

# Python勉強会

## 6歩目

チャンスラボ株式会社 岡本



CoderDojo太宰府

## 0. はじめに。

ネットに繋いで情報取得するので、

以下のWi-Fi設定をしてください。

**Python\_Lab**

**20200123**

# 1. ゲーム開発のプログラム基礎知識

今回はゲームを作る際に必要なプログラムの基礎知識を勉強していきます。

今回は「当たり判定」



★解らないときは周りに聞いてください。仲魔が増えます。



## 2. 当たり判定とは

ある物体が別の物体に当たったか（衝突したか）どうかを判定するプログラム処理のこと。

引用元: ウィキペディア



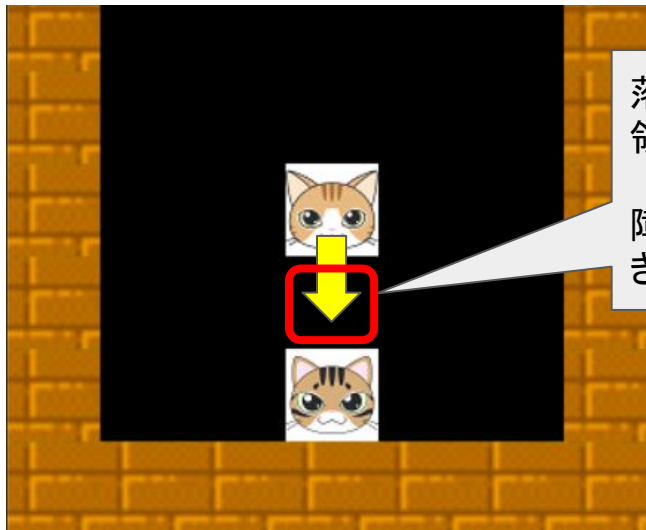
実際の処理では、進行方向に障害物があるかどうかを判定します。

車の自動ブレーキでも同様の処理を行っている（多分）



### 3. 実際の使われ方は？

以前紹介した落ち物パズルゲームの場合。



落ちてくる猫ブロックの進行方向の一つ下の領域を当たり判定チェックする。

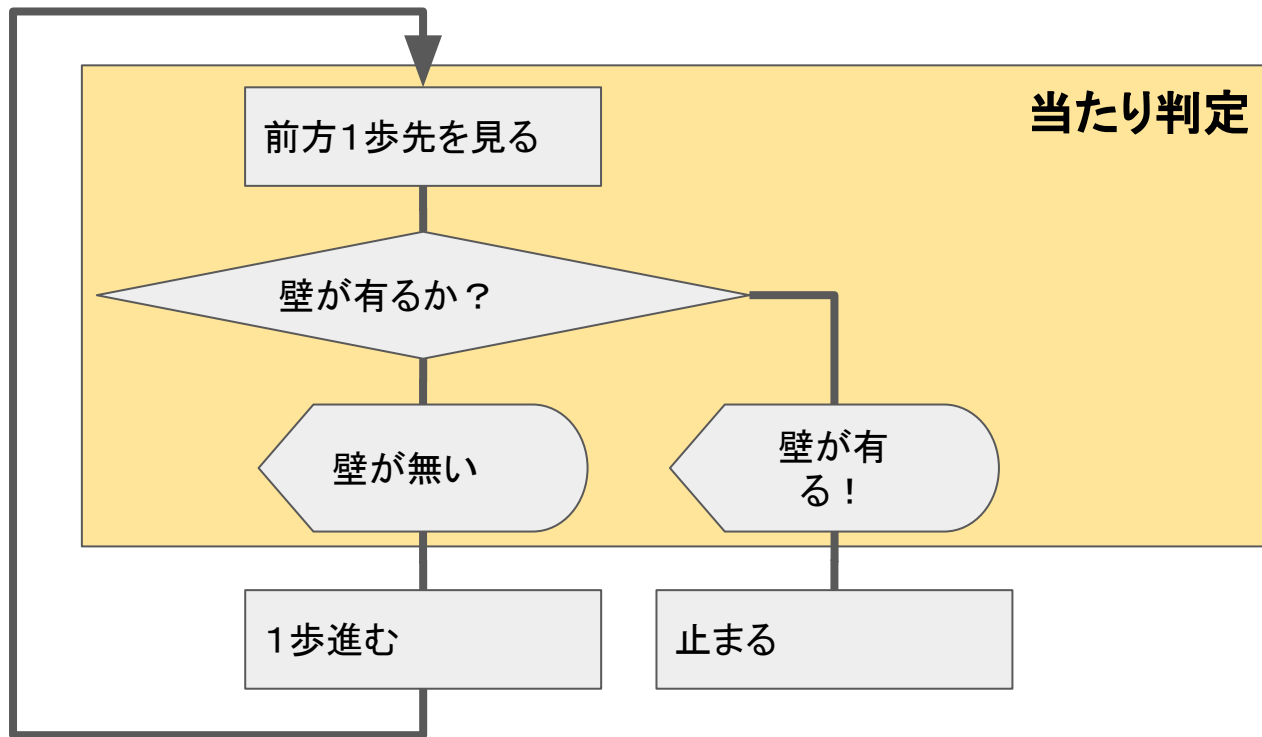
障害物が無ければひとつ下の領域に移動できる。

左右に移動する際には横方向の当たり判定を行う必要がある。

※py\_game.pyでは実装していない。

## 4. 当たり判定の考え方

歩く際に進行方向に障害物があるかどうかを判定する場合は？



## 5. 実装してみよう 一次元

先ず、配列の指定位置が0もしくはそれ以外かを判定してbool型で返却する関数を作ってください。 ※bool型のオブジェクトTrueとFalseで表される。(真:True、偽:False)

if文などの条件式で使用する。

col\_test.py

```
def func_collision(arr_map, idx) -> bool:
```

```
    ret_collision:bool = False
```

```
    #処理
```

```
    return ret_collision
```

「#処理」の部分にプログラムで埋めてください。  
複数行でOK

```
def main():
```

```
    COL_MAP:list = [0,1,2,3,0,5]
```

```
    print(func_collision(COL_MAP, 0))
```

```
if __name__ == "__main__":
```

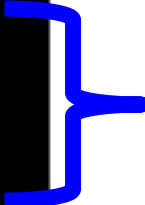
```
    main()
```

## 6. 実装してみよう 二次元

先程のソースを参考して、二次元配列の COL\_MAP から指定の位置を取り出せるように、関数 func\_collision をプログラムしてください。

col\_test.py

```
def main():  
    COL_MAP:list = [[0,1,2,3],[0,4,0,5],[6,7,8,0]]  
    print(func_collision(COL_MAP, 1,2))  
  
if __name__ == "__main__":  
    main()
```



この2行は  
修正して使う



## 7. 実際のソースコードを見てみよう

落ち物パズルゲームのソースコード「py\_game.py」の中では以下の様に処理しています。

### 呼び出し部分

```
x1_pt, y1_pt = event_proc(x1_pt, y1_pt)      # キー入力など、イベント処理
y1_pt = y1_pt + 50                            # ブロック落下
ret_collision: bool = chek_collision(int(x1_pt / 50), int(y1_pt / 50), list0)    # 当たりチェック
if ret_collision:
    y1_pt = y1_pt - 50 # 衝突。落下できないのでY座標を戻す
```

### 当たり判定関数

```
## ブロックの当たり判定 ※簡易なので下しか見ない。
def chek_collision(x, y, list0: list) -> bool:
    block = list0[y][x]
    if block == 0:
        return False
    else:
        return True
```

# 今回使用した資料について

今回使用した資料等は、以下のgithubに置いていきますので  
ダウンロードして使用可能です。

[https://github.com/okamotomasatosi/py\\_doc](https://github.com/okamotomasatosi/py_doc)

