



Dirbato

04 Reboot

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

Contents

I	Instructions	2
II	Foreword	3
III	Mandatory Exercise 00 : String Basic 1	4
IV	Mandatory Exercise 01 : String Basic 2	5
V	Mandatory Exercise 02 : String Basic 3	6
VI	Mandatory Exercise 03 : String Basic 4	7
VII	Mandatory Exercise 04 : String Basic 5	8
VIII	Mandatory Exercise 05 : List Basic 1	9
IX	Mandatory Exercise 06 : List Basic 2	10
X	Mandatory Exercise 07 : List Basic 3	11
XI	Mandatory Exercise 08 : List Basic 4	12
XII	Mandatory Exercise 09 : List Basic 5	13
XIII	Mandatory Exercise 10 : Dictionary Basic 1	14
XIV	Mandatory Exercise 11 : Dictionary Basic 2	15
XV	Mandatory Exercise 12 : Dictionary Basic 3	16
XVI	Mandatory Exercise 13 : Dictionary Basic 4	17
XVII	Mandatory Exercise 14 : Dictionary Basic 5	18
XVIII	Mandatory Exercise 15 : Basic 1	19
XIX	Mandatory Exercise 16 : Basic 2	20
XX	Mandatory Exercise 17 : Basic 3	21
XXI	Mandatory Exercise 18 : Basic 4	22
XXII	Mandatory Exercise 19 : Basic 5	23
XXIII	Bonus Exercise 20 : Create your own command	24

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

Python 20本ノック

Chapter III

Mandatory Exercise 00 : String Basic 1

	Exercise 00
String Basic 1	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : <i>string_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 渡された文字列の長さを標準出力に出力するstring_basic関数が定義されている、string_basic.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 string_basic.py | cat -e
5$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    string_basic("Hello")
```

Chapter IV

Mandatory Exercise 01 : String Basic 2

	Exercise 01
String Basic 2	
提出するディレクトリ : <i>ex01/</i>	
提出するファイル : <code>string_basic.py</code>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 渡された文字列の長さを標準出力に出力する`string_basic`関数が定義されている、`string_basic.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 string_basic.py | cat -e
{'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    string_basic("google.com")
```

Chapter V

Mandatory Exercise 02 : String Basic 3

	Exercise 02
String Basic 3	
提出するディレクトリ : <i>ex02/</i>	
提出するファイル : <i>string_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 渡された文字列の最初の2文字と最後の2文字で構成される文字列を標準出力に出力するstring_basic関数が定義されている、string_basic.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 string_basic.py | cat -e
dito$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    string_basic("dirbato")
```

Chapter VI

Mandatory Exercise 03 : String Basic 4

	Exercise 03
String Basic 4	
提出するディレクトリ : <i>ex03/</i>	
提出するファイル : <i>string_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 渡された文字列から「s」を「S」に変更された文字列を出力するstring_basic関数が定義されている、string_basic.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 string_basic.py | cat -e
paSS$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    string_basic("pass")
```


Chapter VII

Mandatory Exercise 04 : String Basic 5

	Exercise 04
String Basic 5	
提出するディレクトリ : <i>ex04/</i>	
提出するファイル : <i>string_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- *n* 番目の文字を削除し、標準出力に出力する `string_basic` 関数が定義されている、`string_basic.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 string_basic.py | cat -e
dirbato$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    string_basic("dirbaato", 5)
```

Chapter VIII

Mandatory Exercise 05 : List Basic 1

	Exercise 05
List Basic 1	
提出するディレクトリ : <i>ex05/</i>	
提出するファイル : <i>list_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 整数型のリストの要素の総和を標準出力に出力する`list_basic`関数が定義されている、`list_basic.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 list_basic.py | cat -e
6$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    list_basic([1,2,3])
```

Chapter IX

Mandatory Exercise 06 : List Basic 2

	Exercise 06
List Basic 2	
提出するディレクトリ : <i>ex06/</i>	
提出するファイル : <i>list_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 整数型のリストの要素で一番大きい値を標準出力に出力する`list_basic`関数が定義されている、`list_basic.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 list_basic.py | cat -e
3$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    list_basic([1,2,3])
```

Chapter X

Mandatory Exercise 07 : List Basic 3

	Exercise 07
List Basic 3	
提出するディレクトリ : <i>ex07/</i>	
提出するファイル : <i>list_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 文字列型のリストの要素で一番長い文字列を標準出力に出力する`list_basic`関数が定義されている、`list_basic.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 list_basic.py | cat -e
ccc$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    list_basic(['a', 'bb', 'ccc'])
```

Chapter XI

Mandatory Exercise 08 : List Basic 4

	Exercise 08
List Basic 4	
提出するディレクトリ : <i>ex08/</i>	
提出するファイル : <i>list_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- リストの中で重複する値を削除し、新しいリストを標準出力に出力する`list_basic`関数が定義されている、`list_basic.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 list_basic.py | cat -e
['a', 'b', 'c']$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    list_basic(['a', 'b', 'c', 'c'])
```

Chapter XII

Mandatory Exercise 09 : List Basic 5

	Exercise 09
List Basic 5	
提出するディレクトリ : <i>ex09/</i>	
提出するファイル : <i>list_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 2つのリストを結合し、1つになったリストを標準出力に出力する`list_basic`関数が定義されている、`list_basic.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 list_basic.py | cat -e
['d', 'i', 'r', 'b', 'a', 't', 'o']$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    list_basic(['d', 'i', 'r'], ['b', 'a', 't', 'o'])
```

Chapter XIII

Mandatory Exercise 10 : Dictionary Basic 1

	Exercise 10
Dictionary Basic 1	
提出するディレクトリ : <i>ex10/</i>	
提出するファイル : <i>dictionary_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 辞書のキーと値を標準出力に出力する`dictionary_basic`関数が定義されている、`dictionary_basic.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 dictionary_basic.py | cat -e
key:value$
dir:bato$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    dictionary_basic({'key':'value', 'dir':'bato'})
```

Chapter XIV

Mandatory Exercise 11 : Dictionary Basic 2

	Exercise 11
	Dictionary Basic 2
	提出するディレクトリ : <i>ex11/</i>
	提出するファイル : <i>dictionary_basic.py</i>
	使用可能な標準ライブラリ : <i>n/a</i>

- 二つの辞書を結合して、内容を標準出力に出力する`dictionary_basic`関数が定義されている、`dictionary_basic.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 dictionary_basic.py | cat -e
{1: 10, 2: 20, 3: 30, 4: 40}$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    dictionary_basic({1:10, 2:20}, {3:30, 4:40})
```


Chapter XV

Mandatory Exercise 12 : Dictionary Basic 3

	Exercise 12
	Dictionary Basic 3
	提出するディレクトリ : <i>ex12/</i>
	提出するファイル : <i>dictionary_basic.py</i>
	使用可能な標準ライブラリ : <i>n/a</i>

- 二つの辞書を結合して、内容を標準出力に出力するdictionary_basic関数が定義されている、dictionary_basic.pyというスクリプトを作成せよ。
- 同じキーが二つの辞書に存在する場合、値を合算すること。
- 例を参考にして、同じ出力になるようにすること。

例)


```
?>python3 dictionary_basic.py | cat -e
{'a': 400, 'b': 400, 'd': 400, 'c': 300}$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    dictionary_basic({'a': 100, 'b': 200, 'c':300}, {'a': 300, 'b': 200, 'd':400})
```

Chapter XVI

Mandatory Exercise 13 : Dictionary Basic 4

	Exercise 13
Dictionary Basic 4	
提出するディレクトリ : <i>ex13/</i>	
提出するファイル : <i>dictionary_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 渡された文字列から現れる文字をキーとし、現れた回数を値とする辞書を作成し、その内容を標準出力に出力する`dictionary_basic`関数が定義されている、`dictionary_basic.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 dictionary_basic.py | cat -e
{'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    dictionary_basic('google.com')
```

Chapter XVII

Mandatory Exercise 14 : Dictionary Basic 5

	Exercise 14
Dictionary Basic 5	
提出するディレクトリ : <i>ex14/</i>	
提出するファイル : <i>dictionary_basic.py</i>	
使用可能な標準ライブラリ : <i>n/a</i>	

- 以下の例を参考にして、`dictionary_basic`関数が定義されている、`dictionary_basic.py`というスクリプトを作成せよ。

例)


```
?>python3 dictionary_basic.py | cat -e
{'yellow': [1, 3], 'blue': [2, 4], 'red': [1]}$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    dictionary_basic([('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)])
```

Chapter XVIII

Mandatory Exercise 15 : Basic 1

	Exercise 15
	Basic 1
提出するディレクトリ : <i>ex15/</i>	
提出するファイル : basic.py	
使用可能な標準ライブラリ : n/a	

- 以下の例を参考にして、`basic`関数が定義されている、`basic.py`というスクリプトを作成せよ。

例)


```
?>python3 basic.py | cat -e
['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4', 'p5', 'q5']$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    basic(['p', 'q'], 5)
```

Chapter XIX

Mandatory Exercise 16 : Basic 2

	Exercise 16
	Basic 2
提出するディレクトリ : <i>ex16/</i>	
提出するファイル : basic.py	
使用可能な標準ライブラリ : n/a	

- 以下の例を参考にして、`basic`関数が定義されている、`basic.py`というスクリプトを作成せよ。

例)


```
?>python3 basic.py | cat -e
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    basic([1, 3, 5, 7, 9, 11], [0, 2, 4, 6, 8, 10])
```

Chapter XX

Mandatory Exercise 17 : Basic 3

	Exercise 17
	Basic 3
提出するディレクトリ : <i>ex17/</i>	
提出するファイル : basic.py	
使用可能な標準ライブラリ : n/a	

- 3桁の数字の異なるすべての組み合わせを、昇順で標準出力に出力するbasic関数が定義されている、basic.pyというスクリプトを作成せよ。

例)


```
?>python3 basic.py | cat -e
012, 013, 014, 015, 016, 017, 018, 019, 023, ..., 789$
?>
```

- 定義した関数を、以下のように呼び出すこと。

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

Chapter XXI

Mandatory Exercise 18 : Basic 4

	Exercise 18
	Basic 4
提出するディレクトリ : <i>ex18/</i>	
提出するファイル : basic.py	
使用可能な標準ライブラリ : n/a	

- 00から99まで2つの2桁の数字の異なるすべての組み合わせ(XX XX)を、昇順で標準出力に出力するbasic関数が定義されている、basic.pyというスクリプトを作成せよ。

例)


```
?>python3 basic.py | cat -e
00 01, 00 02, 00 03, 00 04, 00 05, ..., 00 99, 01 02, ..., 97 99, 98 99$
?>
```

- 定義した関数を、以下のように呼び出すこと。

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

Chapter XXII

Mandatory Exercise 19 : Basic 5

	Exercise 19
	Basic 5
	提出するディレクトリ : <i>ex19/</i>
	提出するファイル : <i>basic.py</i>
	使用可能な標準ライブラリ : <i>n/a</i>

- *n*の数の異なるすべての組み合わせを、昇順で標準出力に出力する*basic*関数が定義されている、*basic.py*というスクリプトを作成せよ。
- *n*は、次のようになる。 : $0 < n < 10$

例)


```
?>python3 basic.py | cat -e
01, 02, 03, ..., 09, 12, ..., 79, 89$
?>
```

- 定義した関数を、以下のように呼び出すこと。

```
if __name__ == '__main__':
    basic(2)
```


Chapter XXIII

Bonus Exercise 20 : Create your own command

	Exercise 20
Create your own command	
提出するディレクトリ : <i>ex20/</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
%>
```