プログラミング講座(9)：Processingその３アニメーションを作ろう

今日のゴール

Processingを使ってキャラクターをアニメーションさせます

setup()とdraw()

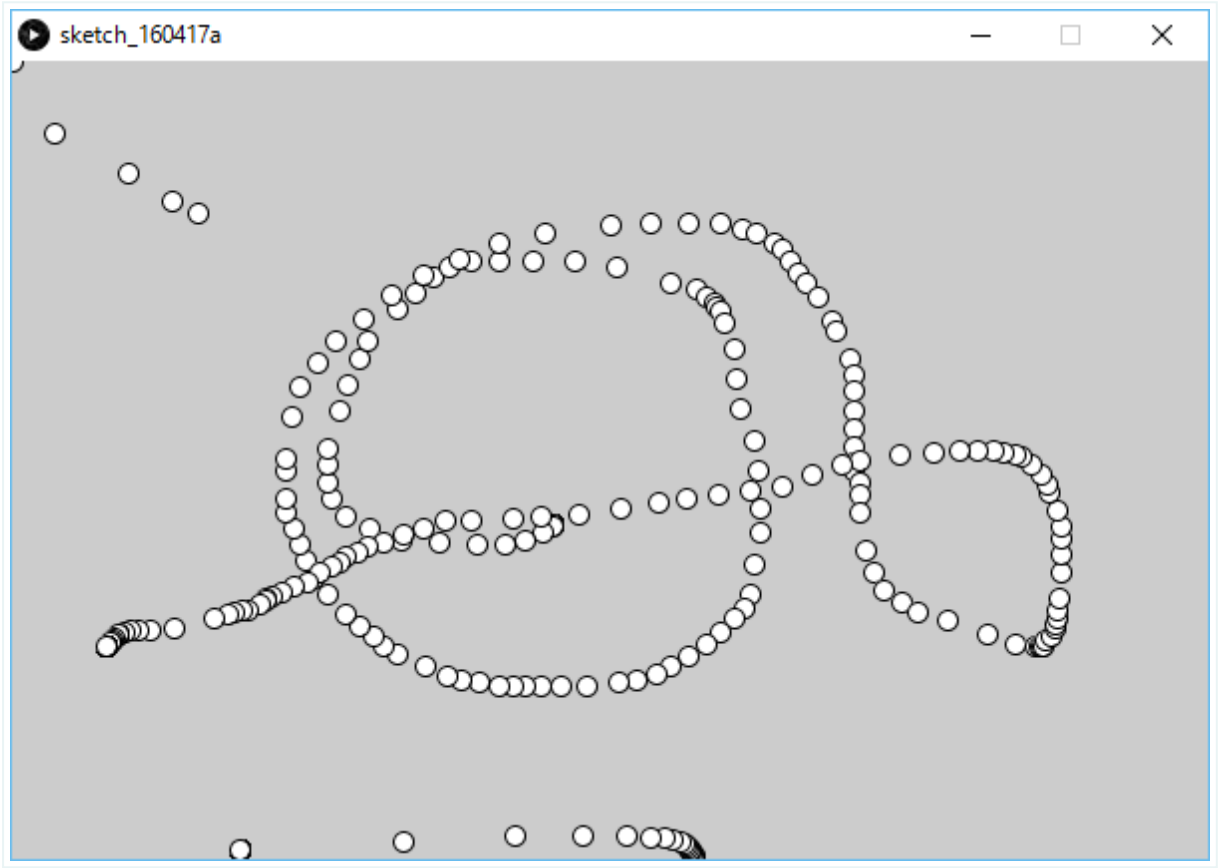
Processingでは、setupとdrawという関数（かんすう）を使うと、動きアニメーションを作ることができます。最初に一回だけ実行されるプログラムを**setup（セットアップ）**に、毎回繰り返されるプログラムを**draw（ドロー）**に書きます。

マウスポインターのところに円を描く

```
void setup(){
    size(600,400);
}

void draw(){
    ellipse(mouseX,mouseY,10,10);
}
```

マウスポインターを動かした場所に円が描かれます。



setup部分

```
void setup(){
    size(600,400);
}
```

この部分では、画面のサイズを決めています。サイズぎめは最初に一回だけ行えば良いので、**setup関数（かんすう）**の中に入れています。**void（ボイド）**というのは、なにかデータが帰ってきたりしないという意味です。

draw部分

```
void draw(){
    ellipse(mouseX,mouseY,10,10);
}
```

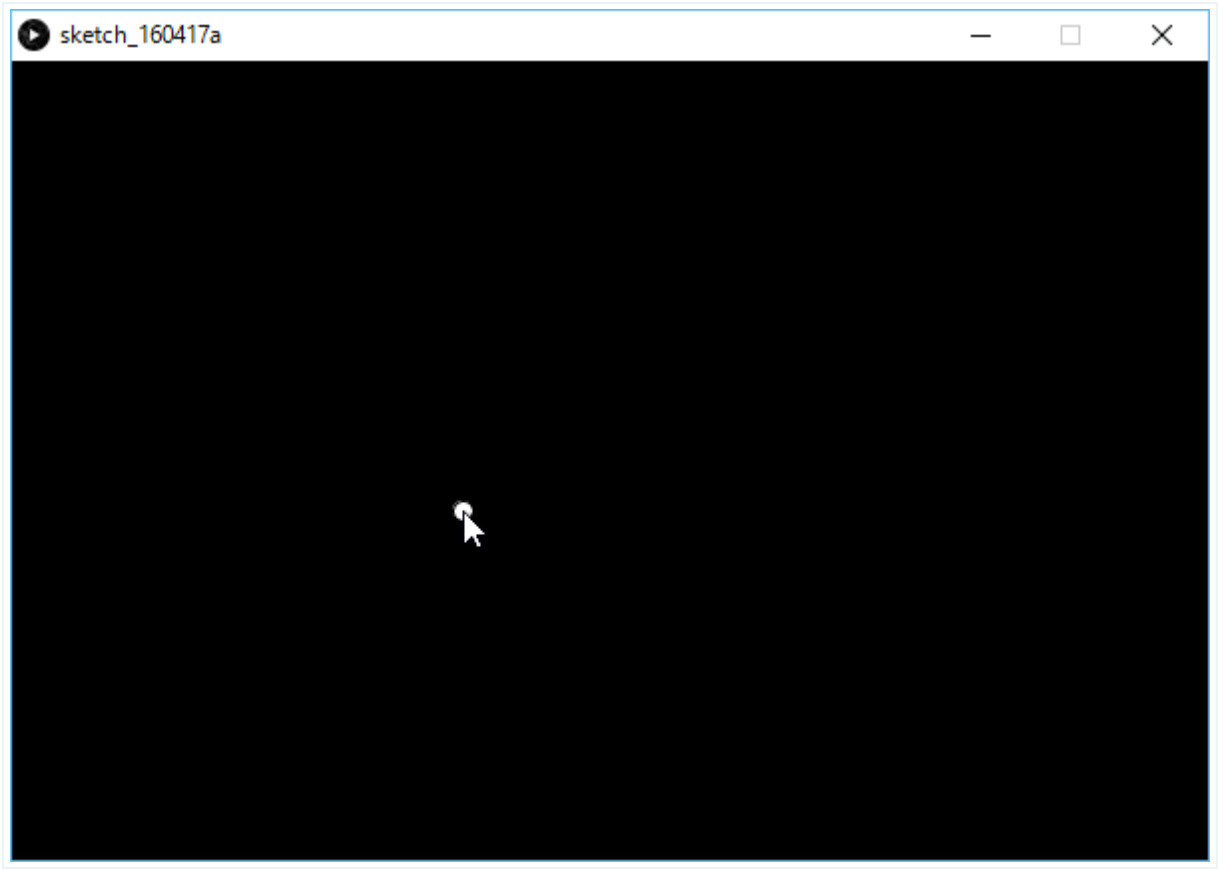
この部分では、**draw関数（かんすう）**の中に毎回実行されるコードを書いています。このコードは１秒間に60回実行されます。**mouseX（マウスエックス）**はマウスポインターのX座標、**mouseY（マウスワイ）**はマウスポインターのY座標です。
マウスの座標を中心として直径が10の円が毎回描かれます。

マウスポインターについてくる円

```
void setup(){
    size(600,400);
}

void draw(){
    background(0);
    ellipse(mouseX,mouseY,10,10);
}
```

マウスポインターのあとを円がついてきます。



draw関数部分

```
void draw(){
    background(0,0,0);
    ellipse(mouseX,mouseY,10,10);
}
```

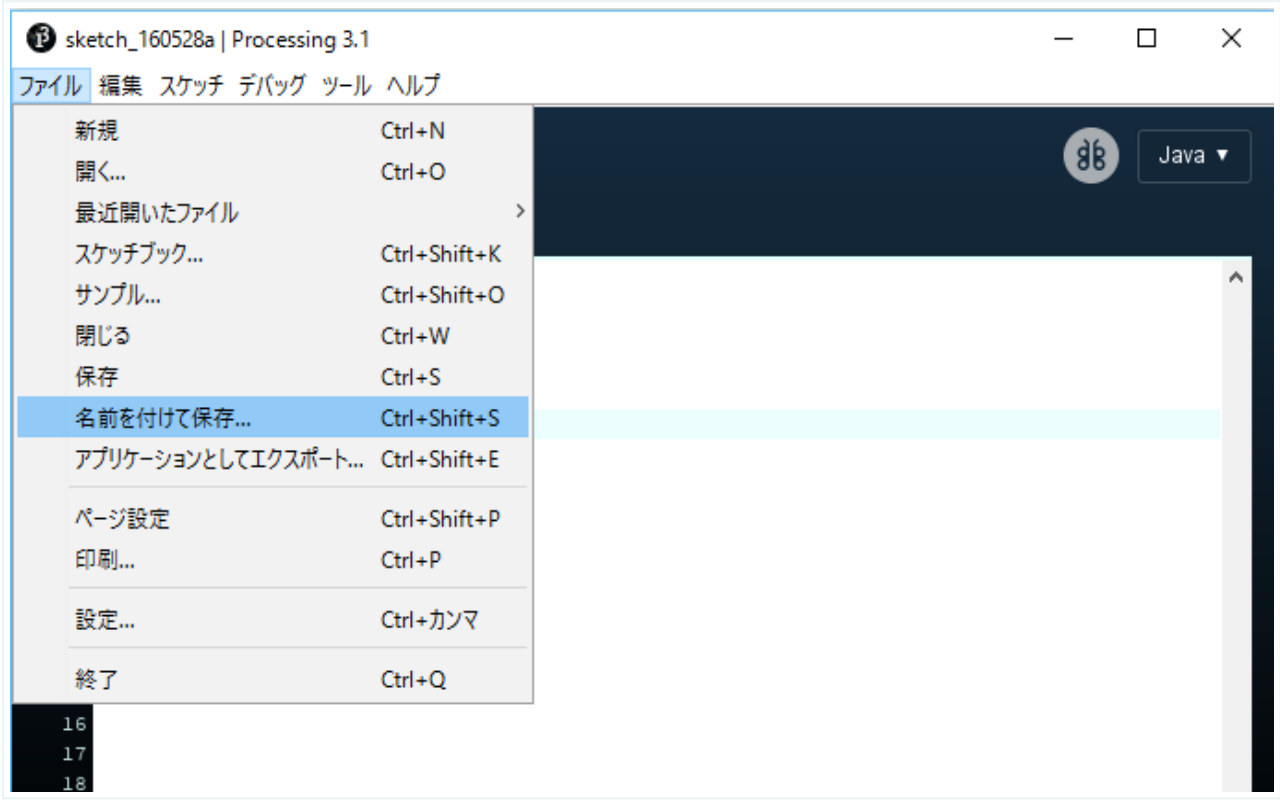
今回は、円を描く前に`background(0,0,0)`という命令を実行しています。これは、**全体を黒く塗りつぶす**という命令です。このために、今までに表示されていた円が塗りつぶされて、新しい円だけが表示されるので、マウスに円がついてくるように見えます。

プログラムを保存しよう

画像を表示するには、まずプログラムを保存する必要があります。

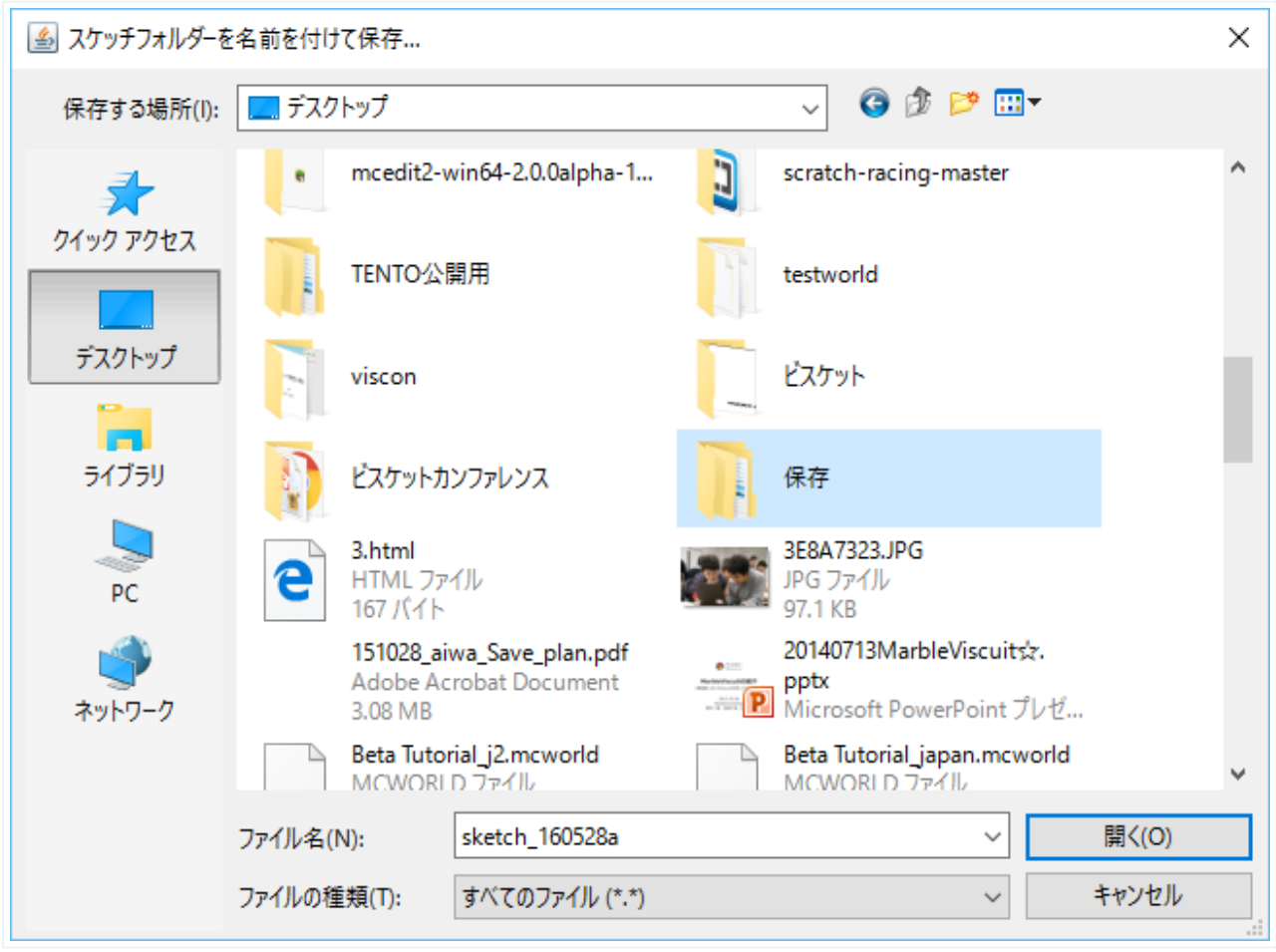
「名前をつけて保存」を実行

[ファイル]メニューから[名前をつけて保存]を選びます。



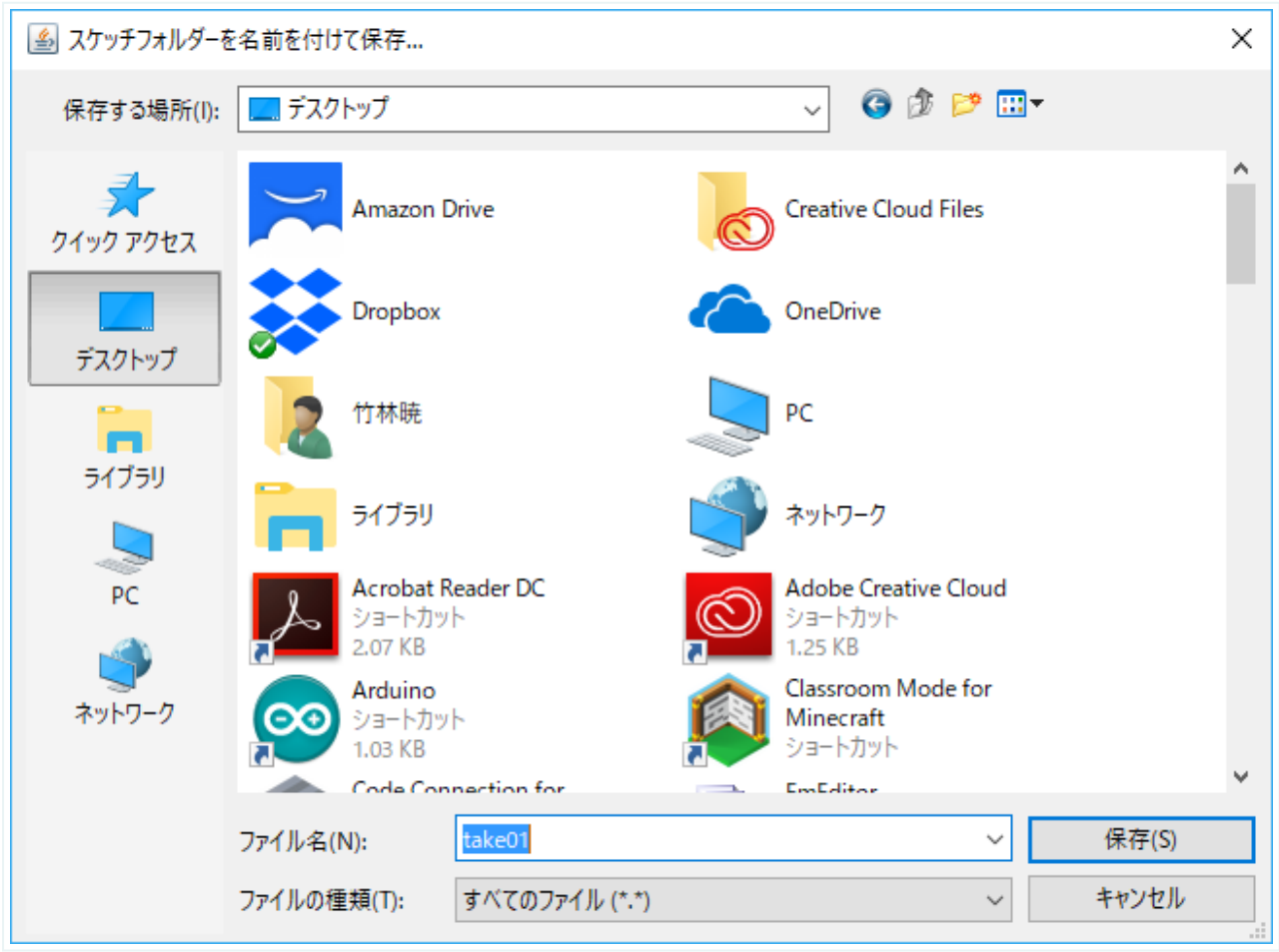
「デスクトップ」に保存する

[デスクトップ]を開きます。



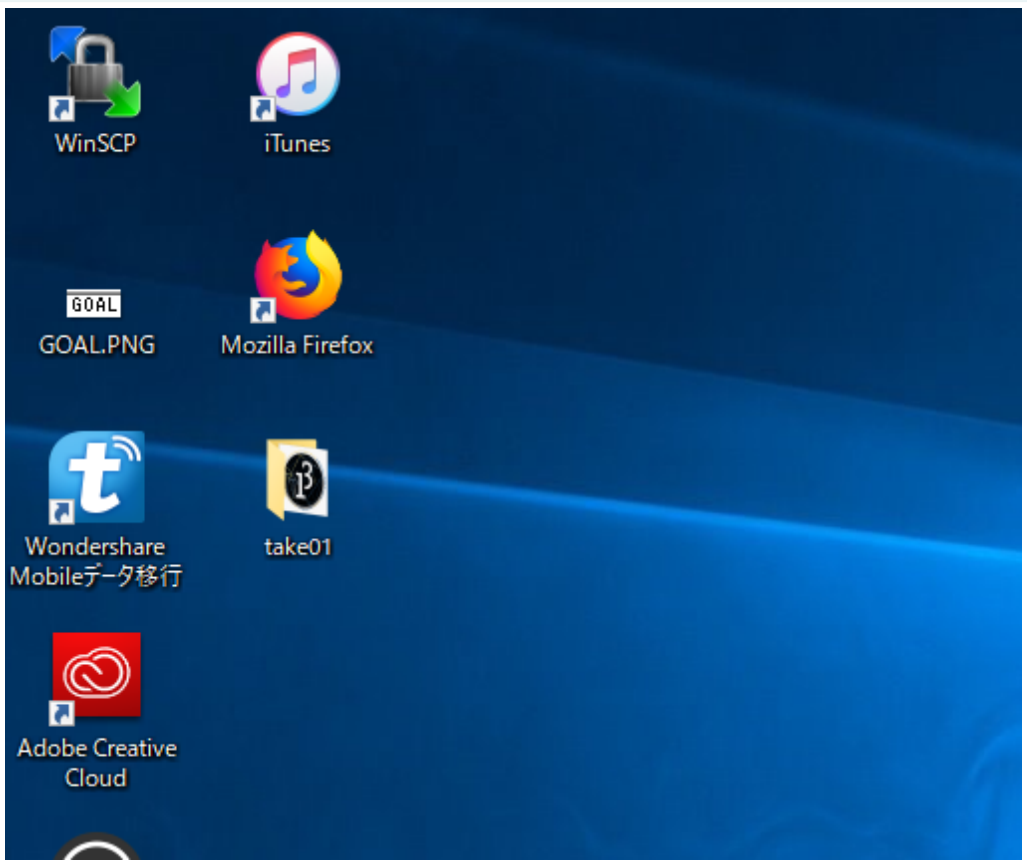
英語の名前で保存する

Processingは日本語のファイル名を使えないので、[ファイル名]は英語でつけましょう。今回は「take01」という名前で保存しています。



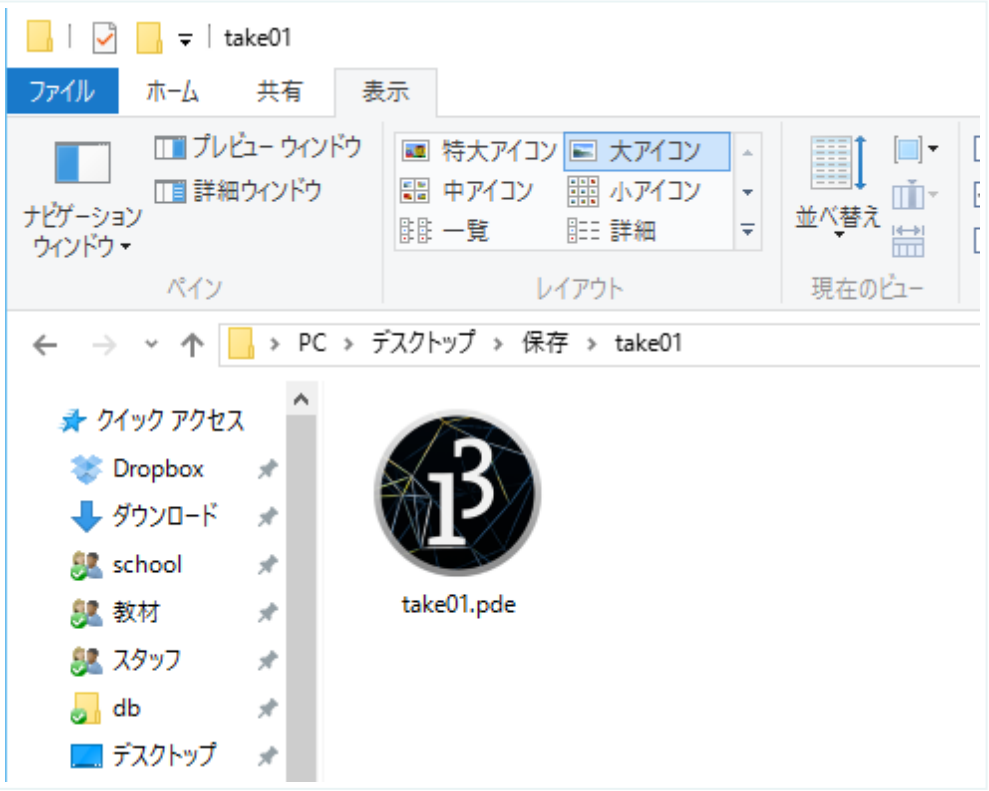
保存されたもの

保存すると、[take01]というフォルダができます。



プログラム本体

フォルダを開くと、その中に**take01.pde**というファイルがあります。これがプログラム本体です。

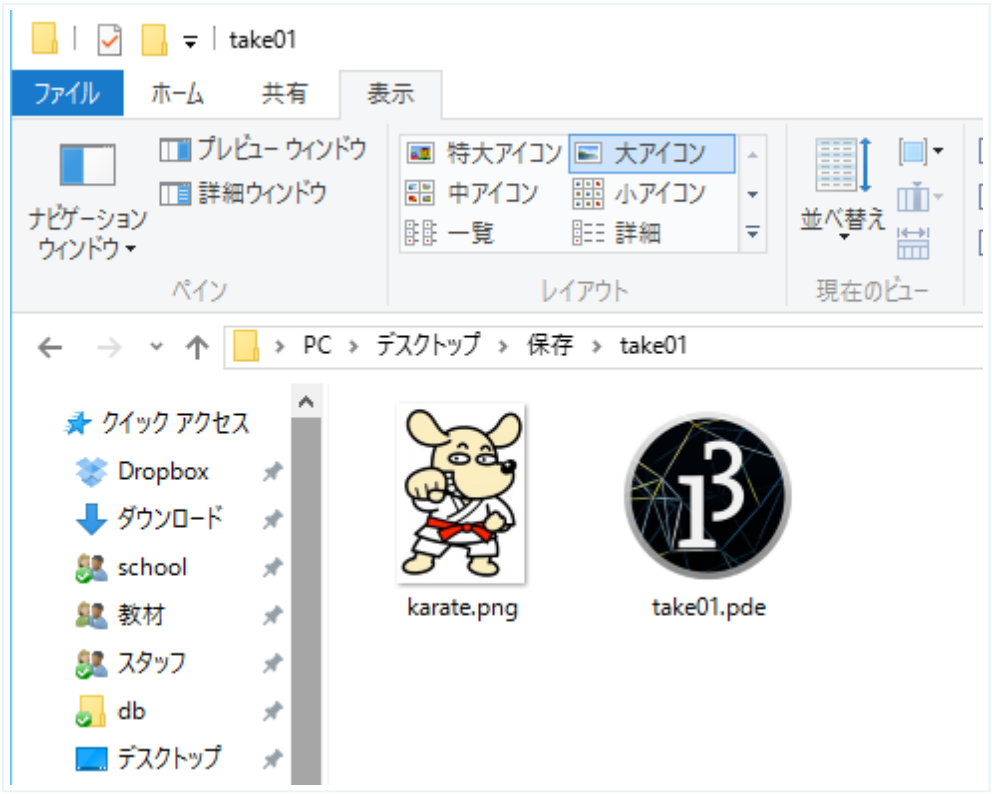


画像を表示する

画像は、適当なファイルを用意してください。ただし、発表用のプログラムの場合は、自分で作ったものか、著作権フリーの物を使いましょう。
今日はテントくんの画像を使います。

画像の置き場所

Processingで画像を表示したい場合、画像はプログラム本体のフォルダに置きます。ここでは、**karate.png**というファイルを用意しています。

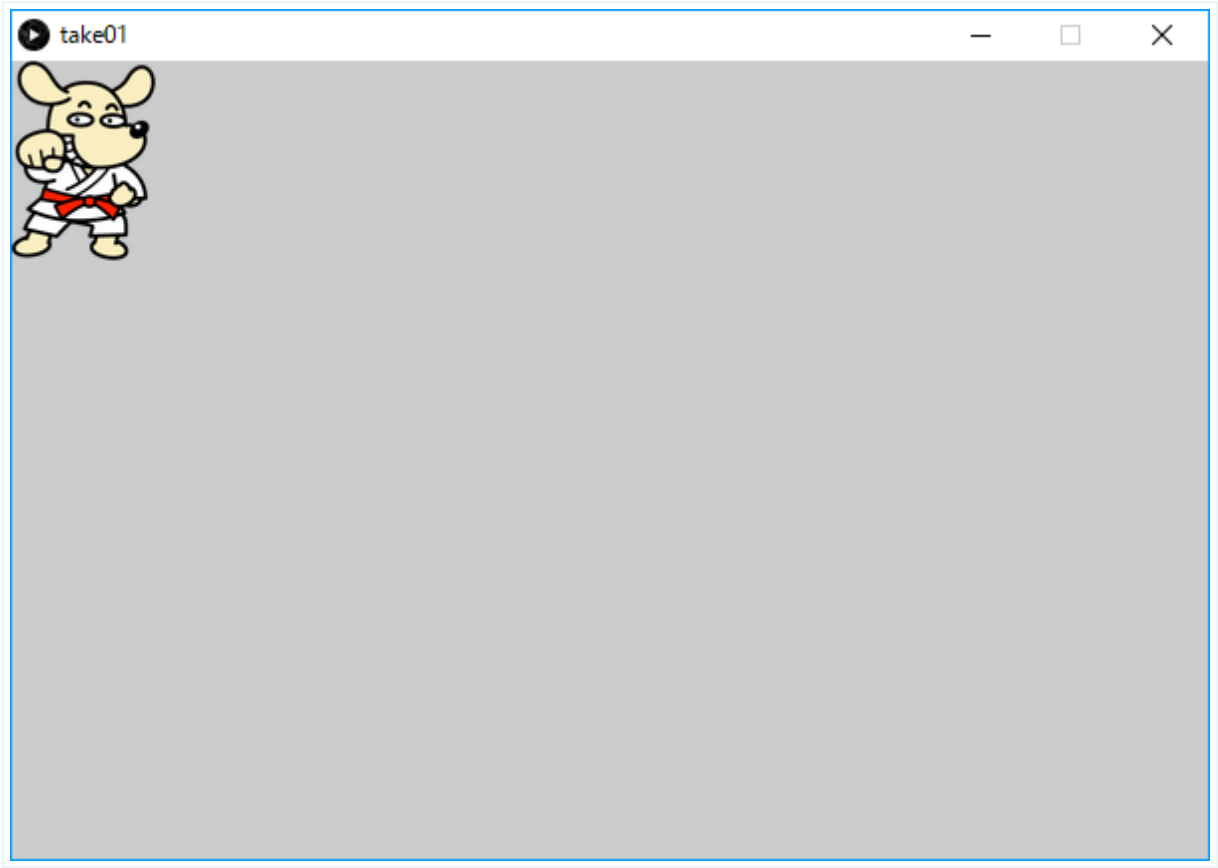


画像を表示する

画像を表示するには以下のようにします。

```
PImage karate;
void setup(){
    size(600,400);
    karate = loadImage("karate.png");
}

void draw(){
    image(karate,0,0);
}
```



画像を表示するしくみ

画像関係だけ抜き出し

```
PImage karate; // 画像の変数の定義
karate = loadImage("karate.png");
image(karate,0,0);
```

まず、`PImage karate;`で、画像をあつかうための変数（へんすう）karateを宣言します。
次に、先ほどフォルダに入れたkarate.pngを**loadImage()**で変数karateに読み込みます。最後は、**image()**を使って実際に表示します。

loadImage()の使い方

loadImage（ロードイメージ）は英語で「画像を読み込む」という意味です。

```
loadImage(画像ファイル);
```

画像ファイルは文字で指定するので、**"karate.png"**のように""で囲む必要があることに注意してください。

image()の使い方

image（イメージ）は英語で「画像」という意味です。

```
image(画像の変数,x座標,y座標);
```

最初に画像の変数を入れ、次に表示する場所を座標で表します。この座標は、画像の**左上の位置**になります。

変数を使おう

変数（へんすう）を使うと、図形や画像を動かしてアニメーションを作ることができます。

変数とは

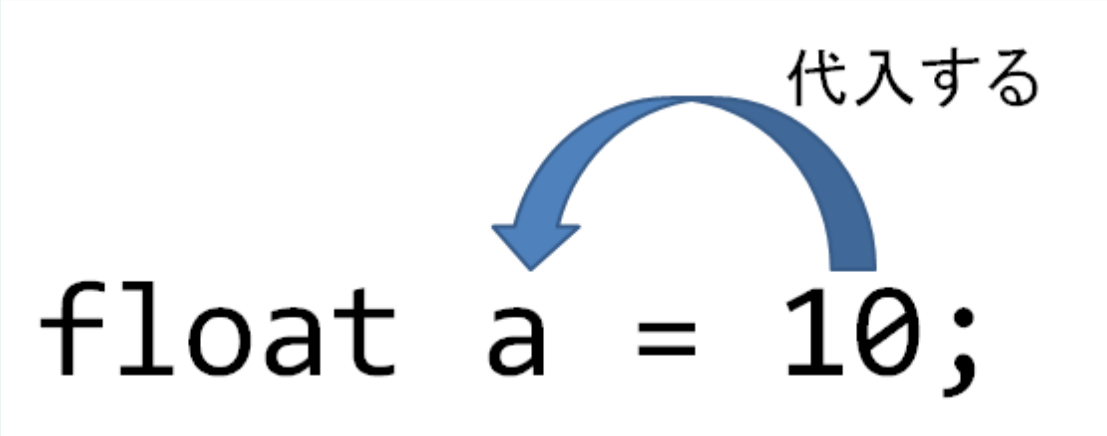
変数というのは、その名のとおり中身が変えられる数のことです。

```
float x = 10;
x = 24.2;
```

ここでは、`x`という名前の変数を作り、最初に中身に`10`を入れています。そのあと、中身を`24.2`に変えています。`float`（フロート）というのは、この変数xの種類で、「小数」を意味しています。Processingでは、変数を使うとき必ず変数の種類を決める必要があります。変数の名前は`x`でも`a`でも`tento`でも`TENTO`でもなんでもかまいません。わかりやすい名前にしましょう。

代入

Processingでは、`=`は「同じ」という意味ではありません。右のデータを左の変数に入れるという意味です。これを**代入（だいにゅう）**といいます。



変数の中身を増やす

変数は、中身を増やしたり減らしたりして使うことがふつうです。以下は増やす例です。

```
float a = 23;
a = a + 5;
```

ここでは、aの値（あたい）を23にしたあと、5を足しています。`a = a + 5`は、右側のデータを左に代入するということなので、**「aに5を足した数をaに代入する」**という意味です。なので、この行の後、aには`28`が入っています。

変数の中身を減らす

変数の中身は減らすこともできます。

```
float a = 5;
a = a - 8;
```

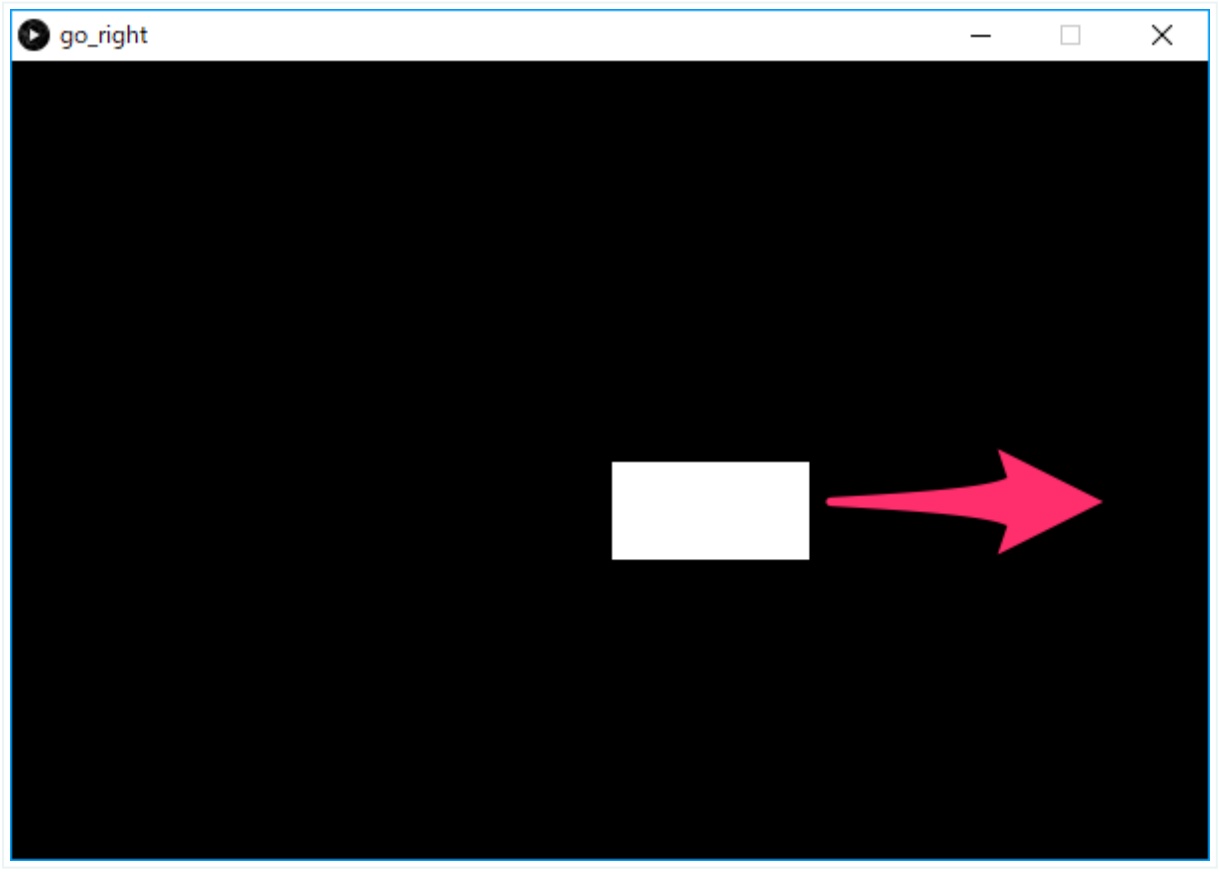
ここでは、aの値（あたい）を5にしたあと、8を引いています。`a = a - 8`は、**「aから8を引いた数をaに代入する」**という意味です。なので、この行の後、aには`-3`が入っています。

右に動く四角形

```
float x = 0;
void setup(){
    size(600,400);
}

void draw(){
    background(0,0,0);
    x = x + 5;
    rect(x,200,100,50);
}
```

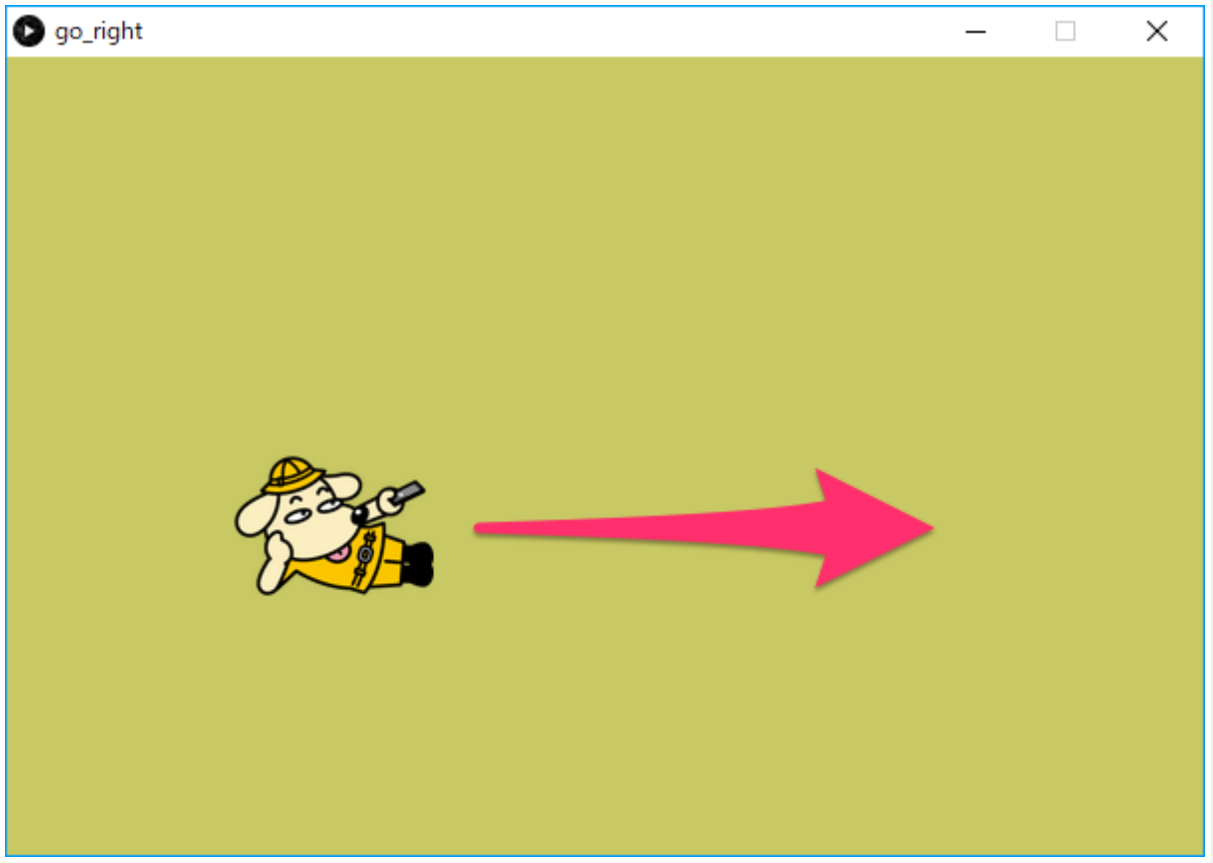
drawの中でxを毎回５ずつ大きくしています。rectの位置も変わっていきますね。



右に動くテントくん

```
float x;
PImage p;
void setup(){
  size(600,400);
  x = 0;
  p = loadImage("42.png");
}
```

```
void draw(){
  x = x + 10;
  background(200,200,100);
  image(p,x,200);
}
```



プログラミング授業(10)：副教材　繰り返す画像のアニメーション

画面の右端まで行ったらまた左にもどるアニメーション

```
PImage a;
float x = 0;
void setup(){
    size(600,400);
    a = loadImage("20.png");
}
void draw(){
    x = x + 1;
    if(x > 600){
        x = 0;
    }
    background(0, 0, 0);
    image(a,x,0);
}
```

ポイント

```
    if(x > 600){
        x = 0;
    }
```

画面の横幅は600なので、xが600を超えたら0にする、というプログラムを書いている。

画面の下まで行ったらまた上にもどるアニメーション

```
PImage a;
float y = 0;
void setup(){
    size(600,400);
    a = loadImage("20.png");
}
void draw(){
    y = y + 1;
    if(y > 400){
        y = 0;
    }
    background(0, 0, 0);
    image(a,0,y);
}
```

ポイント

```
    if(y > 400){
        y = 0;
    }
```

画面の横幅は600なので、xが600を超えたら0にする、というプログラムを書いている。

image()で画像を拡大する方法

```
image(a, 0, 0, 100, 150);
```

のように使う。`100`が画像の横幅で、`150`が画像の縦の高さになる。
まとめると、

`image(画像の変数, 表示する位置のX座標, 表示する位置のY座標, 画像の幅, 画像の高さ)`