



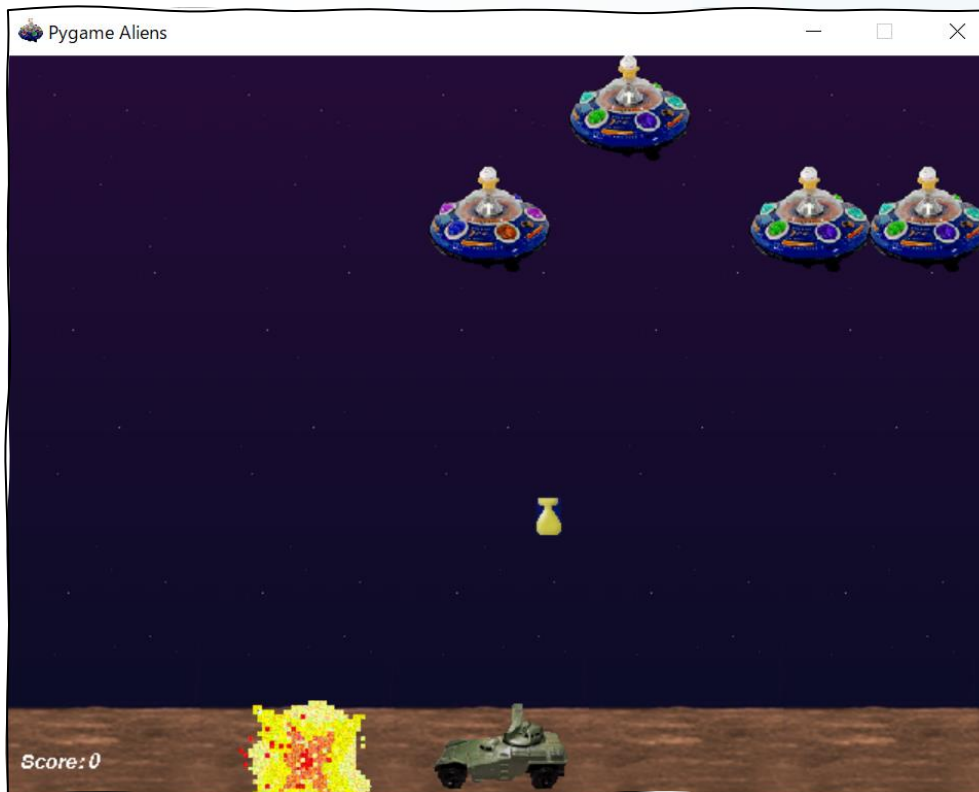
Pythonの道

pygameの使い方③

もくじ

・ pygameの使い方(②の続き)

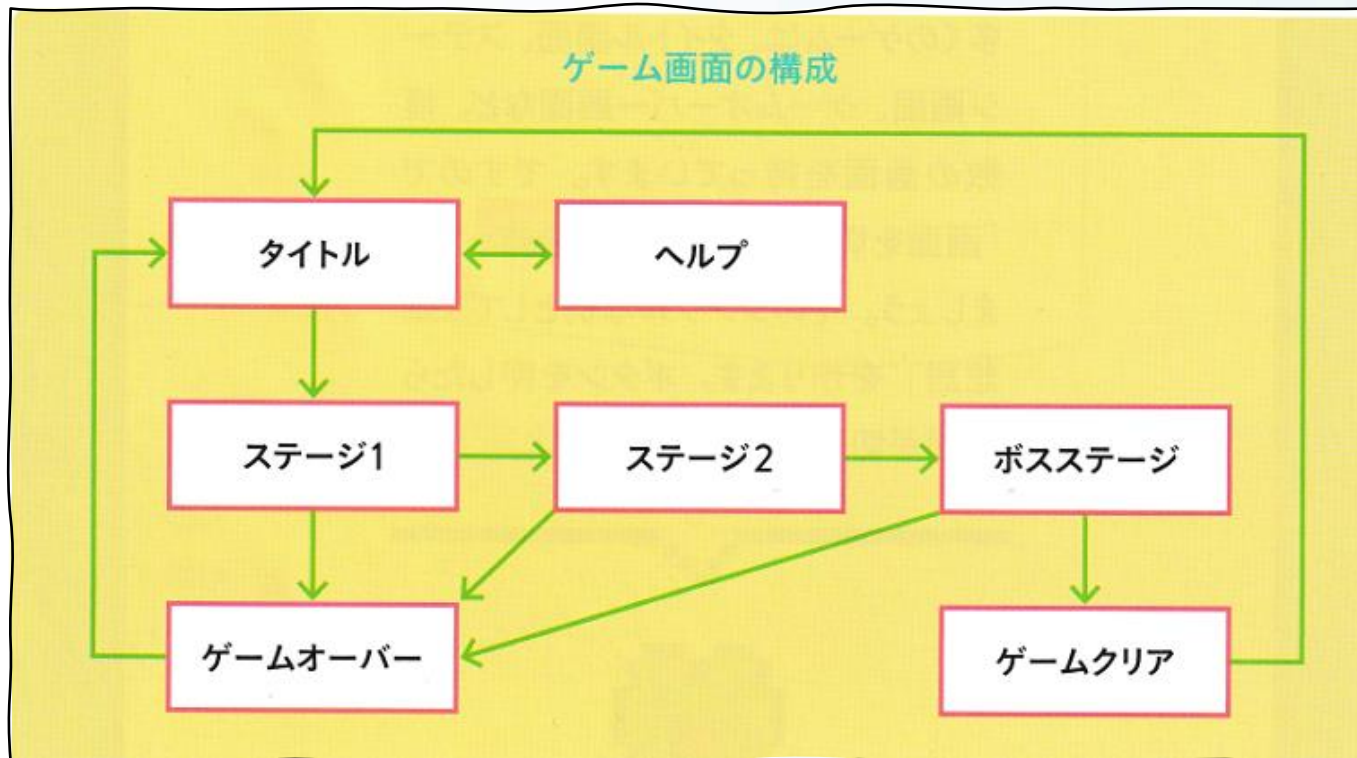
pygameを使って本格的なゲーム開発を学ぶ



画面の切り替えで紙芝居

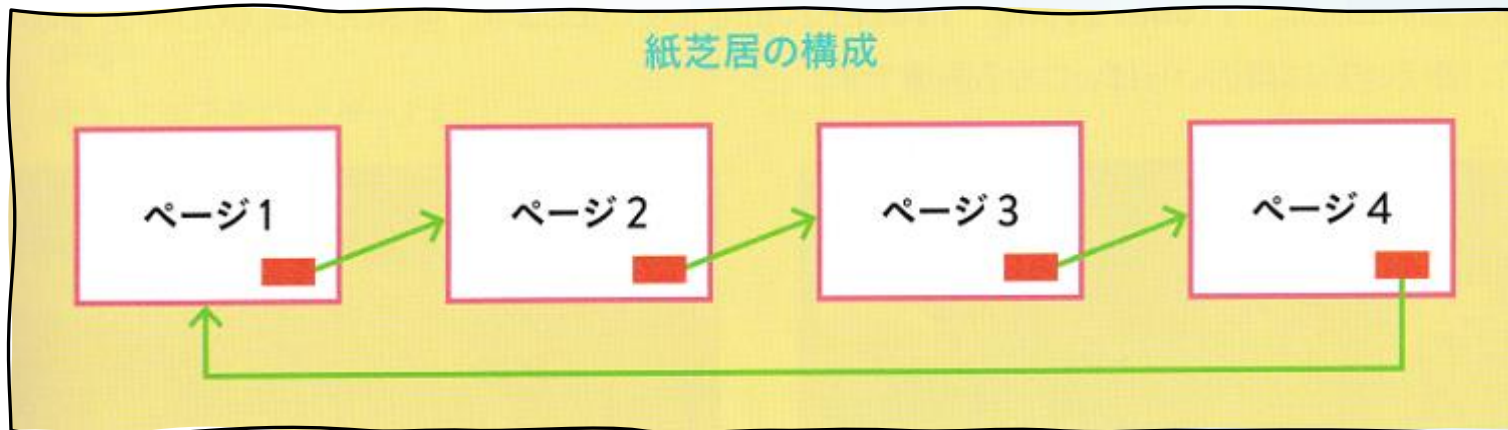
多くのゲームは、タイトル画面、ステージ画面、ゲームオーバー画面など、複数の画面を持っている。画面を切り替える仕組みを使って、紙芝居を作る。ボタンを押したら画面が切り替わる。

pygameではループを利用してゲームを作ってきたが、このような「複数の画面が切り替わるしかけ」はどうすれば、作れるのだろうか？



画面の切り替えで紙芝居

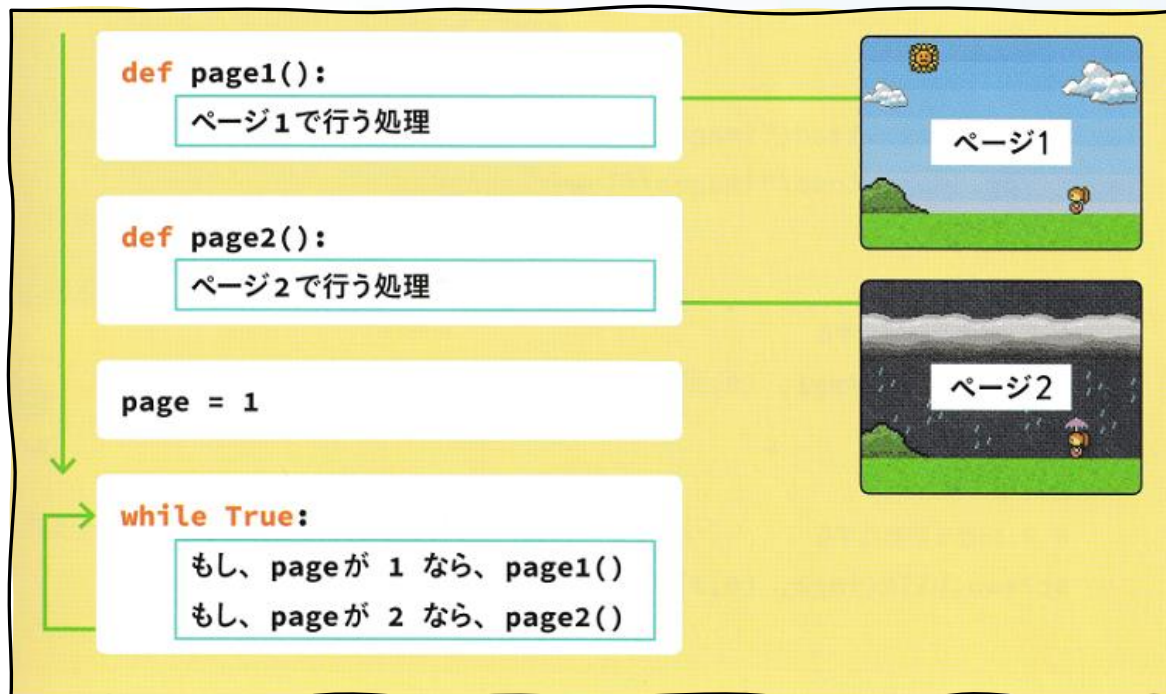
複数の画面を切り替えるしかけを一番シンプルにすると「紙芝居」になる。「各ページにボタンがあって、ボタンを押すと次のページへ進むだけ」というもの。



紙芝居を作ることができれば、あるページを「ステージ1」や「ボスステージ」に改造することで、画面が切り替わるゲームにしていけることができる。

画面の切り替えで紙芝居

紙芝居を作るうえで重要なのは1ページの処理は1つにまとめること。それぞれのページの処理を同時に行うことはないので、ページごとに行う処理をひとまとめにして分けて考える。



「1ページを1つの関数にまとめて」分けて作っていく。「今、どのページを表示するのか」は変数pageで設定する。ページ番号を入れておき、もし、pageが1ならページ1を表示するようにif文で判断する

画面の切り替えで紙芝居

ページ1が表示されるプログラムを作る

#ゲームの準備をする

```
import pygame as pg, sys
```

```
pg.init()
```

```
screen = pg.display.set_mode((800, 600))
```

#紙芝居をつくる

```
img1 = pg.image.load("flower1.png")
```

```
img2 = pg.image.load("flower2.png")
```

```
def page1():
```

#画面を初期化する

```
screen.blit(img1, (0,0))
```

```
def page2():
```

次のページに続く

画面の切り替えで紙芝居

#画面を初期化する

```
screen.blit(img2, (0,0))
```

```
page = 1
```

#この下をずっとループする

```
while True:
```

```
    if page == 1:
```

```
        page1()
```

```
    elif page == 2:
```

```
        page2()
```

#画面を表示する

```
pg.display.update()
```

```
pg.time.Clock().tick(60)
```

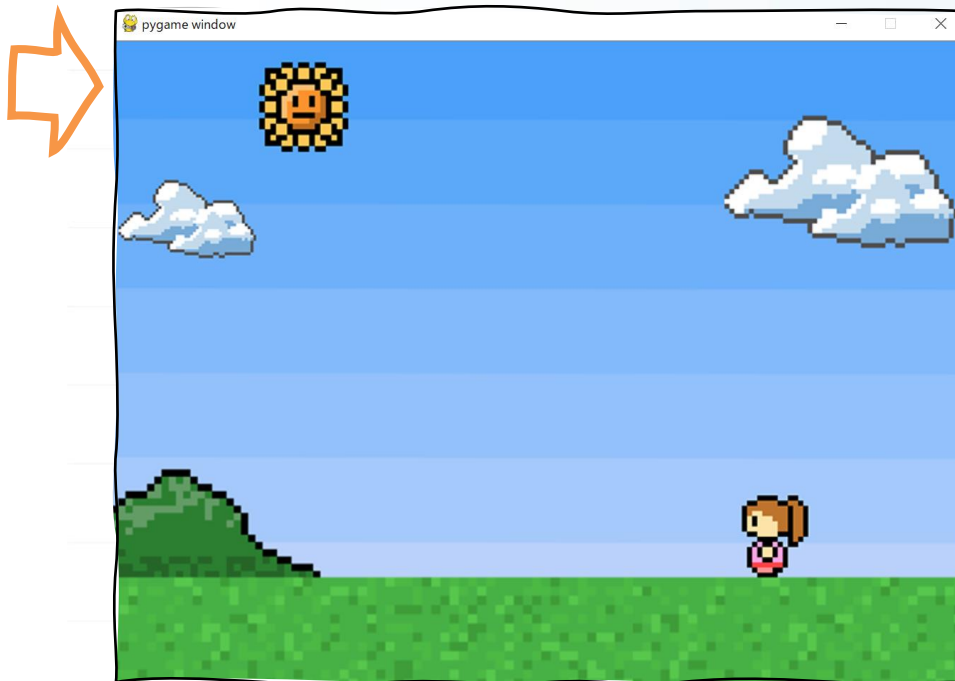
次のページに続く

画面の切り替えで紙芝居

#閉じるボタンが押されたら、終了する

```
for event in pg.event.get():  
    if event.type == pg.QUIT:  
        pg.quit()  
        sys.exit()
```

「Run Module」またはF5キーを押してプログラムを実行する。



画面の切り替えで紙芝居

次にページ2が表示されるプログラムを作る

#ゲームの準備をする

```
import pygame as pg, sys
```

```
pg.init()
```

```
screen = pg.display.set_mode((800, 600))
```

#紙芝居をつくる

```
img1 = pg.image.load("flower1.png")
```

```
img2 = pg.image.load("flower2.png")
```

```
def page1():
```

#画面を初期化する

```
screen.blit(img1, (0,0))
```

```
def page2():
```

次のページに続く

画面の切り替えで紙芝居

#画面を初期化する

```
screen.blit(img2, (0,0))
```

```
page = 2
```

#この下をずっとループする

```
while True:
```

```
    if page == 1:
```

```
        page1()
```

```
    elif page == 2:
```

```
        page2()
```

#画面を表示する

```
pg.display.update()
```

```
pg.time.Clock().tick(60)
```

次のページに続く

画面の切り替えで紙芝居

#閉じるボタンが押されたら、終了する

```
for event in pg.event.get():  
    if event.type == pg.QUIT:  
        pg.quit()  
        sys.exit()
```

「Run Module」またはF5キーを押してプログラムを実行する。



一本道の紙芝居

変数の値を変えるだけで、ページの切り替えができるようになったので、これに「ボタン機能」を追加して「紙芝居」を作る。ボタンを押したら、変数pageの値を変えていく。

```
def page1():
```

ページ1で行う処理

```
def page2():
```

ページ2で行う処理

```
page = 1
```

```
while True:
```

もし、pageが 1 なら、page1()

もし、pageが 2 なら、page2()



ボタンを押したら、page = 2



ボタンを押したら、page = 1

画面の切り替えで紙芝居

ボタンを1つ設置して、ページが切り替わるプログラムを作る。

#ゲームの準備をする

```
import pygame as pg, sys
```

```
pg.init()
```

```
screen = pg.display.set_mode((800, 600))
```

#紙芝居

```
img1 = pg.image.load("flower1.png")
```

```
img2 = pg.image.load("flower2.png")
```

#ボタン

```
next_img = pg.image.load("nextbtn.png")
```

```
pushFlag = False
```

#btnを押したら、newpageにジャンプする

```
def button_to_jump(btn, newpage):
```

次のページに続く

画面の切り替えで紙芝居

global page, pushFlag

#ユーザーからの入力を調べる

mdown = pg.mouse.get_pressed()

・ ・ マウスのどのボタンが押されたかを調べる関数

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

・ ・ マウスが画面のどこを指しているかを調べる関数

if mdown[0]: ・ ・ マウスの左ボタンがクリックされたとき

if btn.collidepoint(mx, my) and pushFlag == False:

page = newpage

・ ・ 関数が実行されたときにボタンの絵の範囲に入っているか

pushFlag = True

else:

pushFlag = False

def page1():

#画面を初期化する

screen.blit(img1, (0,0)) ・ ・ 読み込んだ画像を描画する

btn1 = screen.blit(next_img,(600, 540)) ・ ・ 画像を描画してその範囲を取得する
次のページに続く

画面の切り替えで紙芝居

#絵を描いたり、判定したりする

button_to_jump(btn1, 2)

def page2():

#画面を初期化する

screen.blit(img2, (0,0)) · · 読み込んだ画像を描画する

btn1 = screen.blit(next_img,(600, 540)) · · 画像を描画してその範囲
を取得する

#絵を描いたり、判定したりする

button_to_jump(btn1, 1)

page = 1

#この下をずっとループする

while True:

if page == 1:

page1()

elif page == 2:

次のページに続く

画面の切り替えで紙芝居

```
page2()
```

```
#画面を表示する
```

```
pg.display.update()
```

```
pg.time.Clock().tick(60)
```

```
#閉じるボタンが押されたら、終了する
```

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

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

```
        pg.quit()
```

```
        sys.exit()
```


画面の切り替えで紙芝居

「Run Module」またはF5キーを押してプログラムを実行する。



ボタンをクリック
するとページが変わる



画面の切り替えで紙芝居

紙芝居のページを増したプログラムを作る。

#ゲームの準備をする

```
import pygame as pg, sys
```

```
pg.init()
```

```
screen = pg.display.set_mode((800, 600))
```

#紙芝居

```
img1 = pg.image.load("flower1.png")
```

```
img2 = pg.image.load("flower2.png")
```

```
img3 = pg.image.load("flower3.png")
```

```
img4 = pg.image.load("flower4.png")
```

・ ・ 追加する箇所

#ボタン

```
next_img = pg.image.load("nextbtn.png")
```

```
pushFlag = False
```

次のページに続く

画面の切り替えで紙芝居



#btnを押したら、newpageにジャンプする

```
def button_to_jump(btn, newpage):
```

```
    global page, pushFlag
```

#ユーザーからの入力を調べる

```
    mdown = pg.mouse.get_pressed()
```

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

```
    if mdown[0]:
```

```
        if btn.collidepoint(mx, my) and pushFlag == False:
```

```
            page = newpage
```

```
            pushFlag = True
```

```
    else:
```

```
        pushFlag = False
```

```
def page1():
```

#画面を初期化する

次のページに続く

画面の切り替えで紙芝居

```
screen.blit(img1, (0,0))
```

```
btn1 = screen.blit(next_img,(600, 540))
```

```
#絵を描いたり、判定したりする
```

```
button_to_jump(btn1, 2)
```

```
def page2():
```

```
#画面を初期化する
```

```
screen.blit(img2, (0,0))
```

```
btn1 = screen.blit(next_img,(600, 540))
```

```
#絵を描いたり、判定したりする
```

```
button_to_jump(btn1, 3) . . . 修正する箇所
```

```
def page3():
```

```
#画面を初期化する
```

```
screen.blit(img3, (0,0))
```

```
btn1 = screen.blit(next_img,(600, 540))
```

. . . 追加する箇所

次のページに続く

画面の切り替えで紙芝居

#絵を描いたり、判定したりする

button_to_jump(btn1, 4)

def page4():

#画面を初期化する

screen.blit(img4, (0,0))

btn1 = screen.blit(next_img,(600, 540))

#絵を描いたり、判定したりする

button_to_jump(btn1, 1)

・ ・ 追加する箇所

page = 1

#この下をずっとループする

while True:

if page == 1:

page1()

elif page == 2:

次のページに続く

画面の切り替えで紙芝居



```
page2()
```

```
elif page == 3:
```

```
    page3()
```

```
elif page == 4:
```

```
    page4()
```

・ ・ 追加する箇所

```
#画面を表示する
```

```
pg.display.update()
```

```
pg.time.Clock().tick(60)
```

```
#閉じるボタンが押されたら、終了する
```

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

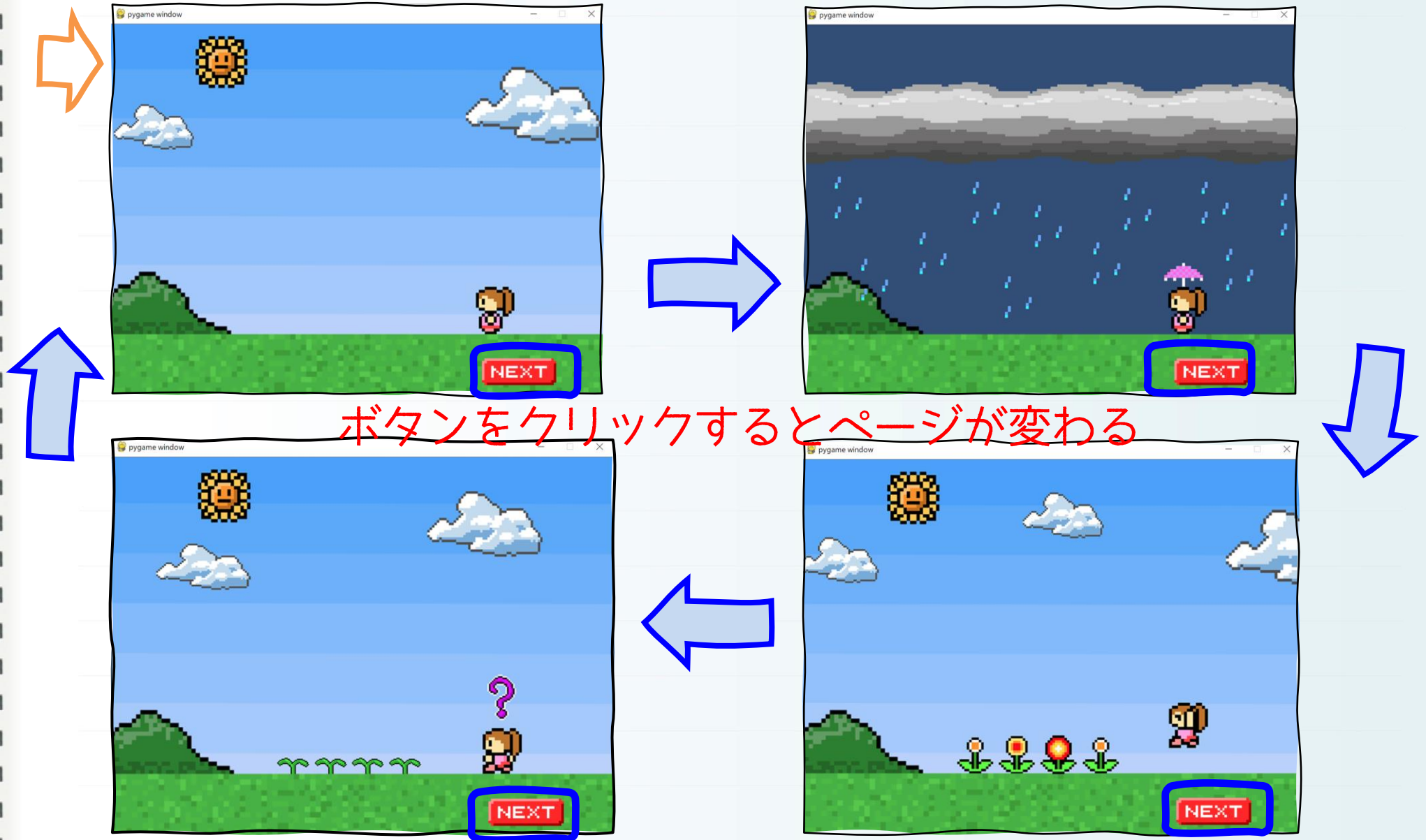
```
    if event.type == pg.QUIT:
```

```
        pg.quit()
```

```
        sys.exit()
```

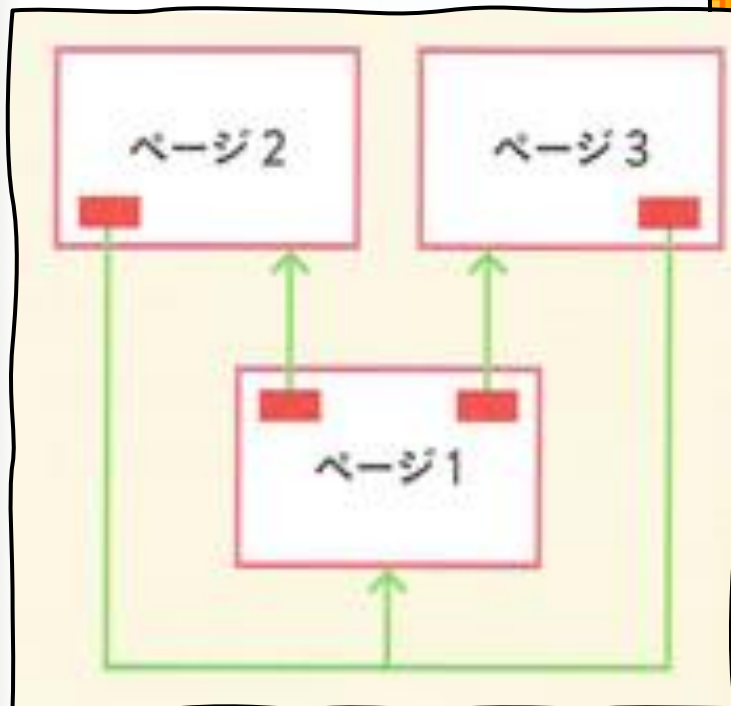
画面の切り替えで紙芝居

「Run Module」またはF5キーを押してプログラムを実行する。



枝分かれの紙芝居

1ページに複数ボタンを置いて、枝分かれして進む紙芝居を作ってみよう。最初のページで「2つの扉のどちらに進むか」を選択すると、ゲームオーバーかゲームクリアが表示される。



画面の切り替えで紙芝居

進む道を選択する紙芝居プログラムを作る。

#ゲームの準備をする

```
import pygame as pg, sys
```

```
pg.init()
```

```
screen = pg.display.set_mode((800, 600))
```

#紙芝居

```
img1 = pg.image.load("root1.png")
```

```
img2 = pg.image.load("root2.png")
```

```
img5 = pg.image.load("root5.png")
```

#ボタン

```
next_img = pg.image.load("nextbtn.png")
```

```
replay_img = pg.image.load("replaybtn.png")
```

```
pushFlag = False
```

次のページに続く

画面の切り替えで紙芝居



#btnを押したら、newpageにジャンプする

```
def button_to_jump(btn, newpage):
```

```
    global page, pushFlag
```

#ユーザーからの入力を調べる

```
    mdown = pg.mouse.get_pressed()
```

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

```
    if mdown[0]:
```

```
        if btn.collidepoint(mx, my) and pushFlag == False:
```

```
            page = newpage
```

```
            pushFlag = True
```

```
    else:
```

```
        pushFlag = False
```

```
def page1():
```

#画面を初期化する

次のページに続く

画面の切り替えで紙芝居



```
screen.blit(img1, (0,0))
```

```
btn1 = screen.blit(next_img,( 90, 220))
```

```
btn2 = screen.blit(next_img,(590, 220))
```

```
#絵を描いたり、判定したりする
```

```
button_to_jump(btn1, 2)
```

```
button_to_jump(btn2, 3)
```

```
def page2():
```

```
#画面を初期化する
```

```
screen.blit(img2, (0,0))
```

```
btn1 = screen.blit(replay_img,(600, 520))
```

```
#絵を描いたり、判定したりする
```

```
button_to_jump(btn1, 1)
```

```
def page3():
```

```
#画面を初期化する
```

次のページに続く

画面の切り替えで紙芝居

```
screen.blit(img5, (0,0))
```

```
btn1 = screen.blit(replay_img,(600, 520))
```

```
#絵を描いたり、判定したりする
```

```
button_to_jump(btn1, 1)
```

```
page = 1
```

```
#この下をずっとループする
```

```
while True:
```

```
    if page == 1:
```

```
        page1()
```

```
    elif page == 2:
```

```
        page2()
```

```
    elif page == 3:
```

```
        page3()
```

```
#画面を表示する
```

次のページに続く

画面の切り替えで紙芝居

```
pg.display.update()
```

```
pg.time.Clock().tick(60)
```

#閉じるボタンが押されたら、終了する

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

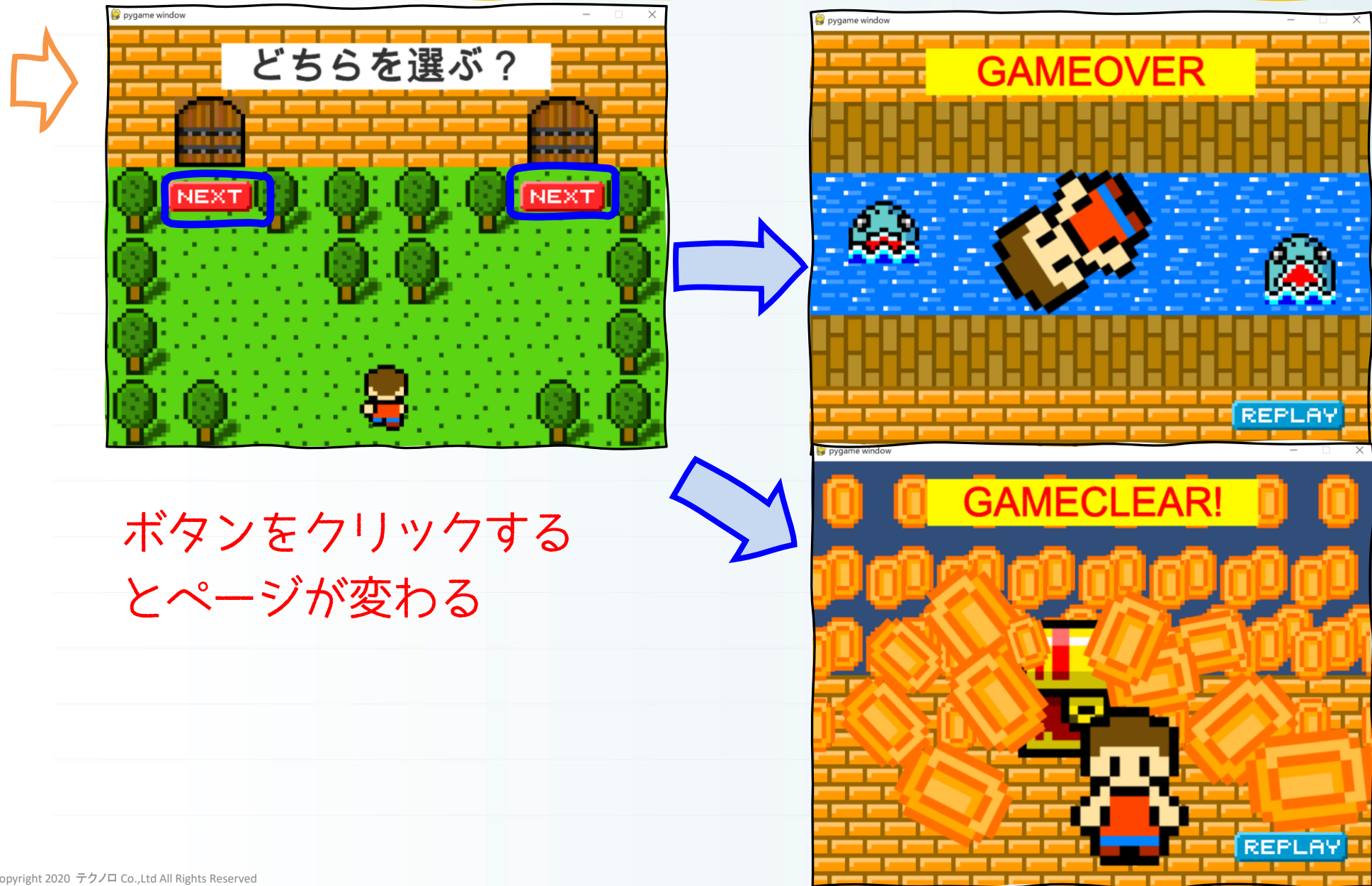
```
    if event.type == pg.QUIT:
```

```
        pg.quit()
```

```
        sys.exit()
```

画面の切り替えで紙芝居

「Run Module」またはF5キーを押してプログラムを実行する。



画面の切り替えで紙芝居

選択画面を増やして、2回枝分かれするプログラムを作る。

#ゲームの準備をする

```
import pygame as pg, sys
```

```
pg.init()
```

```
screen = pg.display.set_mode((800, 600))
```

#紙芝居

```
img1 = pg.image.load("images/root1.png")
```

```
img2 = pg.image.load("images/root2.png")
```

```
img3 = pg.image.load("images/root3.png")
```

```
img4 = pg.image.load("images/root4.png")
```

```
img5 = pg.image.load("images/root5.png")
```

#ボタン

```
next_img = pg.image.load("images/nextbtn.png")
```

・追加修正
する箇所

次のページに続く

画面の切り替えで紙芝居

```
replay_img = pg.image.load("images/replaybtn.png")
```

```
pushFlag = False
```

```
#btnを押したら、newpageにジャンプする
```

```
def button_to_jump(btn, newpage):
```

```
    global page, pushFlag
```

```
    #ユーザーからの入力を調べる
```

```
    mdown = pg.mouse.get_pressed()
```

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

```
    if mdown[0]:
```

```
        if btn.collidepoint(mx, my) and pushFlag == False:
```

```
            page = newpage
```

```
            pushFlag = True
```

```
    else :
```

```
        pushFlag = False
```

次のページに続く

画面の切り替えで紙芝居

```
def page1():
```

```
    #画面を初期化する
```

```
    screen.blit(img1, (0,0))
```

```
    btn1 = screen.blit(next_img,( 90, 220))
```

```
    btn2 = screen.blit(next_img,(590, 220))
```

```
    #絵を描いたり、判定したりする
```

```
    button_to_jump(btn1, 2)
```

```
    button_to_jump(btn2, 3)
```

```
def page2():
```

```
    #画面を初期化する
```

```
    screen.blit(img2, (0,0))
```

```
    btn1 = screen.blit(replay_img,(600, 520))
```

```
    #絵を描いたり、判定したりする
```

```
    button_to_jump(btn1, 1)
```

次のページに続く

画面の切り替えで紙芝居

```
def page3():
```

```
    #画面を初期化する
```

```
    screen.blit(img3, (0,0))
```

```
    btn1 = screen.blit(next_img,(190, 320))
```

```
    btn2 = screen.blit(next_img,(490, 320))
```

```
    #絵を描いたり、判定したりする
```

```
    button_to_jump(btn1, 4)
```

```
    button_to_jump(btn2, 5)
```

```
def page4():
```

```
    #画面を初期化する
```

```
    screen.blit(img4, (0,0))
```

```
    btn1 = screen.blit(replay_img,(600, 520))
```

```
    #絵を描いたり、判定したりする
```

```
    button_to_jump(btn1, 1)
```

・ ・ 追加修正
する箇所

次のページに続く

画面の切り替えで紙芝居

```
def page5():
```

```
    #画面を初期化する
```

```
    screen.blit(img5, (0,0))
```

```
    btn1 = screen.blit(replay_img,(600, 520))
```

```
    #絵を描いたり、判定したりする
```

```
    button_to_jump(btn1, 1)
```

・追加修正
する箇所

```
page = 1
```

```
#この下をずっとループする
```

```
while True:
```

```
    if page == 1:
```

```
        page1()
```

```
    elif page == 2:
```

```
        page2()
```

```
    elif page == 3:
```

次のページに続く

画面の切り替えで紙芝居

```
page3()
```

```
elif page == 4:
```

```
    page4()
```

```
elif page == 5:
```

```
    page5()
```

・追加修正
する箇所

```
#画面を表示する
```

```
pg.display.update()
```

```
pg.time.Clock().tick(60)
```

```
#閉じるボタンが押されたら、終了する
```

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

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

```
        pg.quit()
```

```
        sys.exit()
```

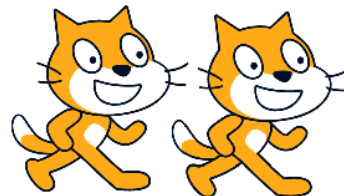
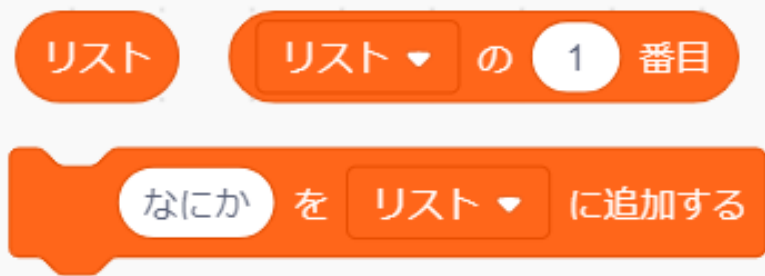

する。



復習 & チャレンジ

ここまで習ったことをScratchでもできるかチャレンジしてみよう。

その過程でScratchでできること、Pythonでないとできないことを整理してみよう。



メモ



プログラミング教室の テクノロ

なまえ：