

プログラミング演習3 (Python) ⑥ゲーム制作ライブラリ Pygame

【Python を使ったゲーム制作】

ゲームを作るためにプログラムを学ぶのであれば、自分で使いたい開発環境、もしくは、学校や会社等で使っている開発環境に合わせて言語を選択した方が効率が良いだろう。ここでは Python のゲーム開発ライブラリ pygame を使ってゲーム制作をしていく。

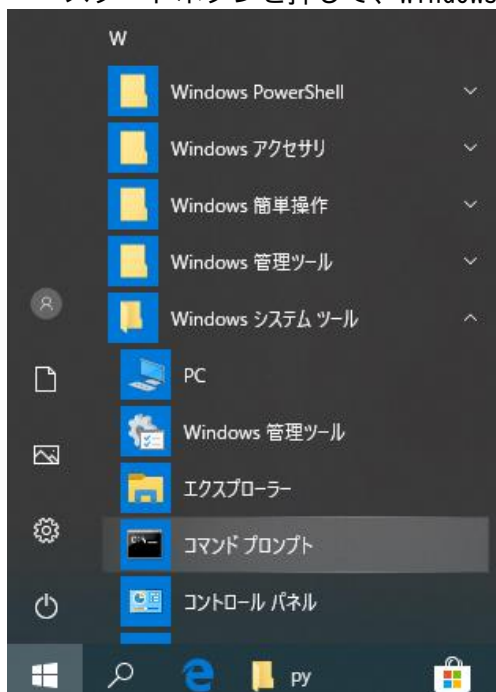
0. Pygame をインストールする

Python を Google Colab 上で実行した場合、pygame の使えない機能があるため、自分の PC に Python をインストールし、ローカル環境で実行する。実行する方法はいくつか紹介する。

1. コマンドプロンプトや PowerShell、Python シェルなどを使ってプログラムを実行する。

■コマンドプロンプトを起動する

スタートボタンを押して、Windows システムツールからコマンドプロンプトを選択する



「Python -V」と入力して、Python がインストールされているか確認する



バージョンが表示されているのなら、インストールされているが、エラーメッセージが出たら、下記の PythonJAPAN 「Windows 版 Python のインストール」

<https://www.python.jp/install/windows/install.html>

からインストールする。

Pygame がインストールされていない場合は

```
py -m pip install pygame
```

とコマンドラインに入力してインストールする。

このとき、pip コマンドが古くなっていると警告文が出たときは、

```
py -m pip install -upgrade pip
```

とすると警告文が出なくなる。

※注意※ 学校等のパソコンの場合、電源を切って再起動すると、古いバージョンに戻ってしまうので、毎回上記のコマンドを入力する必要がある。

6-1. サンプルプログラムを起動する

Pygame にはサンプルプログラムがあり、それを使って動作確認ができる。

下記のコードをメモ帳などのエディタで入力して、ファイルの種類はテキストファイルではなく、すべてのファイルにして、1.py と拡張子を変えて名前を付け、デスクトップに「py」という名前のフォルダを作って、その中に保存する。

Py6-1.py サンプルプログラムを動かす

001	a = int(input('gameNo?(1-3)'))
002	if a==1:
003	import pygame.examples.aliens as game
004	game.main()
005	elif a==2:
006	import pygame.examples.stars as game
007	game.main()
008	elif a==3:
009	import pygame.examples.testsprite as game
010	game.main()
011	else:
012	print('入力が間違っています')

コマンドプロンプト上でファイルのある場所に移動するには、

cd と入力した後、半角スペースを押して、目的のファイルがあるフォルダの場所を指定する。フォルダの中身を確認するには dir と入れて Enter を押す。

目的の場所へ移動するとき、下記の状態で

C:\Users\●●>cd ←半角スペースが入っている

デスクトップ上の py フォルダをコマンドプロンプトにドラッグ&ドロップすると

C:\Users\●●>cd C:\Users\●●\Desktop\py

とパスが入力されるので（このやり方は PowerShell ではできない。Python シェルはメニューからエディタを開いてファイルを指定する）

Enter キーを押すと

C:\Users\●●\Desktop\py>

となるので、

C:\Users\●●\Desktop\py>py3-1.py

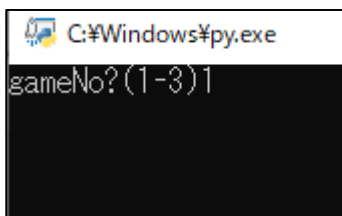
と py フォルダの中に保存したプログラムのファイル名を入力して

```
コマンド プロンプト
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

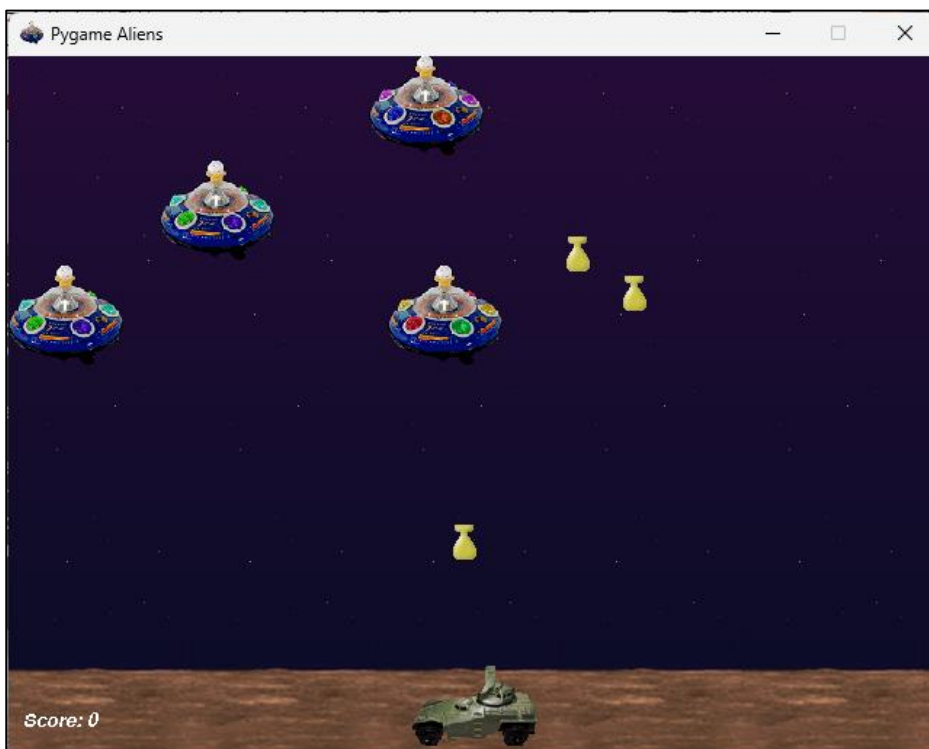
C:\Users\>cd C:\Users\\Desktop\py

C:\Users\Desktop\py>1.py
```

Enter キーを押すと、ウィンドウが表示されて、1-3 中の数字を選択すると



サンプルプログラムが実行される

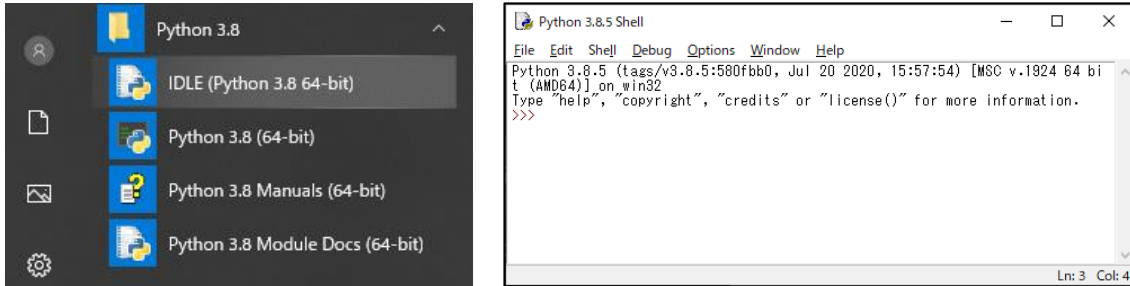


なお、python を起動させた後、コマンドラインで
import pygame.examples.aliens as game と入力して Enter キーを押した後、
game.main() （Enter キーを押す） と入力して Enter キーを押せば、
1 番のゲームが起動できる

■IDLEのエディタで起動する方法

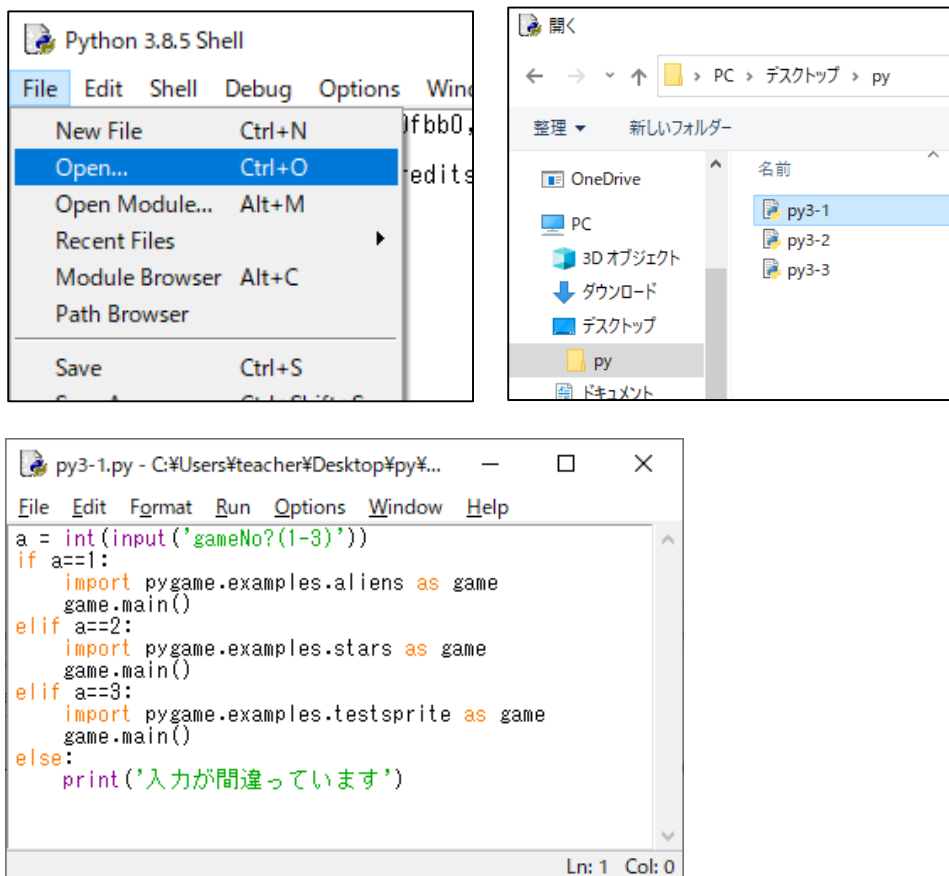
1. Python シェルを起動する

スタートボタンを押し、Python のアイコンをクリックすると「IDLE~~」という名前のアイコンが出てくるので、それをクリックすると Python シェルが起動される。

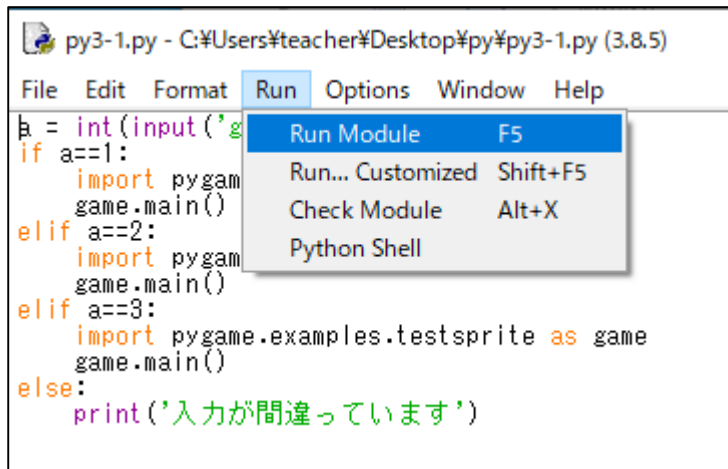


2. File の Open から目的のファイルを選択する

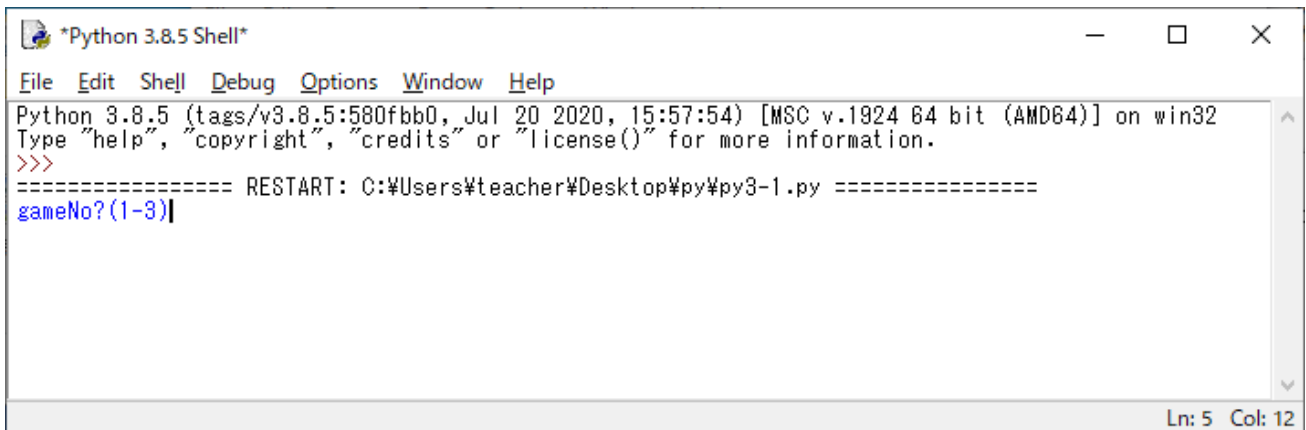
Python シェルの File タブをクリックし、Open を押して、目的の python ファイルを選択すると、エディタが開く。



Run メニューの「Run Module」を選択すれば、

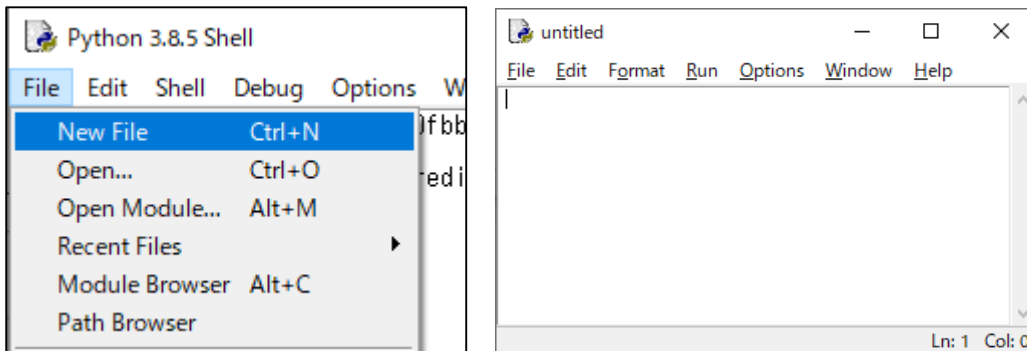


プログラムが Python シェルで実行される。

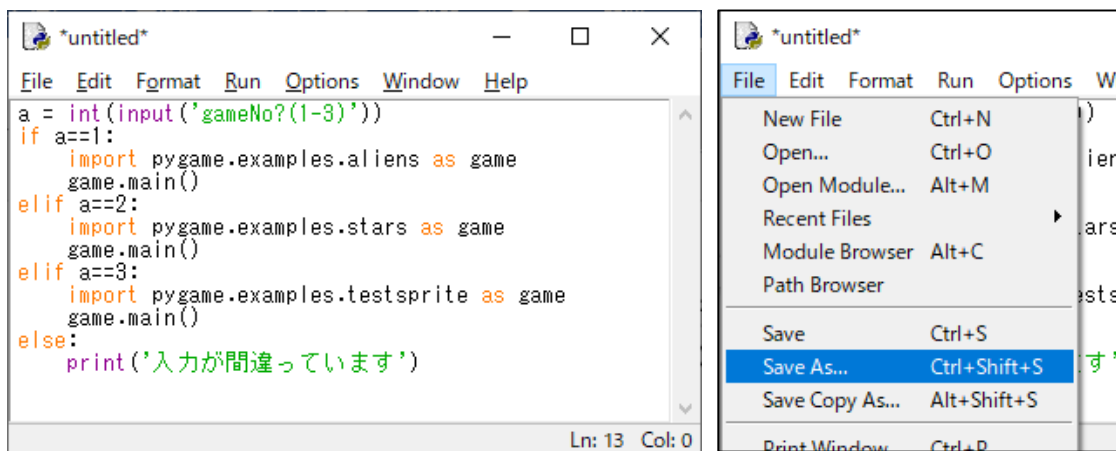


■プログラムファイルを用意していない場合

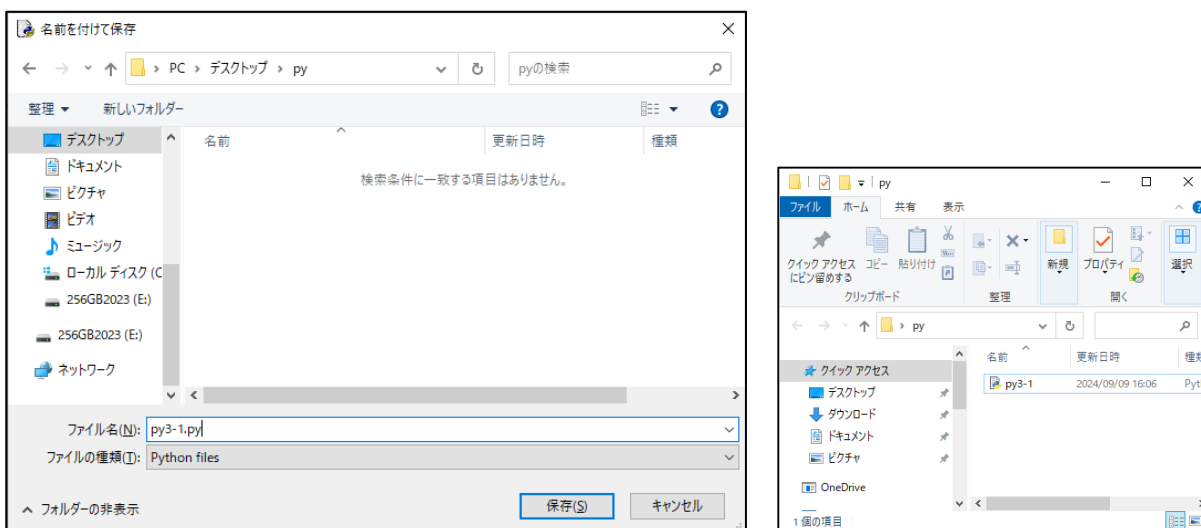
Python シェルの File の New File をクリックしてエディタを立ち上げる



プログラムを入力し、File の Save As を選択し



デスクトップの py フォルダに入って、ファイル名を入力して保存ボタンを押せば、py フォルダに入力したファイル名の py ファイルができる。



あとは、先述のように Run メニューの「Run Module」を選択して、Python シェルでプログラムを実行する。

6-2. 図形を描く

Pygame を使って図形を描画する。座標は左上が原点（0，0）で右に進むと x の値が増え、下に進むと y の値が増える。

四角形を描画する

pygame.draw.rect(画面オブジェクト, (赤, 緑, 青), (上座標, 左座標, 幅, 高さ), 線の幅)

円を描画する

pygame.draw.circle(画面オブジェクト, (赤, 緑, 青), 中心座標, 半径, 線の幅)

円を描画する 2

pygame.draw.ellipse(画面オブジェクト, (赤, 緑, 青), (X, Y, 幅, 高さ), 線の幅)

線を描画する

pygame.draw.line(screen, (赤, 緑, 青), (x1, y1), (x2, y2), 線の幅)

py6-2.py 四角形を描く

```
001 import pygame
002 pygame.init()
003 screen = pygame.display.set_mode((600, 400))
004 while True:
005     pygame.event.clear()
006     key_pressed = pygame.key.get_pressed()
007     if key_pressed[pygame.K_ESCAPE]:
008         break
009     screen.fill(pygame.Color("white"))
010     pygame.draw.rect(screen, pygame.Color("blue"), (100, 50, 100, 150))
011     pygame.display.update()
012 pygame.quit()
```

(解説)

1 行目 pygame をインポートする

2 行目 これから pygame を使うという初期化をする

3 行目 ゲーム用のウィンドウ（幅 600 縦 400）を作る

4 行目 while の条件を True として、ずっと繰り返すようにしている。

6～8 行目 グループの中にエスケープキーが押されるという条件で break を入れて、while から抜けられるようにしている。

9 行目 スクリーンの色を白で塗りつぶす（画面を初期化している）

10 行目 (100, 50) の場所に幅 100、高さ 150 の青い四角形を描く

11 行目 あらかじめスクリーンで描いておいたもの（白く塗りつぶしてその上に青い四角形を描いたもの）を表示させる

12 行目 pygame を終わらせる。

6-2-2. プログラムの改良

pygame と毎回書くのは面倒なので、

```
import pygame as pg
```

として、pg と省略して書けるようにする。

エスケープキーを押すのではなく、閉じるボタンでとじられるように

```
for event in pg.event.get():
```

```
    if event.type == pg.QUIT:
```

```
        pg.quit
```

と入力する。

ボタンが押された時、Python に「何かが押された」というお知らせが届き、イベントがあったときに、event.type を調べ、閉じるボタンが押されたときは pg.QUIT なので、それを条件に終了処理が行わせる。

また、pygame 試合を終了させただけになるので、外側のウィンドウが終了できるように、pg.quit() の下に sys.exit() を入力する。その際、sys を呼び出せるように、import pygame as pg.sys と追加する。

py6-2-2.py 四角形を描く2

```
001 import pygame as pg, sys
002 pg.init()
003 screen = pg.display.set_mode((600, 400))
004 while True:
005     pg.event.clear()
006     screen.fill(pg.Color("white"))
007     pg.draw.rect(screen, pg.Color("blue"), (100, 50, 100, 150))
008     pg.display.update()
009     for event in pg.event.get():
010         if event.type == pg.QUIT:
011             pg.quit()
012             sys.exit()
```

問 四角形の代わりに円を表示するプログラムを作りなさい

(参考) IT よろず雑記帳

<https://mulberrytassel.com/pygame-start-11/>

(参考) コンピュータ実習 I (Python) » 2. hello, pygame

https://web.tohoku.ac.jp/kc_kyomu/computer_seminar1/py/textbook_ja/hello.html

(参考図書) 森巧尚著「ゲーム作りで楽しく学ぶ Python の基本」(マイナビ)

6-3. 画像ファイルを読み込んでスクリーンに描く

描画する画像ファイルを用意して、それを読み込みこませて、その画像をスクリーンに描画する

画像の読み込みには image.load 関数を使う。画像ファイルが py ファイルと同じフォルダ内の img フォルダの中の pic6-3.png を読み込ませるときは、

```
pygame.image.load("img/pic6-3.png")
```


と指定する。

- ・ 画像の読み込み

```
画像変数=pygame.image.load( “画像ファイルパス” )
```

その画像を描画するには、screen の blit 関数を使う。読み込んだ画像を入れた、上記の画像変数と表示する位置 (X,Y) を指定して、下記のように記述する

- ・ 画像の描画

```
screen.blit (画像変数, (X,Y))
```

ライブラリ名を pg 省略して、画像変数 pic1 を設定して、上記2つを続けて記述すると、

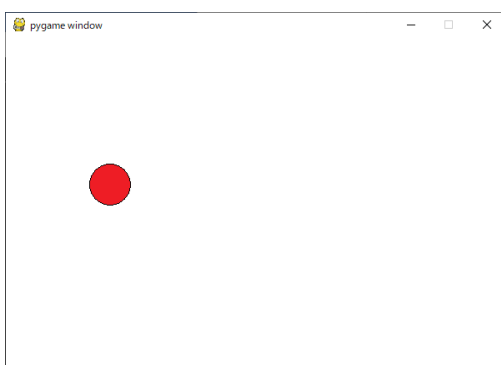
```
pic1 =pg.image.load("img/pic6-3.png" )  
screen.blit(pic1, (100,150))
```

となり、それを py3-2-2 の四角形を描く 2 の四角形の代わりに記述すると、指定した画像が表示される。

py6-3.py 画像ファイルを読み込んで描画する

```
001 import pygame as pg, sys  
002 pg.init()  
003 screen = pg.display.set_mode((600, 400))  
004 while True:  
005     pg.event.clear()  
006     screen.fill(pg.Color("white"))  
007     pic1 =pg.image.load( “img/pic6-3.png” )  
008     screen.blit(pic1, (100,150))  
009     pg.display.update()  
010     for event in pg.event.get():  
011         if event.type == pg.QUIT:  
012             pg.quit()  
013             sys.exit()
```

(実行結果)



■画像のサイズを変更する

画像のサイズを変更する場合

画像変数 = pg.transform.scale(画像変数, (幅, 高さ))
と記述する

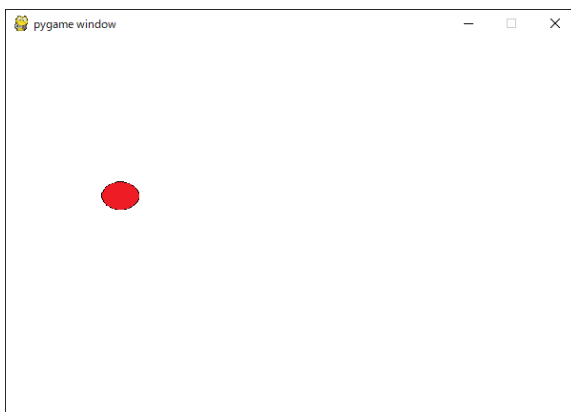
py6-4.py 画像のサイズを変更する

```
001 import pygame as pg, sys
002 pg.init()
003 screen = pg.display.set_mode((600, 400))
004 while True:
005     pg.event.clear()
006     screen.fill(pg.Color("white"))
007     pic1 = pg.image.load( "img/pic6-3.png" )
008     pic1 = pg.transform.scale(pic1, (80, 30))
009     screen.blit(pic1, (100, 150))
010     pg.display.update()
011     for event in pg.event.get():
012         if event.type == pg.QUIT:
013             pg.quit()
014             sys.exit()
```

(説明)

8行目 p i c 1 の画像を幅 40, 高さ 30 とする

(実行結果)



6-5. 文字列を描く

Pygame は文字を画像にしてから描画をするので、フォントの設定をし、フォントを使って文字列の画像を作り、その画像を描画する手順を踏む。

フォントを読み込んで文字列の画像を作るには、

font = pg.font.Font(None, 文字サイズ) ※別途日本語フォントを用意するときは None に入れる
画像変数=font.render(文字列, True, 色)

画像が用意できたら、screen の blit 関数で描画する

py6-5.py 文字列を描画する

```
001 import pygame as pg, sys
002 pg.init()
003 screen = pg.display.set_mode((600, 400))
004 font = pg.font.Font(None, 50)
005 textimg = font.render("hello!", True, pg.Color("BLUE"))
006 while True:
007     pg.event.clear()
008     screen.fill(pg.Color("white"))
009     screen.blit(textimg, (200, 100))
010     pg.display.update()
011     for event in pg.event.get():
012         if event.type == pg.QUIT:
013             pg.quit()
014             sys.exit()
```

(説明)

4行 フォントの設定をする（フォントは用意していない、フォントサイズは50）

5行 変数 textimg に、4行で設定したフォントに青い色の設定をして、文字列「hello!」を入れる

9行 screen の 200, 100 の場所に textimg の画像を描画する

(実行結果)



■日本語を使う場合

Pygame は日本語に対応していない。表示するにはパソコンに合わせて、設定をする必要があり、2通りの方法がある。

1. システムフォントを使う方法

そのパソコンが持っている日本語フォントを探して使う方法で、内部的なフォント名を使う。下記のコードで調べることができる。

```
for x in pg.font.get_fonts():  
    print(x)
```

先ほどの font の設定の代わりに下記のように入力すると、動くこともある。

Windows の場合、以下のいずれかで動くこともある

```
font=pg.font.SysFont( "msgothicmsuigothicmspgothic" ,50)  
font=pg.font.SysFont( "meiryomeiryomeiryomeiryomeiitalic" ,50)  
font=pg.font.SysFont( "yumincho" ,50)
```

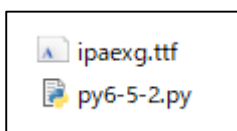
macOS の場合、以下のいずれかで動くこともある

```
font=pg.font.SysFont( "hiraginosansgb" ,50)  
font=pg.font.SysFont( "fottsukuardgothicstde" ,50)  
font=pg.font.SysFont( "applegothic" ,50)
```

2. IPAex フォントを使う方法

独立行政法人情報処理推進機構（IPA）によって配布されているフォントを使う方法。プログラムと一緒に用意する手間はあるが、確実にそのフォントを使って表示できる。

IPA の URL <https://moji.or.jp/ipafont/ipafontdownload/> から「IPAex フォント Ver. xxx.xx」をクリックすると、ダウンロードページが表示される。フォントのリンクをクリックして、利用規約を読み、2 書体パックをクリックして、圧縮ファイルをダウンロードし、そのファイルをダブルクリックして解凍して出てくるフォルダの中の「ipaexg.ttf」ファイルを py ファイルと同じ場所にコピーする。



py6-5-2.py 日本語フォントを使って描画する

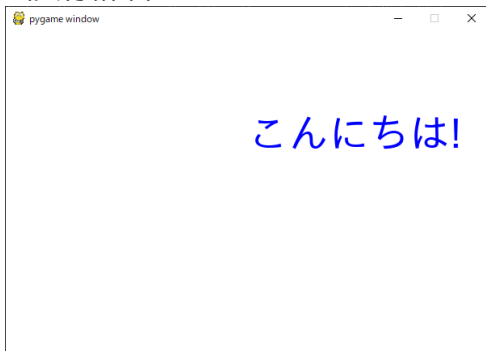
001	import pygame as pg, sys
002	pg.init()
003	screen = pg.display.set_mode((600, 400))
004	font = pg.font.Font("ipaexg.ttf",50)
005	textimg = font.render("こんにちは!", True, pg.Color("BLUE"))
006	while True:
007	pg.event.clear()

```

008     screen.fill(pg.Color("white"))
009     screen.blit(textimg, (200,100))
010     pg.display.update()
011     for event in pg.event.get():
012         if event.type == pg.QUIT:
013             pg.quit()
014             sys.exit()

```

(実行結果)



6-6. 画像を動かす

ループの中で、少しずつ位置を変えていけば、人間には動いているように見える。ゲームでは、キャラクターの位置だけでなく、弾の当たり判定などで「幅、高さ」もまとめて扱う必要があり、変数 Rect を使って、まとめて扱う。

位置と大きさをまとめて扱う

変数=pg.Rect(X,Y,幅,高さ)

Rect のそれぞれの値には、「x,y,width,height」をつけてアクセスする。たとえば、X座標を指定するときは、「変数.x」。幅を指定するときは、「変数.width」と記述する。

ここで、1フレーム間のデータの処理を表現する

py6-6.py 右に2、下に1動かすとした時の座標の値を表示

```

001 Import pygame as pg,sys
002 a=pg.Rect(40,30,50,50)
003 print("POS:",a.x, a.y," SIZE:",a.width, a.height)
004 a.x=a.x+2
005 a.y=a.y+1
006 print("POS:",a.x, a.y," SIZE:",a.width, a.height)

```

(実行結果)

```
C:\Users\student\Desktop\py>py py6-6.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org
POS: 40 30 SIZE: 50 50
POS: 42 31 SIZE: 50 50
```

2行目の位置の値が変わっている

6-6-2 位置を繰り返し変えて表示する
この変更をループの中に組み込んで、動き続けるように

py6-6-2.py 四角形を右に1ずつ動かす

```
001 import pygame as pg, sys
002 pg.init()
003 screen = pg.display.set_mode((600, 400))
004 myrect = pg.Rect(40, 30, 50, 50)
005 while True:
006     pg.event.clear()
007     screen.fill(pg.Color("white"))
008     myrect.x = myrect.x + 1
009     pg.draw.rect(screen, pg.Color("RED"), myrect)
010     pg.display.update()
011     pg.time.Clock().tick(60)
012     for event in pg.event.get():
013         if event.type == pg.QUIT:
014             pg.quit()
015             sys.exit()
```

(説明)

4行 図形をどこに置くか、変数 myrect を作っておいて、位置と大きさの設定をする

8行 myrect のx座標を1増やす(右に1動く)

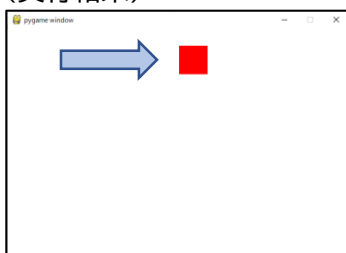
9行 draw.rect 関数で、位置と大きさの設定を myrect の値に変える

11行 処理の早いパソコンではスピードが出すぎるため、1秒間に60回以下のスピードに調整する

スピードを調整する

pg.time.Clock().tick (〇〇) 1秒間に〇〇回数以下のスピードにする

(実行結果)



問 四角形の代わりに 6-3.png ファイルを動かすプログラムを作れ

(ヒント)

画像を読み込み、読み込んだ画像を blit 関数で描画する。座標は myrect で渡せばいいので、

```
pic1 = pg.image.load( "img/pic6-3.png" )
```

で読み込ませた後、座標を変えて

```
screen.blit(pic1,myrect)
```

で描画する

(答え) `py6-6-2t.py` 画像を右に 1 ずつ動かす

```
004  pic1=pg.image.load("img/pic6-3.png")
005  myrect = pg.Rect(40, 30, 50, 50)
006  while True:
007      pg.event.clear()
008      screen.fill(pg.Color("white"))
009      myrect.x = myrect.x + 1
010      screen.blit(pic1,myrect)
```

6-7 画像をキーで動かす

キーやマウスを操作して、図形や画像が動かせるようにする。まず、押すキーのコードを読み取るプログラムを作る

どのキーが押されているか調べるコード

```
キー変数 = pg.key.get_pressed()
```

`py6-7.py` 左矢印と右矢印を判別する

```
001  import pygame as pg, sys
002  pg.init()
003  screen = pg.display.set_mode((600, 400))
004  while True:
005      pg.event.clear()
006      screen.fill(pg.Color("white"))
007      key=pg.key.get_pressed()
008      if(key[pg.K_RIGHT]):
009          print("RIGHT")
010      if(key[pg.K_LEFT]):
011          print("LEFT")
012      pg.display.update()
013      pg.time.Clock().tick(60)
```

014	for event in pg.event.get():
015	if event.type == pg.QUIT:
016	pg.quit()
017	sys.exit()

(説明)

7行 変数 key にどのキーが押されているかわかる情報が入る

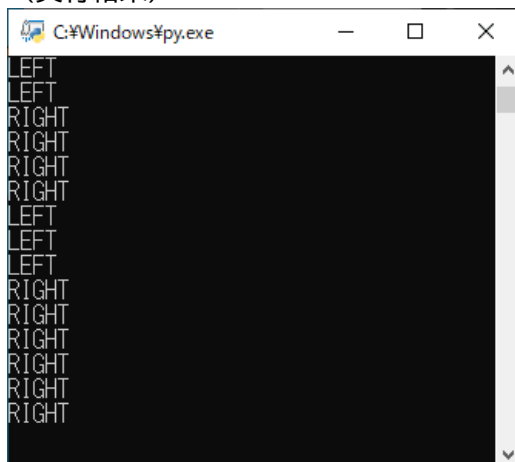
8行 右矢印が押されると key[pg.K_RIGHT]の値が True になり、9行の処理をする。

9行 RIGHT と表示する

10行 左矢印が押されると key[pg.K_LEFT]の値が True になり、11行の処理をする。

11行 LEFT と表示する

(実行結果)



py6-7-2.py (+α) キーコードを出して判別する

001	from pygame.locals import *
002	import pygame
003	import sys
004	pygame.init() # Pygame を初期化
005	screen = pygame.display.set_mode((400, 330)) # 画面を作成
006	pygame.display.set_caption("keyboard event") # タイトルを作成
007	while True:
008	screen.fill((0, 0, 0))
009	for event in pygame.event.get():
010	if event.type == QUIT:
011	pygame.quit()
012	sys.exit()
013	if event.type == KEYDOWN: # キーを押したとき
014	# ESC キーならスクリプトを終了
015	if event.key == K_ESCAPE:
016	pygame.quit()

017	sys.exit()
018	else:
019	print("押されたキー = " + pygame.key.name(event.key))
020	print("キーコード = " + str(event.key))
021	if event.key == 275:
022	print("右")
023	if event.key == 276:
024	print("左")
025	pygame.display.update()

(説明)

20～24行 文字コードを出して判別している

(実行結果)

```

コマンドプロンプト - py p...
押されたキー = up
キーコード = 273
押されたキー = down
キーコード = 274
押されたキー = left
キーコード = 276
左
押されたキー = right
キーコード = 275
右
押されたキー = a
キーコード = 97
押されたキー = b
キーコード = 98
押されたキー = c
キーコード = 99

```

(参考) <https://shizenkarasuzon.hatenablog.com/entry/2019/02/08/184932>

py6-7-3.py 左矢印と右矢印を使ってキャラクタを動かす

001	import pygame as pg, sys
002	pg.init()
003	screen = pg.display.set_mode((600, 400))
004	imgR = pg.image.load("img/pic6-7R.png")
005	myrect = pg.Rect(300, 200, 80, 100)
006	while True:
007	pg.event.clear()
008	screen.fill(pg.Color("white"))

```

009     vx=0
010     key=pg.key.get_pressed()
011     if(key[pg.K_RIGHT]):
012         vx = 5
013     if(key[pg.K_LEFT]):
014         vx = -5
015     myrect.x = myrect.x + vx
016     screen.blit(imgR,myrect)
017     pg.display.update()
018     pg.time.Clock().tick(60)
019     for event in pg.event.get():
020         if event.type == pg.QUIT:
021             pg.quit()
022             sys.exit()

```

(説明)

1 0行 キー入力を調べる

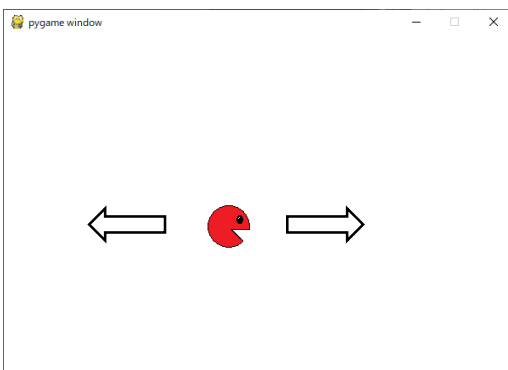
1 1、1 2行 右が押されていたらxの値を5増やせるようにvxに5を入れる

1 3、1 4行 左が押されていたらxの値を5減らせるようにvxに-5を入れる

1 5行 現在の場所にvxの値を足して動かす

1 6行 imgRの画像を現在の場所に描画する

(実行結果)



6-7-4. キャラクタの向きを変える

py6-7-3.py ではキャラクターが右向きなので、左矢印を押したときに左向きになるようにする。

画像を反転する

画像変数=pg.transform.flip(画像変数, 左右反転, 上下反転)

py6-7-4.py 左矢印と右矢印を使ってキャラクタを動かす

```
001 import pygame as pg, sys
002 pg.init()
003 screen = pg.display.set_mode((600, 400))
004 imgR = pg.image.load("img/pic6-7R.png")
005 myrect = pg.Rect(300, 200, 80, 100)
006 rightFlag = True
007 while True:
008     pg.event.clear()
009     screen.fill(pg.Color("white"))
010     vx=0
011     key=pg.key.get_pressed()
012     if(key[pg.K_RIGHT]):
013         vx = 5
014         rightFlag = True
015     if(key[pg.K_LEFT]):
016         vx = -5
017         rightFlag = False
018     imgL = pg.transform.flip(imgR, True, False)
019     myrect.x = myrect.x + vx
020     if rightFlag:
021         screen.blit(imgR, myrect)
022     else:
023         screen.blit(imgL, myrect)
024     pg.display.update()
025     pg.time.Clock().tick(60)
026     for event in pg.event.get():
027         if event.type == pg.QUIT:
028             pg.quit()
029             sys.exit()
```

(説明)

6行 右向きか左向きかを判別する変数 rightFlag を作る

14行 右ボタンを押したときは変数 rightFlag を True にする

17行 左ボタンを押したときは変数 rightFlag を False にする

18行 imgL に imgR を左右反転させた画像を入れる

20～23行 rightFlag が True だったら右向きの画像、False だったら左向きの画像を表示する

■マウスでキャラクターを動かす

6-8. マウスポインタの場所に円を表示する

マウスのどのボタンが押されているか取得するには

```
マウス変数=pg.mouse.get_pressed()
```

を使い、左ボタンなら[0]、右ボタンなら[2]が入る。

マウスポインタの座標を取得するには

```
(mx, my) = pg.mouse.get_pos()
```

と記述する。これを円を描画する座標に入れば、マウスポインタの場所に円が描画される

py6-8.py マウスポインタの場所を表示する

```
001 import pygame as pg, sys
002 pg.init()
003 screen = pg.display.set_mode((600, 400))
004 while True:
005     pg.event.clear()
006     screen.fill(pg.Color("white"))
007     mdown = pg.mouse.get_pressed()
008     (mx, my) = pg.mouse.get_pos()
009     if mdown[0]:
010         print(mx, my)
011     pg.display.update()
012     pg.time.Clock().tick(60)
013     for event in pg.event.get():
014         if event.type == pg.QUIT:
015             pg.quit()
016             sys.exit()
```

(説明)

7行 mdown にどのボタンが押されたか入る

8行 押した座標を取得する

9、10行 左ボタンを押したとき、コンソールに座標を表示する。

6-8-2. マウスでクリックした位置に四角形を表示する

先ほどの座標を表示させる代わりに、クリックしたところに四角形が表示されるようにする。

`py6-8-2.py` マウスポインタの場所に四角形を描画する

```
001 import pygame as pg, sys
002 pg.init()
003 screen = pg.display.set_mode((600, 400))
004 while True:
005     pg.event.clear()
006     screen.fill(pg.Color("white"))
007     mdown = pg.mouse.get_pressed()
008     (mx, my) = pg.mouse.get_pos()
009     if mdown[0]:
010         pg.draw.rect(screen, pg.Color("Red"), (mx-50, my-50, 100, 100))
011     pg.display.update()
012     pg.time.Clock().tick(60)
013     for event in pg.event.get():
014         if event.type == pg.QUIT:
015             pg.quit()
016             sys.exit()
```

9、10行 左ボタンを押したとき、赤い幅 100 高さ 100 の四角形が表示される四角形の中心がマウスポインタの場所になるように資格の左上の場所をそれぞれ-50 にしている。

(おまけ) GoogleColab でネズミを動かす

ネットで検索をすると ChatGPT を使って GoogleColab で画像を動かすプログラムコードを作っており、それを参考にして下記のソースを作った。

```
import pygame
from google.colab.patches import cv2_imshow
from google.colab import output
import cv2
import numpy as np
pygame.init()
width,height = 450,100
screen=pygame.display.set_mode((width,height))
def draw_nezumi(x):
    screen.fill((255,255,255))
    pygame.draw.rect(screen, (50,220,220), (x,height // 2,30,30))
    pygame.draw.circle(screen, (50,220,220), (x,height // 2), 10)
    pygame.draw.circle(screen, (50,220,220), (x+30,height // 2), 10)
    pygame.draw.circle(screen, (0,0,0), (x+8,height // 2+15), 3)
    pygame.draw.circle(screen, (0,0,0), (x+22,height // 2+15), 3)
    pygame.draw.circle(screen, (0,0,0), (x+2,height // 2+30), 4)
    view = pygame.surfarray.array3d(screen)
    view = view.transpose([1,0,2])
    img_bgr =cv2.cvtColor(view,cv2.COLOR_RGB2BGR)
    cv2_imshow(img_bgr)
for x in range(0,width - 50,5):
    draw_nezumi(x)
    output.clear(wait = True)
    pygame.time.delay(30)
pygame.quit()
```

(参考) https://note.com/sakura_kuku_/n/n04a25bfcbade

実際の画面

```
import pygame
from google.colab.patches import cv2_imshow
from google.colab import output
import cv2
import numpy as np
pygame.init()
width,height = 450,100
screen=pygame.display.set_mode((width,height))
def draw_nezumi(x):
    screen.fill((255,255,255))
    pygame.draw.rect(screen,(50,220,220),(x,height//2,30,30))
    pygame.draw.circle(screen,(50,220,220),(x,height//2),10)
    pygame.draw.circle(screen,(50,220,220),(x+30,height//2),10)
    pygame.draw.circle(screen,(0,0,0),(x+8,height//2+15),3)
    pygame.draw.circle(screen,(0,0,0),(x+22,height//2+15),3)
    pygame.draw.circle(screen,(0,0,0),(x+2,height//2+30),4)
    view = pygame.surfarray.array3d(screen)
    view = view.transpose([1,0,2])
    img_bgr = cv2.cvtColor(view,cv2.COLOR_RGB2BGR)
    cv2_imshow(img_bgr)
for x in range(0,width-50,5):
    draw_nezumi(x)
    output.clear(wait=True)
    pygame.time.delay(30)
pygame.quit()
#参考 https://note.com/sakura\_kuku/n/n04a25bfcgade
```



左から右へネズミの画像が動く