



Piscine 101

Python 01

Summary: このドキュメントは、*Piscine 101 @ 42Tokyo*の *PYTHON 01*モジュール用の課題である。

Contents

| | | |
|-------------|--|-----------|
| I | Instructions | 2 |
| II | Foreword | 4 |
| III | Exercise 00 : Hello from python! | 5 |
| IV | Exercise 01 : my first variables | 7 |
| V | Exercise 02 : FizzBuzz | 8 |
| VI | Exercise 03 : Numbers | 10 |
| VII | Exercise 04 : My first dictionary | 11 |
| VIII | Exercise 05 : Key search | 13 |
| IX | Exercise 06 : Search by value | 15 |
| X | Exercise 07 : Search by key or value | 16 |
| XI | Exercise 08 : Create your own command | 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、どんなファイルも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
- 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。

- guacamole.42tokyo.jp上で課題に取り組むこと。


Chapter II

Foreword

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

Chapter III

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
Hello!
?>
```

- 定義した関数を、以下のように呼び出すこと。

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



<https://bit.ly/3tkznIU>




<https://bit.ly/2MGe2Jb>



<https://bit.ly/3tsTAMW>

Chapter IV

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()
```



<https://bit.ly/3cykcpu>

Chapter V

Exercise 02 : FizzBuzz

| | |
|---|-------------|
|  | Exercise 02 |
| FizzBuzz | |
| 提出するディレクトリ : <i>ex02/</i> | |
| 提出するファイル : <i>fizzbuzz.py</i> | |
| 使用可能な標準ライブラリ : <i>sys</i> | |

- 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  
%>
```



<https://bit.ly/39JsskR>

Chapter VI

Exercise 03 : Numbers

| | |
|---|------------------------------|
|  | Exercise 03 |
| | Numbers |
| | 提出するディレクトリ : <i>ex03/</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、`piscine-101-01.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、`piscine-101-01.tar.gz`の`ex03`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。



<https://bit.ly/3cEA8qm>

Chapter VII

Exercise 04 : My first dictionary

| | |
|---|-------------|
|  | Exercise 04 |
| My first dictionary | |
| 提出するディレクトリ : <i>ex04/</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
...
```



<https://bit.ly/2Muc12V>

Chapter VIII

Exercise 05 : Key search

| | |
|---|-------------|
|  | Exercise 05 |
| Key search | |
| 提出するディレクトリ : <i>ex05/</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
$>
```



<https://bit.ly/2Muc12V>

Chapter IX

Exercise 06 : Search by value

| | |
|---|----------------------------|
|  | Exercise 06 |
| | Search by value |
| | 提出するディレクトリ : <i>ex06/</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
$>
```



<https://bit.ly/2Muc12V>

Chapter X

Exercise 07 : Search by key or value

| | |
|---|-------------|
|  | Exercise 07 |
| Search by key or value | |
| 提出するディレクトリ : <i>ex07/</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  
$>
```



<https://bit.ly/2Muc12V>

Chapter XI

Exercise 08 : Create your own command

| | |
|---|-------------|
|  | Exercise 08 |
| Create your own command | |
| 提出するディレクトリ : <i>ex08/</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
%>
```



<https://bit.ly/3tpqrC3>