

Pythonでリファクタリング… 5つのパターンから基本を学ぶ

土屋 健

リスト1 マジック・ナンバを定数にして意味を明確にする

```

1 # ケース1
2
3 from datetime import datetime, date
4
5 def case1():
6     print('=== Start case1 ===')
7
8     # 現在日時を取得
9     dt = datetime.now()
10
11     if dt.month == 10 and dt.day == 10:
12         print('誕生日おめでとう')
13
14     if dt.month == 1 and dt.day == 1:
15
16         print('あけましておめでとう')
17
18         if 6 <= dt.hour <= 8:
19             print('おはよう')
20
21         if 22 <= dt.hour <= 24:
22             print('おやすみ')
23
24         print('=== Finish case1 ===')
25
26 # メイン
27 if __name__ == "__main__":
28     case1()

```

(a) ビフォー

```

1 # ケース1 修正版
2
3 from datetime import datetime, date
4
5 def case1_mod():
6     print('=== Start case1 MOD ===')
7
8     # メッセージを出力する条件を定義
9     ## 元旦の定義
10    NEW_YEARS_DAY = [1,1]
11    ## 誕生日の定義
12    BIRTH_DAY = [10, 10]
13    ## 起床時間帯の定義
14    GOOD_MORNING = [6,8]
15    ## 就寝時間帯の定義
16    GOOD_NIGHT = [22,24]
17
18    # 現在日時を取得
19    dt = datetime.now()
20
21    if [dt.month, dt.day] == BIRTH_DAY:
22        print('誕生日おめでとう')
23
24    if [dt.month, dt.day] == NEW_YEARS_DAY:
25        print('あけましておめでとう')
26
27    if GOOD_MORNING[0] <= dt.hour <=
28        GOOD_MORNING[1]:
29        print('おはよう')
30
31    if GOOD_NIGHT[0] <= dt.hour <= GOOD_NIGHT[1]:
32        print('おやすみ')
33
34    print('=== Finish case1 MOD ===')
35
36 # メイン
37 if __name__ == "__main__":
38     case1_mod()

```

(b) アフター

● Pythonでさまざまなリファクタリングを見てみよう

ひと口にリファクタリングと言っても、変数名を変更する簡単なものから、処理順序を変えたり、モジュール構成を変えたりする、大きなものまでさまざまです。ここでは、Pythonのプログラムを使ってリファクタリングではどのようなことを行うのか説明します。

技1 マジック・ナンバを定数にして意味を明確にする

特定の値の場合の処理を記述する場合に、値を直接プログラムに埋め込む場合があります。そういった値は見ただけでは意味が分かりにくいので、意味を明確にするために定数定義値を使うのがよいです。

ここでは、マジック・ナンバを定数定義します。