたら `rocket` で保存しよう。
- ② 月を追加して、月 (`moon.png`) を横切らせよう。ロケットと月が交差するかな？ `moon` で保存しよう。

画像や写真は下記のサイトから借用しました。

FLAT ICON DESIGN <http://flat-icon-design.com/>
photoAC <https://www.photo-ac.com/>

次回の予告

信号機を描いてみよう。
緑色→黄色→赤色と点灯するように、プログラミングしてみよう。そして、自分だけのオリジナル信号機にしてみよう。



Bye-bye!

伴走の方へ 第8回

① 図形を描く命令は、三角形までしかやりませんでした。四角形は `quad`(4 頂点の座標値); で表示できます。しかし、頂点を4つも書くのは、大変なわりに面白さに欠けるので、本ガイドでは扱いませんでした。

② 一連の処理をまとめて名前を付けたものを関数とよび、プロセッシングで使う命令はすべて関数と呼んでも過言ではありません。プロセッシングでは予め多くの関数を用意しています。たとえば、`ellipse` は円を描くという関数で、どんな円を描くかは毎回違いうだろうから、()内にその時に必要な数値や文字をコードを書く人が記入します。

③ `star(int x, int y);` という関数を作りましたが、この関数が実行されても星を描くだけで、関数の結果が他の式に代入されることはありません。つまりこの関数は返り値をもっていません。返り値をもっていない場合には、`void` で導き `void star(int x, int y);` と記述します。

④ `beginShape`、`vertex`、`endShape` は関数化して使うと、とても便利です。C言語やJAVAなど他の言語では、関数の学習が必須のようです。しかし、関数を使わなくてもコードは書けるので、本ガイドでは自作関数を1つしか扱いませんでした。そうは言っても、関数が使えるようになると便利なことは間違いないです。

⑤ 自作関数は作ってしまえば、加工して使うことができます。自作関数の中に指定しても良いし、使う場所で個々別々に指定することもできます。

⑥ 画像や写真の引用は、コードの文字が一字でも間違えると、画像や写真が表れないだけではありません。何も表れずにフリーズ気味にエラーが出ます。半ばフリーズ状態にはなりますが、実行をキャンセルすれば元に戻ります。改めて文字の間違いを探して下さい。`.jpg` か `.png` かの違いもエラーになります。

⑦ 画像や写真は楽しい素材なので、その取り扱いにはぜひ習熟したいものです。画像の基準点は、デフォルトで左上の隅です。`imageMode(CENTER);` と書けば、基準点を画像の中心に変更することができます。また、画像や写真は素材が大きさを持っているから、特殊な効果を狙うのであれば、相似形で用いることに注意して下さい。

ステップ5-①

```
1 PImage p102, rocket;
2 int a=500;
3
4 void setup() {
5     size(800, 600);
6     p102 = loadImage("p102.png");
7     rocket = loadImage("rocket.png");
8 }
9
10 void draw() {
11     image(p102, 0, 0);
12     image(rocket, 400, a, 100, 200);
13     a=a-1;
14 }
```

ステップ5-②

```
1 PImage p102, rocket, moon;
2 int a=500; //ロケットの出発点
3 float b, c=0; //月の出発点
4
5 void setup() {
6     size(800, 600);
7     p102 = loadImage("p102.png");
8     rocket = loadImage("rocket.png");
9     moon =loadImage("moon.png");
10 }
11
12 void draw() {
13     image(p102, 0, 0);
14     image(moon, b, c+200, 100, 100); //月を後ろに
15     b=b+0.3;
16     c=c-0.1;
17
18     image(rocket, 400, a, 100, 200);
19     a=a-1;
20 }
```

×座標とY座標の数字を変えて、
月を右上に少し昇らせている。