

Summary: このドキュメントは、Piscine 101 @ 42Tokyoの PYTHON 02モジュール 用の課題である。

Contents

1	Instructions	2
II	Foreword	3
III	Exercise 00: Buy this product!	4
IV	Exercise 01 : Be more specific!	6
V	Exercise 02 : Sell them now.	8
VI	Exercise 03: Sell them more.	10
VII	Exercise 04: Can't be infinite.	12
VIII	Exercise 05: I am your manager.	15
\mathbf{IX}	Exercise 06: Bonus	18

Chapter I

Instructions

- 課題に関する噂に惑わされないよう気をつけ、信用しないこと。
- この書類は、提出前に変更になる可能性があるため、気をつけること。
- ファイルとディレクトリへの権限があることを、あらかじめ確認すること。
- すべての課題は、提出手順に従い行うこと。
- 課題の確認と評価は、あなたの周りにいるPiscine101受験者により行われる。
- 課題は、簡単なものから徐々に難しくなるように並べられている。
- 課題で指定されていないものは、 <u>どんなファイルも</u>ディレクトリ内に置かない こと。
- 助けてくれるのは、Google / 人間 / インターネット / ...と呼ばれているものたちである。
- 出力例には、問題文に明記されていない細部まで表示されている場合があるため、入念に確認すること。
- 各問題でPythonのバージョンの指定がない場合は、次のバージョンを使用すること。: Python python3.9.0
- "使用可能な標準ライブラリ"に記載されているライブラリやあなたが作成したモジュールのみを、インポートすること。
- 使用が禁止されている関数はありません。
- guacamole.42tokyo.jp上で課題に取り組むこと。

Chapter II

Foreword

Brendan : Hi, you there! Mysterious stranger! Yoohoo!

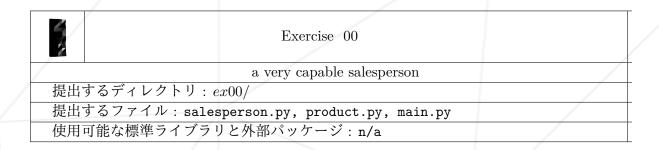
Brendan : Year, you! There any other mysterious stranger around here?

Brendan : You know what? How 'bout I tell you a joke?

V : Sure, be my guest.

Chapter III

Exercise 00: Buy this product!



あなたは世界で一番効果が高いエナジードリンクを開発しました。さっそくこの 商品を販売しましょう。

• 以下の例を参考にして、 docstringに記載されている挙動を行うProduct クラスをproduct.pyに実装せよ。

例)

```
class Product:
    def __str__(self):
        """
        return a string such as 'This is the best product in the world.'
        """
        pass
```

1人で商品を販売するのは大変なため、販売員を雇います。

• 以下の例を参考にして、 docstringに記載されている挙動を行うSalesperson クラスをsalesperson.pyに実装せよ。

例)

```
class Salesperson:

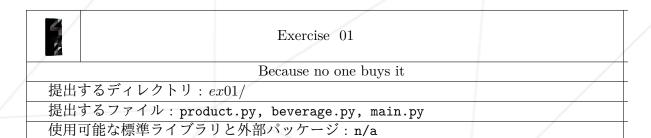
def __init__(self, name='nameless salesperson.'):
    """
    assign its value to a name attribute.
    """
    pass

def __str__(self):
    """
    return name attribute of the instance.
    """
    pass
```

```
def promote(self):
    """
    return a Product instance.
    """
    pass
```

Chapter IV

Exercise 01: Be more specific!



あなたはこの商品のキャッチコピーを決めましたが、価格が決まっていません。 この商品の価格を決めましょう。

• 以下の例を参考にして、 docstringに記載されている挙動を行うProduct クラスをproduct.pyに実装せよ。

例)

• super関数を使用して、docstringに記載されている挙動を行う、以下のようなProductクラスを継承するBeverageというクラスを、beverage.pyに実装せよ。

```
class Beverage(Product):
    def __init__(self, name, price, description, temperature):
        """
        assign its values to name, price, description, temperature attribute, using super(),
        """
        pass

def print_attr(self):
        """
        return its name, price, description and temperature attributes in the format.
        name : <name attribute>
        price : <price attribute limited to two decimal points>
        description : <description attribute>
        temperature : <temperature attribute>
        """
        pass
```

Chapter V

Exercise 02: Sell them now.



Exercise 02

Brendan the Vending Machine

提出するディレクトリ: ex02/

提出するファイル: product.py, beverage.py, vendingmachine.py, main.py 使用可能な標準ライブラリと外部パッケージ: n/a

さらに利益を上げるために、この商品を自動販売機に設置しましょう。

- 前問のbeverage.pyとproduct.pyを使用すること。
- 以下の例を参考にして、docstringに記載されている挙動を行うVendingmachine クラスをvendingmachine.pyに実装せよ。
 例)

```
class Vendingmachine:
   def __init__(self, name, stock):
       taking a string, a list of Beverage instances as parameters,
       assign its value to a name attribute and a stock attribute.
       pass
   def __str__(self):
       return its name attributes in the format.
       <name attribute> the vending machine
       pass
   def sell(self, beverage_name):
       taking a string as a parameter, print a message of your choice.
       the message differ when there is a Beverage instance
       named same as beverage_name in the stock, or not.
       pass
   def ask(self):
       print a message to ask the beverage name,
       store the user input,
       then call sell method with the users input as parameter.
       pass
```

Piscine 101

Python 02

• askメソッドの出力例1)

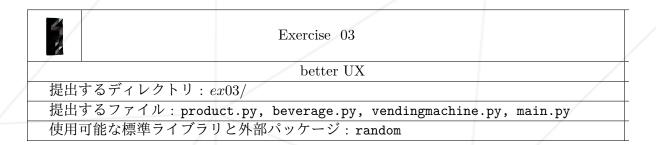
Hello, what would you like? > coffee
Here is your coffee!

• askメソッドの出力例2)

Hello, what would you like?
> milk shake
Sorry! I do not have milk shake...

Chapter VI

Exercise 03: Sell them more.



さらに利益を上げるために、自動販売機のUXを改善しましょう。

- 前問のbeverage.pyとproduct.pyを使用すること。
- 以下の例を参考にして、docstringに記載されている挙動を行うVendingmachine クラスをvendingmachine.pyに実装せよ。

```
class Vendingmachine:
   def __init__(self, name, greeting, stock):
       taking 2 string, a list of Beverage instances as parameters,
       assign its value to a name attribute, a greeting attribute and a stock attribute.
       pass
   def greet(self):
       print a greeting attribute of the instance.
       pass
   def display(self):
       show its lineup of stocks.
       pass
   def recommend(self):
       choose a random instance from its stock,
       then print a recommendation sentence using the name attribute of the selected Beverage
       ....
       pass
```

前問の str メソッドを追加すること。

def __str__(self):

```
return its name attributes in the format.
<name attribute> the vending machine
"""
pass
```

• 前問のask メソッドを追加すること。

```
def ask(self):
    """

print a message to ask the beverage name,
    store the user input,
    then call sell method with the users input as parameter.
    """
pass
```

• 前問のsellメソッドを、以下の例を参考にして書き換えよ。

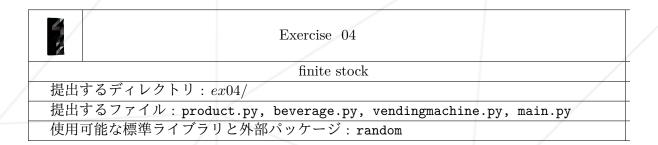
```
def sell(self, beverage_name):
    """
    print a message of your choice.
    the message differ when there is a Beverage instance
    named same as beverage_name in the stock, or not
    call recommend method
    when the given beverage name is not found in its stock.
    """
    pass
```

• Henry the vending machineの askメソッドの出力例)

```
would you like something?
> matcha
sorry, I don't have the product called matcha
I recommend...
coca-cola : the most popular drink!
```

Chapter VII

Exercise 04: Can't be infinite.



自動販売機は、商品在庫を確認せずに商品を販売する仕様になっています。これ を修正しなければなりません。さっそく修正に取りかかりましょう。

- 前問のbeverage.pyとproduct.pyを使用すること。
- 以下の例を参考にして、docstringに記載されている挙動を行うVendingmachine classを書き換えよ。
- __init__メソッド:2つの文字列と辞書のリストを受け取ること。辞書には、 2つのキーと値のペアが含まれています:
 - 。 'product': Beverage インスタンス
 - 。 'amount': 整数 (飲み物の在庫の量)

これらを、各インスタンス変数(name, greeting, stock)に代入すること。

例えば、 Vendingmachine インスタンスを以下のように使用する。

• 前問のすべてのメソッドを追加せよ。

● sellメソッド:前問の条件に、以下の条件を加えること。 商品の在庫があるときのみ、その商品を販売することができ、販売されるたび に在庫を1つ減らす。 • add_stockメソッド:1つの文字列(beverage_name)と整数(number)を受け取ること。beverage_nameに代入された値と同じ値を持つBeverageインスタンスの、nameインスタンス変数が存在する場合、在庫をnumberの個数分増やすこと。存在しない場合は、その名前を持つBeverageインスタンスを、numberの個数分、新しくstockに追加すること。price, description, temperatureなどのインスタンス変数の値は任意とする。

```
def add_stock(self, beverage_name, number):
    """
    taking a string and integer as parameters,
    increase the amount of that drink in stock.
    when the Beverage instance with the same name as beverage_name exists in the stock.
    if not, create a new Beverage instance and add it into its stock with the number
    """
    pass
```

Chapter VIII

Exercise 05: I am your manager.



Exercise 05

mangers manage their stock

提出するディレクトリ: ex05/

提出するファイル: product.py, beverage.py, vendingmachine.py, main.py

使用可能な標準ライブラリと外部パッケージ: random

商品を補充するために、自動販売機に管理者モードを設定しましょう。

- 前問のbeverage.pyとproduct.pyを使用すること。
- 以下の例を参考にして、Vendingmachine classを書き換えよ。
- 前問のすべてのメソッドを追加すること。
- __init__メソッド:前問のパラメータに新たな変数(文字列)を追加し、その値をadmin_codeインスタンス変数に代入すること。

```
class Vendingmachine:
    def __init__(self, name, greeting, stock, admin_code):
        """
        taking 3 string, a list of dictionaries {'product': a Beverage instance, 'amount':
            integer} as parameters,
        assign its value to a name attribute, a greeting attribute a stock attribute, and
            admin_code attribute.
        """
        pass
```

例)

• askメソッド:前間の条件に、以下の条件を加えること。 "admin mode"の文字 列を、beverage_nameの代わりに入力すると、admin_mode関数を呼び出す。

```
def ask(self):
    """

call display method and print a message to ask the beverage name,
    store the user input,
    then call sell(beverage_name) function.
```

```
if "admin mode" is inputted instead of beverage name,
start its admin mode, calling a method admin_mode.
"""
pass
```

• show_stockメソッド:出力するフォーマットは任意とする。

```
def show_stock(self):
    """
    displays all the stocked beverage with each number of stock in the form of your choice.
    """
    pass
```

• admin_modeメソッド:以下に記載されている管理者モードの例とdocstringを基 に、このメソッドを実装すること。出力する文章は任意とする。

```
def admin_mode(self):
    """
    prompt 'Please tell me the admin code:' to the user, wait for the user's input
    and exit if the user's input doesn't match the admin_code attribute.
    If the user's input match with admin_code attribute,
    prompt a name of beverage and the number of stock to add.
    Then, call the previously implemented add_stock method.
    If "exit" is inputted, then it will end admin_mode.
    If "show stock" is inputted instead of a beverage name,
    this method will call another method show_stock instead.
    """
    pass
```

• Henry the vending machine の管理者モードの例)

```
would you like something?
> admin mode
Please tell me the admin code:
> I am your admin.
Do you want to show stock or add a stock? Type "show stock" to show my stock or a beverage name
    to add stock.
> show stock
Here is my current stock:
coffee : It is a must for brainwork! ---> 2 left
coca-cola : the most popular drink! ---> 5 left
Do you want to show stock or add a stock? Type "show stock" to show my stock or a beverage name
    to add stock.
> coca-cola
How many stock do you want to add?
Do you want to show stock or add a stock? Type "show stock" to show my stock or a beverage name
    to add stock.
> show stock
Here is my current stock:
coffee : It is a must for brainwork! ---> 2 left
coca-cola : the most popular drink! ---> 10 left
Do you want to show stock or add a stock? Type "show stock" to show my stock or a beverage name
    to add stock.
> AAAAA
How many stock do you want to add?
```

```
Done.

Do you want to show stock or add a stock? Type "show stock" to show my stock or a beverage name to add stock.

> show stock

Here is my current stock:
coffee : It is a must for brainwork! ---> 2 left
coca-cola : the most popular drink! ---> 10 left
AAAAAA : no description ---> 5 left

Do you want to show stock or add a stock? Type "show stock" to show my stock or a beverage name to add stock.

> coca-cola

How many stock do you want to add?

> A

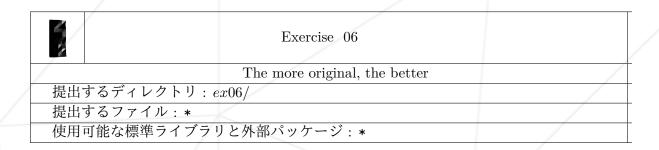
Not a valid input.

Do you want to show stock or add a stock? Type "show stock" to show my stock or a beverage name to add stock.

> exit
Bye !
```

Chapter IX

Exercise 06: Bonus



商品の販売利益が上がるような、独創的なvending machine クラスを実装しましょう。例えば、以下のリストから機能を選び実装することができます。そのほかに創造的な機能を実装しても問題ありません。

- 前問のbeverage.pyとproduct.pyを使用すること。
- 新たな機能を実装する際、ほかの機能に影響を与えないことを確認する、unittestを使用したテスト。(https://docs.python.org/ja/3/library/unittest.html)
- recommendメソッドの最適化。(例:在庫の多いものをおすすめする。)
- 販売された商品の価格を計算し、収益を記録する機能の実装。
- sayメソッド:人間らしい口調のvendingmachineにする。(0.1秒に1文字を出力する)
- lotteryメソッド: 当たりが出たら、無料で商品を渡すシステムの実装。
- vendingmachineのヴィジュアルの実装。
- 各機能の実装をテストする、main.pyのファイルをGitに提出すること。提出されたmain.pyは、レビュー中に確認される。