



Dirbato

04

Summary: このドキュメントは、*Dirbato @ 42 Tokyo*の04モジュール用の課題である。

Contents

I	Instructions	2
II	Foreword	3
III	Mandatory Exercise 00 : Hello from python!	4
IV	Mandatory Exercise 01 : my first variables	5
V	Mandatory Exercise 02 : Hello!	6
VI	Mandatory Exercise 03 : Give me a name!	7
VII	Mandatory Exercise 04 : Hello Admin!	8
VIII	Mandatory Exercise 05 : Hello Hello Hello Hello!	9
IX	Mandatory Exercise 06 : Mini Fizzbuzz	10
X	Mandatory Exercise 07 : FizzBuzz	12
XI	Mandatory Exercise 08 : Numbers	14
XII	Bonus Exercise 09 : My first dictionary	15
XIII	Bonus Exercise 10 : Key search	17
XIV	Bonus Exercise 11 : Search by value	18
XV	Bonus Exercise 12 : Search by key or value	19
XVI	Bonus Exercise 13 : Create your own command	21

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

<https://xkcd.com/353/>

Chapter III

Mandatory Exercise 00 : Hello from python!

	Exercise 00
Only the best know how to print Hello	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : <i>hello.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- Helloを標準出力に出力するhello関数が定義されている、`hello.py`というスクリプトを作成せよ。
- 例を参考にして、同じ出力になるようにすること。

例)


```
?>python3 hello.py | cat -e
Hello$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    hello()
```

Chapter IV

Mandatory Exercise 01 : my first variables

	Exercise 01
	my first variables
	提出するディレクトリ : <i>ex01/</i>
	提出するファイル : <i>var.py</i>
	使用可能な標準ライブラリ : <i>n/a</i>

- `my_var`関数が定義されている、`var.py`というスクリプトを作成せよ。
- 関数の中に、型が異なる9つの各変数を定義し、各変数の値とクラスに関する情報を標準出力に出力すること。
- その際、以下と同じ出力にすること。


```
?>python3 var.py
42 has a type <class 'int'>
42 has a type <class 'str'>
forty-two has a type <class 'str'>
42.0 has a type <class 'float'>
True has a type <class 'bool'>
[42] has a type <class 'list'>
{'42': '42'} has a type <class 'dict'>
(42,) has a type <class 'tuple'>
{42} has a type <class 'set'>
?>
```

- `print("<class 'int'>")`や、クラスに関する情報は、明示的に標準出力しないこと。
- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    my_var()
```

Chapter V

Mandatory Exercise 02 : Hello!

	Exercise 02
Command line argument?	
提出するディレクトリ : <i>ex02/</i>	
提出するファイル : <i>hello.py</i>	
使用可能な標準ライブラリ : <i>sys</i>	

- 以下の参考例をもとに、同じ挙動をするプログラムを作成せよ。
- 参考例)


```
?>python3 hello.py | cat -e
Hello$
?>python3 hello.py World | cat -e
Hello World$
?>python3 hello.py Dirbato | cat -e
Hello Dirbato$
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    hello()
```

Chapter VI

Mandatory Exercise 03 : Give me a name!

	Exercise 03
If?	
提出するディレクトリ : <i>ex03/</i>	
提出するファイル : <i>hello.py</i>	
使用可能な標準ライブラリ : <i>sys</i>	

- 以下の参考例をもとに、同じ挙動をするプログラムを作成せよ。
- 参考例)


```
?>python3 hello.py | cat -e
Please give me a name$
?>python3 hello.py World | cat -e
Hello World$
?>python3 hello.py Dirbato | cat -e
Hello Dirbato$
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    hello()
```


Chapter VII

Mandatory Exercise 04 : Hello Admin!

	Exercise 04
If?	
提出するディレクトリ : <i>ex04/</i>	
提出するファイル : <i>hello.py</i>	
使用可能な標準ライブラリ : <i>sys</i>	

- 以下の参考例をもとに、同じ挙動をするプログラムを作成せよ。
- 参考例)


```
?>python3 hello.py | cat -e
Please give me a name$
?>python3 hello.py World | cat -e
Hello World$
?>python3 hello.py Dirbato | cat -e
Hello Dirbato$
?>python3 hello.py admin | cat -e
Hi Admin, welcome back$
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    hello()
```

Chapter VIII

Mandatory Exercise 05 : Hello Hello Hello Hello!

	Exercise 05
for?	
提出するディレクトリ : <i>ex05/</i>	
提出するファイル : <i>hello.py</i>	
使用可能な標準ライブラリ : <i>sys</i>	

- 以下の参考例をもとに、同じ挙動をするプログラムを作成せよ。
- 参考例)


```
?>python3 hello.py | cat -e
Please give me a name and a positive number$
?>python3 hello.py World 1 | cat -e
Hello World$
?>python3 hello.py Dirbato 2 | cat -e
Hello Dirbato$
Hello Dirbato$
?>python3 hello.py Dirbato 5 | cat -e
Hello Dirbato$
Hello Dirbato$
Hello Dirbato$
Hello Dirbato$
Hello Dirbato$
?>python3 hello.py Dirbato 0 | cat -e
Please give me a name and a positive number$
?>./hello.py admin 3 | cat -e
Hi Admin, welcome back$
Hi Admin, welcome back$
Hi Admin, welcome back$
?>python3 hello.py admin 0 | cat -e
Please give me a name and a positive number$
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    hello()
```

Chapter IX

Mandatory Exercise 06 : Mini Fizzbuzz

	Exercise 06
FizzBuzz	
提出するディレクトリ : <code>ex06/</code>	
提出するファイル : <code>fizzbuzz.py</code>	
使用可能な標準ライブラリ : <code>sys</code>	

- コマンドライン引数で渡された整数を以下の条件が3の倍数である場合、代わりに“fizz”を標準出力に出力すること。
- コマンドライン引数で渡された整数を以下の条件が5の倍数である場合、代わりに“buzz”を標準出力に出力すること。
- コマンドライン引数で渡された整数を以下の条件が3と5の倍数なのである場合、代わりに“fizzbuzz”を標準出力に出力すること。
- コマンドライン引数で渡された整数が上の条件に当てはまらない場合、整数をそのまま標準出力に出力すること。
- コマンドライン引数がない場合や、コマンドライン引数が2つ以上ある場合は、何も標準出力に出力しないこと。
- 試されるコマンドライン引数の値は、1以上100以下とすること。それ以外の値が渡された場合は、未定義とする。

- 参考例)


```
%> python3 fizzbuzz.py 16 | cat -e
16$
%> python3 fizzbuzz.py 6 | cat -e
fizz$
%> python3 fizzbuzz.py 10 | cat -e
buzz$
%> python3 fizzbuzz.py 30 | cat -e
fizzbuzz$
%>
%> python3 fizzbuzz.py | cat -e
%>
%> python3 fizzbuzz.py 16 100 | cat -e
%>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    hello()
```

Chapter X

Mandatory Exercise 07 : FizzBuzz

	Exercise 07
FizzBuzz	
提出するディレクトリ : <i>ex07/</i>	
提出するファイル : fizzbuzz.py	
使用可能な標準ライブラリ : sys	

- 1からコマンドライン引数で渡された整数までの数字を、1行ずつ標準出力に出力するPythonスクリプトを作成せよ。
- 3の倍数である場合、代わりに“fizz”を標準出力に出力すること。
- 5の倍数である場合、代わりに“buzz”を標準出力に出力すること。
- 3と5の倍数である場合、代わりに“fizzbuzz”を標準出力に出力すること。
- コマンドライン引数がない場合や、コマンドライン引数の数が1より大きい場合は、何も出力しないこと。
- テスター（レビューワー/Moulinette）が試すコマンドライン引数の値は、1以上100以下とすること。それ以外の値が渡された場合は、未定義とする。


Example:

```
%> python3 fizzbuzz.py 16 | cat -e
1$
2$
fizz$
4$
buzz$
fizz$
7$
8$
fizz$
buzz$
11$
fizz$
13$
14$
fizzbuzz$
16$
```

```
%>  
%> python3 fizzbuzz.py | cat -e  
%>  
%> python3 fizzbuzz.py 16 100 | cat -e  
%>
```

Chapter XI


Mandatory Exercise 08 : Numbers

	Exercise 08
Numbers	
提出するディレクトリ : <i>ex08/</i>	
提出するファイル : <i>numbers.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- プロジェクトページから、*dirbato-04.tar.gz*をダウンロードすること。今回の問題で使用する、1から100までの値をカンマで区切った数列が記載されている*numbers.txt*を、*dirbato-04.tar.gz*の*ex08*ディレクトリから入手すること。
- *numbers.txt*ファイルから数字を読み込み、読み込んだ数字を1行ずつ標準出力に出力する、*numbers.py*というPythonスクリプトを作成せよ。その際、カンマは標準出力に出力しないこと。
- *numbers.txt*と*numbers.py*は同じディレクトリに配置されていることを想定する。

Chapter XII

Bonus Exercise 09 : My first dictionary

	Exercise 09
My first dictionary	
提出するディレクトリ : <i>ex09/</i>	
提出するファイル : <i>var_to_dict.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 以下の変数を、`var_to_dict.py`の中に定義すること。


```
d = [  
    ('Allman' , '1946'),  
    ('King' , '1925'),  
    ('Clapton' , '1945'),  
    ('Johnson' , '1911'),  
    ('Berry' , '1926'),  
    ('Vaughan' , '1954'),  
    ('Cooder' , '1947'),  
    ('Richards' , '1943'),  
    ('Hammett' , '1962'),  
    ('Cobain' , '1967'),  
    ('Garcia' , '1942'),  
    ('Beck' , '1944'),  
    ('Ramone' , '1948'),  
    ('White' , '1975'),  
    ('Frusciante' , '1970'),  
    ('Thompson' , '1949'),  
    ('Burton' , '1939')  
]
```


- 上記の変数を、辞書に変換するスクリプトを作成せよ。その際、年号をキーにし、ミュージシャンの名前を値にすること。スクリプトは以下のようなフォーマットで、辞書のキーと値を標準出力に出力すること。

```
1946: Allman
1925: King
1945: Clapton
1911: Johnson
1926: Berry
1954: Vaughan
1947: Cooder
1943: Richards
1962: Hammett
...
```

Chapter XIII

Bonus Exercise 10 : Key search

	Exercise 10
Key search	
提出するディレクトリ : <i>ex10/</i>	
提出するファイル : <i>capital_city.py</i>	
使用可能な標準ライブラリ : <i>sys</i>	

- 以下の変数を、スクリプトにコピーすること。


```
states = {
    "Oregon" : "OR",
    "Alabama" : "AL",
    "New Jersey": "NJ",
    "Colorado" : "CO"
}
capital_cities = {
    "OR": "Salem",
    "AL": "Montgomery",
    "NJ": "Trenton",
    "CO": "Denver"
}
```

- コマンドライン引数から、州の名前（Oregonなど）を受け取り、その州にある州都の名前（Salemなど）を標準出力に出力するスクリプトを作成せよ。州の名前から、その州都の名前を割り出せない場合は、Unknown stateと標準出力に出力すること。コマンドライン引数がない場合や、コマンドライン引数の数が1より大きい場合は、何もせずにスクリプトを終了すること。

```
$> python3 capital_city.py Oregon
Salem
$> python3 capital_city.py Ile-De-France
Unknown state
$> python3 capital_city.py
$> python3 capital_city.py Oregon Alabama
$> python3 capital_city.py Oregon Alabama Ile-De-France
$>
```

Chapter XIV

Bonus Exercise 11 : Search by value

	Exercise 11
Search by value	
提出するディレクトリ : <i>ex11/</i>	
提出するファイル : <i>state.py</i>	
使用可能な標準ライブラリ : <i>sys</i>	

- 以下の変数を、スクリプトにコピーすること。


```
states = {
    "Oregon" : "OR",
    "Alabama" : "AL",
    "New Jersey": "NJ",
    "Colorado" : "CO"
}
capital_cities = {
    "OR": "Salem",
    "AL": "Montgomery",
    "NJ": "Trenton",
    "CO": "Denver"
}
```

- コマンドライン引数から、州都の名前を受け取り、その州都がある州の名前を標準出力に出力するスクリプトを作成せよ。州都の名前から、その州都がある州の名前を割り出せない場合は、Unknown capital cityと標準出力に出力すること。コマンドライン引数がない場合や、コマンドライン引数の数が1より大きい場合は、何もせずにスクリプトを終了すること。

```
$> python3 state.py Salem
Oregon
$> python3 state.py Paris
Unknown capital city
$> python3 state.py
$>
```

Chapter XV

Bonus Exercise 12 : Search by key or value

	Exercise 12
Search by key or value	
提出するディレクトリ : <i>ex12/</i>	
提出するファイル : <i>all_in.py</i>	
使用可能な標準ライブラリ : <i>sys</i>	

- 以下の変数を、スクリプトにコピーすること。

```
states = {
    "Oregon" : "OR",
    "Alabama" : "AL",
    "New Jersey": "NJ",
    "Colorado" : "CO"
}
capital_cities = {
    "OR": "Salem",
    "AL": "Montgomery",
    "NJ": "Trenton",
    "CO": "Denver"
}
```

- 以下の条件を満たすスクリプトを作成せよ。
- このプログラムは、カンマで区切られている単語を含む文字列であるコマンドライン引数を、1つのみ受け取ること。
- このプログラムは、各単語が州の名前、州都の名前、その他の単語のいずれかに該当するかを検知すること。
- このプログラムは、大文字と小文字を区別せず、単語の前後についているスペースを考慮しないこと。
- コマンドライン引数がない場合や、コマンドライン引数の数が1より大きい場合は、何もせずにスクリプトを終了すること。


- 例を参考にして、同じ出力になるようにすること。

例)

```
$> python3 all_in.py "New jersey, Tren ton, NewJersey, Trenton, toto, , sAlem"
Trenton is the capital of New Jersey
Tren ton is neither a capital city nor a state
NewJersey is neither a capital city nor a state
Trenton is the capital of New Jersey
toto is neither a capital city nor a state
Salem is the capital of Oregon
$>
```

Chapter XVI

Bonus Exercise 13 : Create your own command

	Exercise 13
Create your own command	
提出するディレクトリ : <i>ex13/</i>	
提出するファイル : <i>file_viewer.py</i>	
使用可能な標準ライブラリ : <i>sys, argparse</i>	

- 指定した2つの行の間にあるものを標準出力に出力する、スクリプトを作成せよ。
- 例を参考にして、同じ出力になるようにすること。

例)

```
%> python3 file_viewer.py --help | cat -e
usage: file_viewer.py [-h] filename start end$
$
Display specific part of the file.$
$
positional arguments:$
  filename  filename$
  start     starting line$
  end       ending line$
$
optional arguments:$
  -h, --help show this help message and exit$
%>
%> python3 file_viewer.py /etc/passwd 11 13 | cat -e
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false$
root:*:0:0:System Administrator:/var/root:/bin/sh$
daemon:*:1:1:System Services:/var/root:/usr/bin/false$
%>
%> python3 file_viewer.py /etc/passwd 11 a | cat -e
usage: file_viewer.py [-h] filename start end
file_viewer.py: error: argument end: invalid int value: 'a'
%>
%> python3 file_viewer.py /etc/passwd a 13 | cat -e
usage: file_viewer.py [-h] filename start end
file_viewer.py: error: argument start: invalid int value: 'a'
%>
%> python3 file_viewer.py /etc/passwd | cat -e
usage: file_viewer.py [-h] filename start end
```

```
file_viewer.py: error: the following arguments are required: start, end
%>
%> python3 file_viewer.py nosuchfile 11 13 | cat -e
file_viewer.py: nosuchfile: No such file$
%>
%> python3 file_viewer.py /etc/passwd 1000 1001
%>
%> python3 file_viewer.py /etc/passwd 10 1
%>
%> python3 file_viewer.py | cat -e
usage: file_viewer.py [-h] filename start end
file_viewer.py: error: the following arguments are required: filename, start, end
%>
```