

# Reactor 初期検討

```
In [1]: from datetime import datetime  
from Reaction import *
```

## 基本動作確認

まずはSeedとなるReactionを作成。

太郎君が「今日いい天気だね、どう思う？」と議題を提出。

次郎くんがすかさず同意する。

```
In [2]: a = Reaction(0, "Taro", None, "It's fine today, don't you think so?", EVAL_NEUT)  
b = Reaction(1, "Jiro", a, "Year, good weather", EVAL_PROS)
```

```
In [3]: print a.reactionFactor  
print b.reactionFactor
```

2.0736

1.44

引用を受けた太郎くんのRF : Reaction Factorが次郎君より高い。

今度は三郎君が異議を唱える。

```
In [4]: c = Reaction(2, "Saburo", a, "I don't think so", EVAL_CONS)  
print a.reactionFactor  
print c.reactionFactor
```

4.1472

1.44

分からぬ、という四郎君がしゃしゃり出てくる。

```
In [5]: d = Reaction(3, "Shiro", a, "I don't know, coz I'm inside whole day", EVAL_NEUT)  
print a.reactionFactor  
print d.reactionFactor
```

5.5872

1.2

反対意見でも、分からぬという意見でも、太郎君のSeedのReaction Factorはぐんぐん上昇。

寡黙な五郎君が黙って太郎くんに反対する。

```
In [6]: e = Reaction(4, "Goro", a, "", EVAL_CONS)
```

```
In [7]: print a.reactionFactor  
print e.reactionFactor
```

```
7.0272  
1.2
```

五郎君の無言の異議でも太郎くんのReaction Factorは上昇しているのがわかる。

太郎君の発言に対するReaction Listを列挙してみる。

```
In [8]: for r in a.reactionList:  
    print r.strFormat()
```

```
Jiro : Year, good weather @ 2015/02/08 20:30:59 (1.44)  
Saburo : I don't think so @ 2015/02/08 20:30:59 (1.44)  
Shiro : I don't know, coz I'm inside whole day @ 2015/02/08 20:30:59 (1.20)  
Goro : @ 2015/02/08 20:30:59 (1.20)
```

## 多階層Reaction

こんどは、異議を唱えた三郎君に花子さんが同調する。

```
In [9]: f = Reaction(5, "Hanako", c, "me too.", EVAL_PROS )
```

```
In [10]: print a.reactionFactor  
print c.reactionFactor
```

```
7.939584  
2.0736
```

花子さんが同意したことで、太郎君、三郎君のRFも上昇している。

## Reaction Chain

花子さんの発言をさかのぼり、太郎君→三郎君→花子さんの発言を全て表示してみる。

```
In [11]: chain = f.getReactionChain([])
```

```
In [12]: for r in reversed(chain):  
    print r.strFormat()
```

```
Taro : It's fine today, don't you think so? @ 2015/02/08 20:30:59 (7.94)  
Saburo : I don't think so @ 2015/02/08 20:30:59 (2.07)  
Hanako : me too. @ 2015/02/08 20:30:59 (1.44)
```

# 解析

## Reaction List解析

太郎くんが作成したSeedに対するReaction Listを、反応毎に整理してみる

```
In [15]: dict = {}
dict["Pros"] = []
dict["Cons"] = []
dict["Neutral"] = []
for reaction in a.reactionList:
    if reaction.evaluation == EVAL_PROS:
        dict["Pros"].append(reaction)
    if reaction.evaluation == EVAL_CONS:
        dict["Cons"].append(reaction)
    if reaction.evaluation == EVAL_NEUT:
        dict["Neutral"].append(reaction)

print "Seed ---"
print a.strFormat()
print

for key in dict:
    print key, "---"
    for r in dict[key]:
        print r.strFormat()
    print
```

Seed ---

Taro : It's fine today, don't you think so? @ 2015/02/08 20:30:59 (7.94)

Neutral ---

Shiro : I don't know, coz I'm inside whole day @ 2015/02/08 20:30:59 (1.20)

Cons ---

Saburo : I don't think so @ 2015/02/08 20:30:59 (2.07)

Goro : @ 2015/02/08 20:30:59 (1.20)

Pros ---

Jiro : Year, good weather @ 2015/02/08 20:30:59 (1.44)

次は、太郎くんへのReactionをRFの高い順にソートしてみる

```
In [14]: sortedReactions = sorted( a.reactionList, key=lambda reaction: reaction.reactionFact
or, reverse=True )
for i, r in enumerate(sortedReactions):
    print i, r.strFormat()
```

0 Saburo : I don't think so @ 2015/02/08 20:30:59 (2.07)

1 Jiro : Year, good weather @ 2015/02/08 20:30:59 (1.44)

2 Shiro : I don't know, coz I'm inside whole day @ 2015/02/08 20:30:59 (1.20)

3 Goro : @ 2015/02/08 20:30:59 (1.20)

