

開発者が行ったソースコード修正作業を学習し代行するボット ー DevReplay : 開発者を再現するLint ー

上田 裕己 (奈良先端科学技術大学院大学)

DevReplayは過去の修正方法を学習,
再現することで修正コストを減らします
<https://devreplay.github.io/>

```
Target.py
1 ary = list([1, 2, 3])
2 if not sum(ary):
3     print("Hello!")
4
5
6
7
8 tmp = a
9 a = b
10 b = tmp
11
```

```
{ devreplay.json > { } 1 > [ ] condition
1 [
2   {"condition": ["not sum($1)",
3     "consequent": ["sum($1) == 0"]},
4   {
5     "condition": [
6       "tmp = $1",
7       "$1 = $2",
8       "$2 = tmp"],
9     "consequent": [
10      "$2, $1 = $1, $2"],
11     "description": "Pythonなら一行で書けます",
12     "severity": "Info",
13     "author": "Yuki"}
14 ]
```

Pythonなら一行で書けます by Yuki devreplay(1)

Peek Problem Quick Fix...

PROBLEMS 2 OUTPUT ... Filter. E.g.: text, **/*.ts, !**/node_modules/**

Target.py 2

- not sum(\$1) should be sum(\$1) == 0 devreplay(0) [2, 1]
- Pythonなら一行で書けます by Yuki devreplay(1) [8, 1]

修正前

```
1 ary = list([1, 2, 3])
2 if not sum(ary):
3     print("Hello!")
4
5 a = 1
6 b = 2
7
8 tmp = a
9 a = b
10 b = tmp
11
```

修正後

```
1 ary = list([1, 2, 3])
2 if sum(ary) == 0:
3     print("Hello!")
4
5 a = 1
6 b = 2
7
8 b, a = a, b
9
```

学習パターン例:

- 曖昧な書き方の修正
- 複数行のコードを一行に書き換える

ほかにも

- ベテランのコーディングスタイル
- チーム内のローカルルール など

DevReplayを使って開発者のコーディングスタイルを学習することで
達人プログラマーの技術を誰でも再現できます



パターン学習

好きな開発者やプロジェクトからスタイルを学習します

スクリプト: <https://github.com/devreplay/devreplay-pattern-generator>

修正

パターン通りにソースコードを自動修正します

プラグイン: <https://marketplace.visualstudio.com/items?itemName=Ikuyadeu.devreplay>



拡張・改善

作ったパターンやサンプルは自分に合わせて拡張・改善できます

サンプル: <https://devreplay.github.io/rules.html>

共有

チームで共有してGitHub上で自動コードレビューが行えます

GitHub App: <https://github.com/marketplace/dev-replay>



対応言語



COBOL