



Dirbato

05

*Summary:* このドキュメントは、*Dirbato @ 42 Tokyo*の05モジュール用の課題である。

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>4</b>
<b>III</b>	<b>Mandatory Exercise 00 : Create a product!</b>	<b>5</b>
<b>IV</b>	<b>Mandatory Exercise 01 : Sell this product!</b>	<b>6</b>
<b>V</b>	<b>Mandatory Exercise 02 : Price!</b>	<b>8</b>
<b>VI</b>	<b>Mandatory Exercise 03 : Super!</b>	<b>10</b>
<b>VII</b>	<b>Mandatory Exercise 04 : Sell them now.</b>	<b>11</b>
<b>VIII</b>	<b>Mandatory Exercise 05 : Sell them more.</b>	<b>13</b>
<b>IX</b>	<b>Bonus Exercise 06 : New Features</b>	<b>16</b>

# Chapter I

## Instructions

- 課題の確認と評価は、あなたの周りにいる受講者により行われる。
- 問題は、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。
- 質問がある場合は、隣の人に聞くこと。それでも分からない場合は、隣の隣の人に聞くこと。
- 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。
- Mandatory Exerciseの問題までは可能な部分まで取り組むこと。
- Bonus Exerciseの問題は時間に余裕がある場合、取り組むこと。
- 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`
- グローバルスコープに、変数やコードを記載しないこと (importを除く)。グローバルスコープには、関数のみを書くこと。
- 関数の定義は、何度でも行うことができる。
- 提出ファイルの中に、以下のようなコードを必ず記載すること。

```
if __name__ == '__main__':  
    your_function( whatever, parameter, is, required )
```

- 上記のコードに、エラー処理に対応するためのコードを追加することができる。
- “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のみを、インポートすること。
- open関数の例外処理は、行わなくても問題ない。

- 課題は、プロジェクトページのGit リポジトリに提出すること。リポジトリ内の提出物のみが、レビュー中の評価対象となる。提出ディレクトリやファイルの名前が正しいことを確認すること。

# Chapter II

## Foreword

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


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

# Mandatory Exercise 00 : Create a product!

	Exercise 00
good quality product	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : <i>product.py, main.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

あなたは世界で一番元気が出る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):
        """
        'This is the best product in the world.'文字列を返す。
        """
        pass
```

- この課題をテストする、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。


例)

```
from product import Product

if __name__ == "__main__":
    product = Product()
    print(product)
```

## Chapter IV

### Mandatory Exercise 01 : Sell this product!

	Exercise 01
a very capable salesperson	
提出するディレクトリ : <i>ex01/</i>	
提出するファイル : <i>salesperson.py, product.py, main.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

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)

```
from product import Product

class Salesperson:
    def __init__(self, name='nameless salesperson.'):
        """
        nameインスタンス変数に引数の値を代入する。
        """
        pass

    def __str__(self):
        """
        nameインスタンス変数の値を返す。
        """
        pass

    def promote(self):
```


```
"""  
Productインスタンスを返す。  
"""  
pass
```

- この課題をテストする、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。



# Chapter V

## Mandatory Exercise 02 : Price!

	Exercise 02
	Price
	提出するディレクトリ : <i>ex02/</i>
	提出するファイル : <i>product.py, main.py</i>
	使用可能な標準ライブラリ : <i>n/a</i>

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)

```
class Product:
    def __init__(self, name, price, description):
        """
        name, price, descriptionインスタンス変数に引数の値を代入する。
        """
        pass

    def __str__(self):
        """
        nameとdescriptionインスタンス変数を以下のフォーマットの文字列
        で返す。
        <nameインスタンス変数の値> : <descriptionインスタンス変数の
        値>
        """
        pass

    def print_attr(self):
        """
```


name, price, description インスタンス変数を以下のフォーマットの文字列で返す。

```
name : <name インスタンス変数の値>
price : <price インスタンス変数の値>
description : <description インスタンス変数の値>
"""
pass
```

- この課題をテストする、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。

# Chapter VI

## Mandatory Exercise 03 : Super!

	Exercise 03
Super	
提出するディレクトリ : <i>ex03/</i>	
提出するファイル : <i>product.py, beverage.py, main.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- `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):
        """
        super()メソッドを活用して、name, price, description,
        temperatureインスタンス変数に引数の値を代入する。
        """
        pass

    def print_attr(self):
        """
        name, price, description, temperatureインスタンス変数
        を以下のフォーマットの文字列で返す。
        name : <nameインスタンス変数の値>
        price : <priceインスタンス変数の値>
        description : <descriptionインスタンス変数の値>
        temperature: <temperatureインスタンス変数の値>
        """
        pass
```

- この課題をテストする、`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。

## Chapter VII

### Mandatory Exercise 04 : Sell them now.

	Exercise 04
Brendan the Vending Machine	
提出するディレクトリ : <i>ex04/</i>	
提出するファイル : <i>product.py, beverage.py, vendingmachine.py, main.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

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):
        """
        パラメーターとして、文字列とBeverageインスタンスの配列を
        受け取りname, stockインスタンス変数に引数の値を代入する。
        """
        pass

    def __str__(self):
        """
        nameインスタンス変数を以下のフォーマットの文字列で返す。
        <nameインスタンス変数の値> the vending machine
        """
        pass

    def sell(self, beverage_name):
        """
```

パラメータとして文字列を受け取り、好きなメッセージを表示する。

メッセージは、在庫にbeverage\_nameと同じ名前のBeverageインスタンスがある場合、専用のメッセージを出力する。ない場合は、適切なメッセージを出力する。

```
"""
pass

def ask(self):
    """
    'Hello, what would you like?'を出力しユーザー入力を待つ。
    ユーザーが入力した値をsellメソッドに受け渡し、呼び出す。
    """
    pass
```

- 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...
```

- この課題をテストする、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。


例)

```
from vendingmachine import Vendingmachine
from beverage import Beverage

if __name__ == "__main__":
    vm = Vendingmachine("Henry", [Beverage('coca-cola', 10, 'COKE', 0)])
    vm.ask()
```

## Chapter VIII

### Mandatory Exercise 05 : Sell them more.

	Exercise 05
	better UX
	提出するディレクトリ : <i>ex05/</i>
	提出するファイル : <i>product.py, beverage.py, vendingmachine.py, main.py</i>
	使用可能な標準ライブラリ : <i>random</i>

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):
        """
        パラメータとして、二つの文字列とBeverageインスタンスの配列
を
        受け取りname, greeting, stockインスタンス変数に代入する。
        """
        pass

    def greet(self):
        """
        greetingインスタンス変数の値を出力する。
        """
        pass

    def display(self):
        """
        在庫の内容を出力する。
```

```

        """
        pass

    def recommend(self):
        """
        在庫からランダムに一つのBeverageインスタンスを選択し、
        その内容を推奨するような文字列を出力する。
        """
        pass

```

- 前問の `__str__` メソッドを追加すること。

```

    def __str__(self):
        """
        nameインスタンス変数を以下のフォーマットの文字列で返す。
        <nameインスタンス変数の値> the vending machine
        """
        pass

```

- 前問の `ask` メソッドを追加すること。

```

    def ask(self):
        """
        'Hello, what would you like?'を出力しユーザー入力を待つ。
        ユーザーが入力した値をsellメソッドに受け渡し、呼び出す。
        """
        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):
        """
        パラメータとして文字列を受け取り、好きなメッセージを表示す
る
        メッセージは、在庫にbeverage_nameと同じ名前の
        Beverageインスタンスがある場合、専用のメッセージを出力する
        ない場合、recommendメソッドを呼び出す。
        """
        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!


```

- この課題をテストする、`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。



# Chapter IX

## Bonus Exercise 06 : New Features

	Exercise 06
The more original, the better	
提出するディレクトリ : <i>ex06/</i>	
提出するファイル : *	
使用可能な標準ライブラリ : *	

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。