

```
# -*- coding:utf-8 -*-
#
# written by Shotaro Fujimoto
# 2016-07-12
```

Modules

- [matplotlib.animation](#)  
[numpy](#)
- [operator](#)  
[matplotlib.pyplot](#)
- [pprint](#)  
[matplotlib.tri](#)

Classes

[base.Main](#)

[SAW](#)

class **SAW**([base.Main](#))  
2次元三角格子上の自己回避ランダムウォーク

Methods defined here:

**`__init__`**(self, Lx=40, Ly=40, boundary={'h': 'periodic', 'v': 'periodic'}, size=[5, 4, 10, 12], plot=True, plot\_surface=True, save\_image=False, save\_video=False, filename\_image="", filename\_video="", frames=1000, beta=2.0, interval=1, weight\_const=0.5, strings=None, pre\_function=None, post\_function=None)  
Init function of [Main](#) class.

Lx (int (even)): 格子のx方向(グラフではy軸)の格子点数  
Ly (int (even)): 格子のy方向(グラフではx軸)の格子点数

**`cleanup_bonding_pairs`**(self, key, index\_start, index\_stop)

**`dot`**(self, v, w)  
0〜5で表された6つのベクトルの内積を計算する。  
  
v, w (int): ベクトル(0〜5の整数で表す)

**`get_bonding_pairs`**(self, s, index\_start, index\_stop)

**`get_neighbor_xy`**(self, key)  
Stringクラスのインスタンスsの隣接する非占有格子点の座標を取得する  
  
s (String): 対象とするStringクラスのインスタンス

**`plot_all`**(self)  
軸の設定, 三角格子の描画, 線分描画要素の用意などを行う  
  
ここからFuncAnimationを使ってアニメーション表示を行うようにする

**`plot_string`**(self)  
self.strings内に格納されているStringを参照し, グラフ上に図示する

**`start_animation`**(self, filename="")

**`update`**(self, num=0)  
FuncAnimationから各フレームごとに呼び出される関数  
  
1時間ステップの間に行う計算はすべてここに含まれる。

**`update_each_string`**(self, key)

Methods inherited from [base.Main](#):

**`create_random_strings`**(self, N=3, size=[10, 5, 3])  
Create N strings of each size is specified with 'size'.  
  
This process is equivalent to self-avoiding walk on triangular lattice.

**`get_next_xy`**(self, x, y, \*vec)

```
pp = <pprint.PrettyPrinter instance>
```

```
print_function = _Feature((2, 6, 0, 'alpha', 2), (3, 0, 0, 'alpha', 0), 65536)
```