

Pythonでワンライナー

2019/03/11 KMC LT大会 @utgwkk

utgw (at) kmc.gr.jp

Pythonでワンライナー

- Pythonの文法の特徴
- ワンライナーでもできること/できないこと
- 関数を定義する
- 関数内で変数を定義する
- 繰り返し処理をする
- 複数の文をつなげる

Pythonの文法の特徴

- いろいろあるけど今日はこれだけ押さえてね
- オフサイドルール
 - インデントの深さによってプログラムの構造を表現

オフサイドルール実例

```
for i in range(1, 30+1):  
    if i % 15 == 0:  
        print( 'FizzBuzz' )  
    elif i % 5 == 0:  
        print( 'Buzz' )  
    elif i % 3 == 0:  
        print( 'Fizz' )  
    else:  
        print(i)
```

ワンライナーでもできること

- 変数への値の代入ができる
- 三項演算子が見える
- print関数が見える
- セミコロンで文をつなげることができる
- `x = 1; y = 2; print(x + y)`

ワンライナーでできないこと

- オフサイドルールが必要なものはだいたい無理と思って挑むことになる
- 制御構造文の組み合わせはだいたい無理

これは無理(再掲)

```
for i in range(1, 30+1):  
    if i % 15 == 0:  
        print( 'FizzBuzz' )  
    elif i % 5 == 0:  
        print( 'Buzz' )  
    elif i % 3 == 0:  
        print( 'Fizz' )  
    else:  
        print(i)
```

関数を定義する

- `def` を使うと一見いけそうに見える
- やったか! ?  やってない
- 例: `def f(x): return x + 1; print(f(1))`

顧客が求めていたもの

```
def f(x):  
    return x + 1
```

```
print(f(2))
```

プログラマの実装

```
def f(x):  
    return x + 1  
print(f(2))
```

lambda

- `lambda` を使う
 - `lambda` の中には式しか書けない
 - 例: `f = lambda x: x + 1; print(f(2))`
- 再帰関数も定義できる
 - 例: `fac = lambda x: 1 if x <= 1 else x * fac(x - 1)`

関数内で変数を定義する

- `lambda` の中には式しか書けない
- 変数定義がしたくなっても書けない気がする

変数定義と関数適用をじっと 見つめる

- $x = 1; M \rightarrow M[x := 1]$
- $(\text{lambda } x: M)(1) \rightarrow M[x := 1]$
- オッ, β 簡約では??????????
- 操作的意味論の気持ちになって考える

関数内で複数の式をつなげる

- タプルの要素が左から順に評価されることを利用
- 例: `f = lambda x: (print(x), print(x+1), print(x+2)); f(1)`

繰り返し処理

- これができればもう優勝同然
- for文はトップレベルでしか使えない
- トップレベル以外では再帰関数を使う
 - スタックオーバーフローに注意
 - 末尾再帰最適化たぶんないと思う
 - あったら教えてください

プログラムの例

- FizzBuzz

FizzBuzz

```
for i in range(1, 40+1): print('FizzBuzz') if
i % 15 == 0 else print('Buzz') if i % 5 == 0
else print('Fizz') if i % 3 == 0 else
print(i)
```

FizzBuzz

```
for i in range(1, 40+1):  
    print('FizzBuzz') if i % 15 == 0  
    else print('Buzz') if i % 5 == 0  
    else print('Fizz') if i % 3 == 0  
    else print(i)
```

感想

- 三項演算子の形に違和感
- if b then M else N って書いててあれっ？ となっていた
- 末尾再帰最適化なんてなかった
- オフサイドルール無視してやっていくのはほぼ不可能
- 関数型でないプログラミング言語で無理をしない

ここまで書いて気づいた

- `exec()` 使えば原理上なんでもワンライナーにできるので
は????????????
- <https://docs.python.org/ja/3/library/functions.html#exec>
- おわり