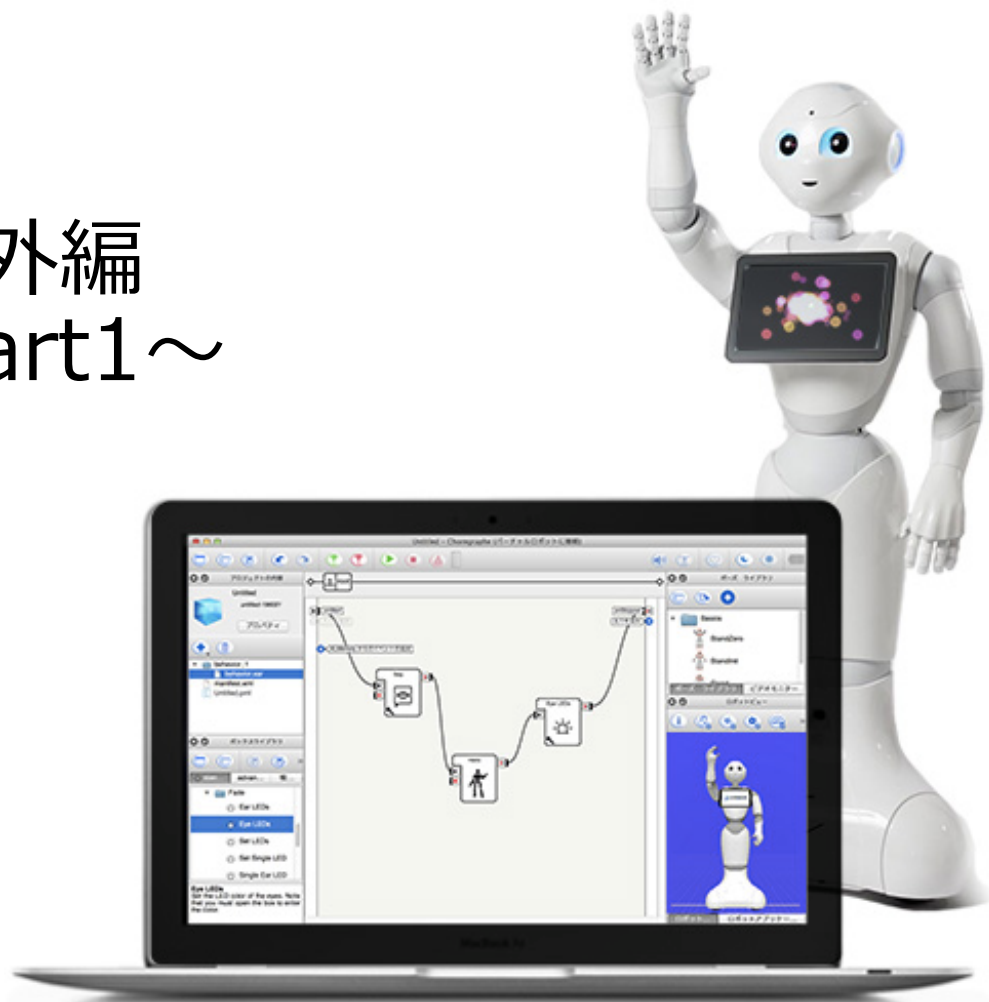


# Atelier Akihabara

## ワークショップ 番外編 Python入門～Part1～

2017/9/08

Softbank Robotics



## 免責事項

このワークショップは  
アトリエのスタッフが作成したものであり  
ソフトバンク公式のものではないことを  
ご承知ください。

## 実体験とコミュニティで開発を促進する

アトリエ



✓ Pepperのアプリ開発を実体験

コミュニティ

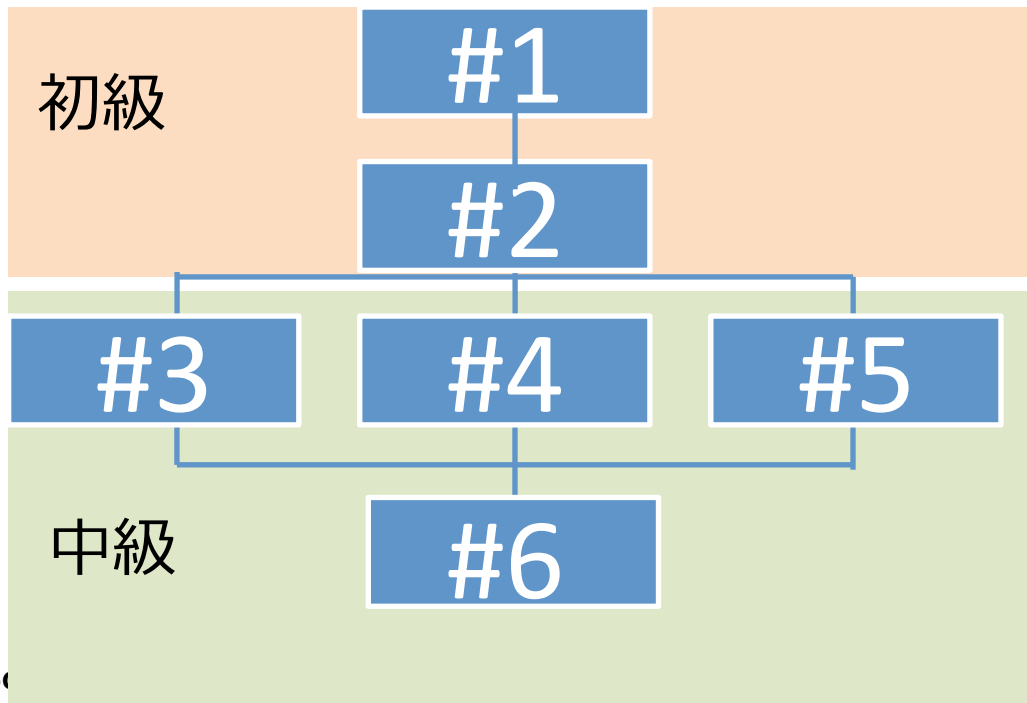


✓ 経験や知見を共有

相互  
促進

## 実体験とコミュニティで開発を促進する

ワークショップ



タッチアンドトライ

自由に開発  
質問はスタッフに  
お客様同士の交流

## 実体験とコミュニティで開発を促進する

### アトリエサテライト

有志でPepperと開発スペースを  
提供している  
企業、大学、コミュニティスペース

秋葉原で回答できない質問は  
各サテライトへ



- ・ **お名前**
- ・ **所属**
- ・ **本日の意気込み**
- ・ **プログラミング経験など**

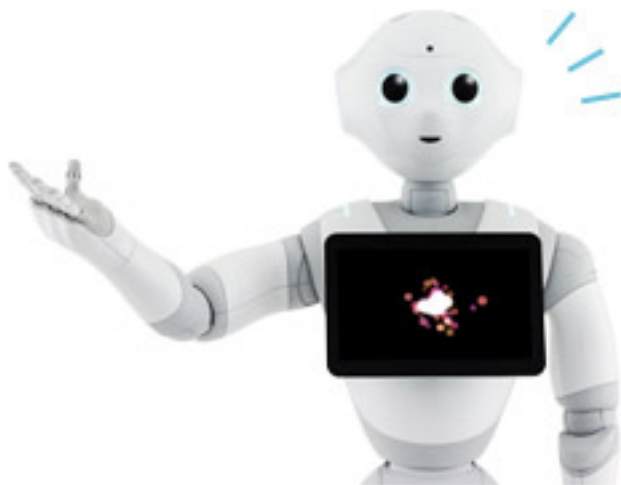
例：

本日の案内を勤めさせていただきます、  
大山 まりほ(Mariho Oyama)と申します。

# Pythonの基本的な記述の仕方を学ぶ

- 出力
- 変数
- リスト
- 辞書

Pythonとは？





# Pythonとは

- 1991年にオランダ人のグイド・ヴァン・ロッサム氏によって開発された汎用的なプログラミング言語
- There's only one way to do it
  - 読みやすく、効率もよいコードをなるべく簡単に書けるようにする

日本語では  
二シキヘビ



## ① シンプルな文法

- インデントを強制することで誰が書いても似たようなコードになる

```
int factorial(int x)
{
    if (x == 0) {
        return 1;
    } else {
        return x * factorial(x - 1);
    }
}
```

C言語

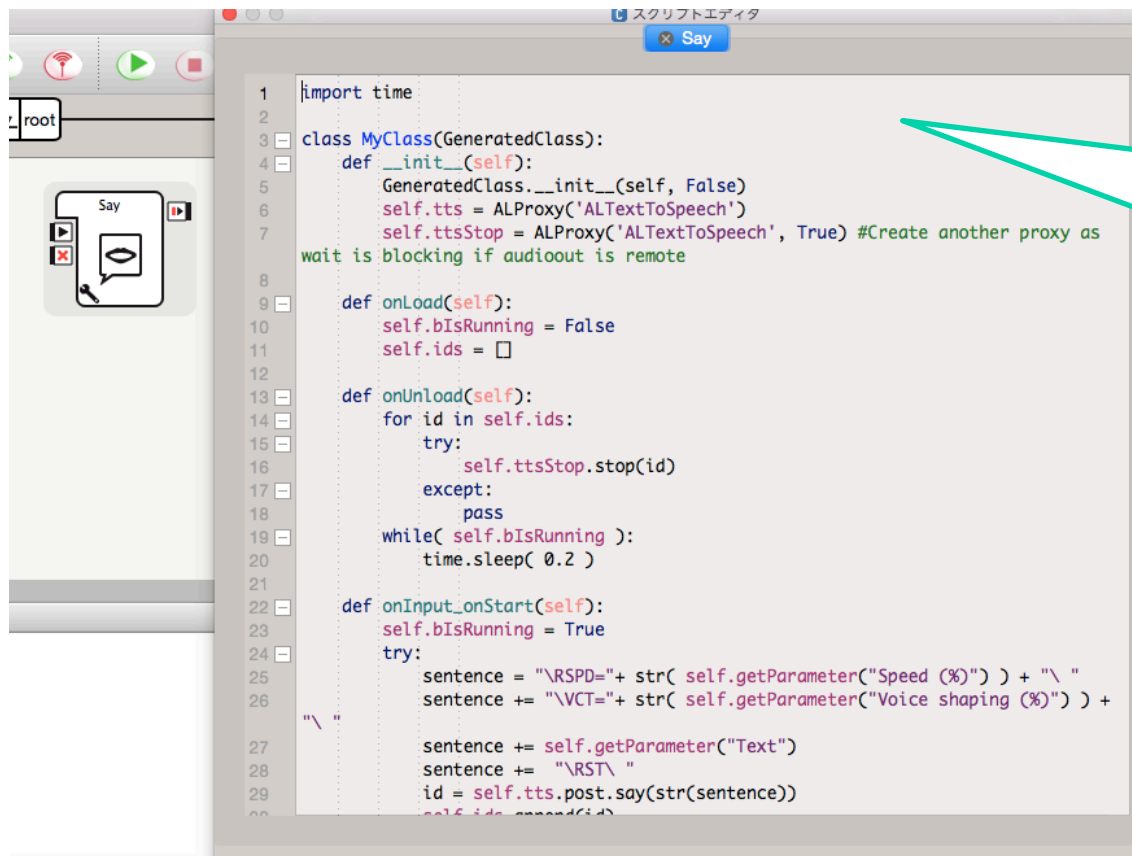
```
int factorial(int x) {
    if(x == 0) {return 1;} else
    {return x * factorial(x - 1); } }
```

Python

```
def factorial(x):
    if x == 0:
        return 1
    else:
        return x * factorial(x - 1)
```

## ② 豊富なライブラリ群

- 特に数学系のライブラリが充実
- 数学計算からデータベース、Web 開発、GUI アプリの作成等



高度なアプリ  
開発にPython  
は必須！

## インストール方法

参考(Windows):

<http://qiita.com/taiponrock/items/f574dd2cddf8851fb02c>

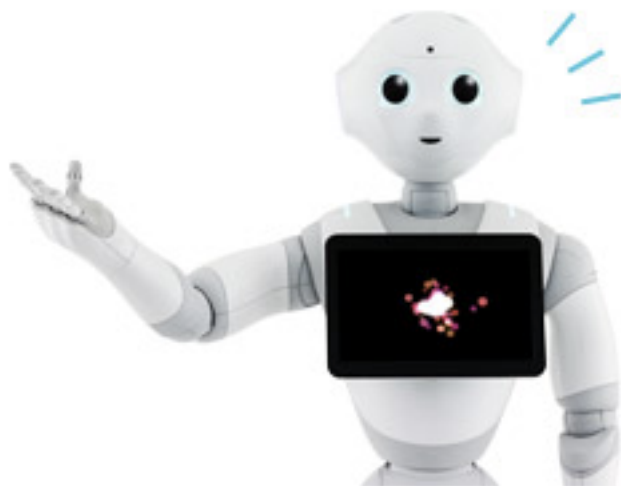
参考(mac):

<http://qiita.com/ms-rock/items/6e4498a5963f3d9c4a67>

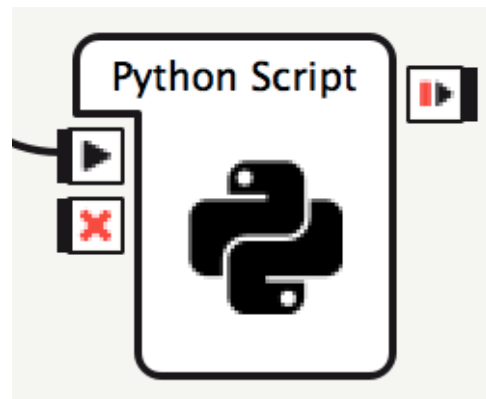
## 実行方法

-コマンドライン上で[python ファイル名]を入力

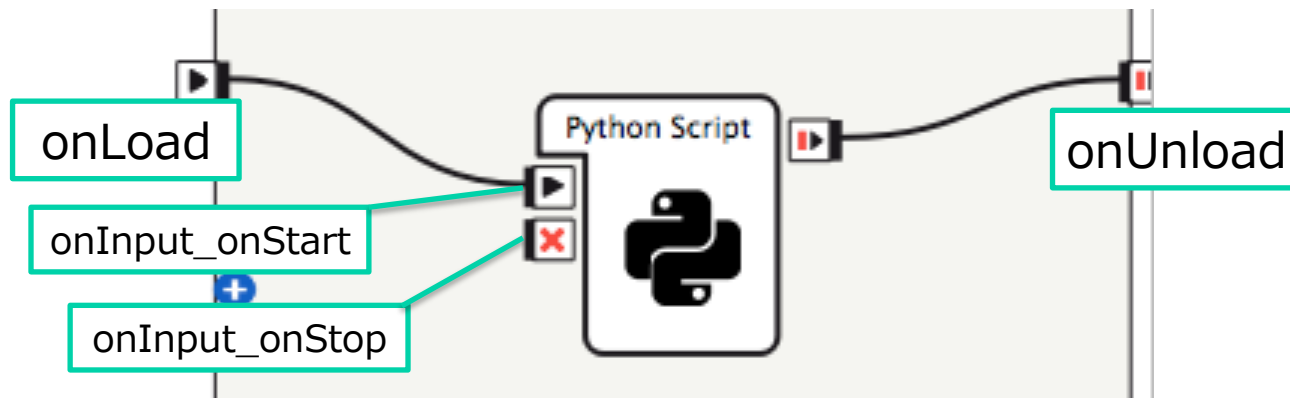
```
ooyamamariho-no-MacBook-Air:jibun mariho$  
ooyamamariho-no-MacBook-Air:jibun mariho$ python panda.py
```



使うボックスはこれ



# Python Script boxの関数



__init__	behaviorが読み込まれた時
onLoad	親ボックスのonStartが実行された時
onUnload	親ボックスのonStopが実行された時
onInput_onStart	ボックスのonStartに信号が入力された時
onInput_onStop	ボックスのonStoppedに信号が入力された時

# Python Script boxの関数

```
class MyClass(GeneratedClass):  
    def __init__(self):  
        GeneratedClass.__init__(self)  
  
    def onLoad(self):  
        #put initialization code here  
        pass  
  
    def onUnload(self):  
        #put clean-up code here  
        pass  
  
    def onInput_onStart(self):  
        #self.onStopped() #activate the output of the box  
        pass  
  
    def onInput_onStop(self):  
        self.onUnload() #it is recommended to reuse the clean-up as the box is stopped  
        self.onStopped() #activate the output of the box
```

ここにかく

## —print文— 出力する

ws.py

```
print 出力させたい内容
```

Choregraphe上

```
self.logger.info(出力させたい内容)
```

```
self.logger.info("Hello World!!")  
self.logger.info(10)  
self.logger.info(3.14)
```

```
Hello World!!  
10  
3.14
```



# 一四則演算一 和・差・積・商・剰余

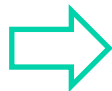
演算	演算子
和	+
差	-
積	*
商	/
剰余	%

```
self.logger.info(11+3)
self.logger.info(11-3)
self.logger.info(11*3)
self.logger.info(11/3)
self.logger.info(11%3)
self.logger.info(11.0/3.0)
```

型	意味	例
int	整数値	1 , 150 , -30
float	浮動小数点数	1.414 , 0.5 , -1.2
str	文字列	“Apple” , “Hello!” , “50”

- Pythonでは型の指定をする必要がない(代入時)

```
x = 5  
y = 1.5  
z = “Apple”
```



自動でxはint型、yはfloat型、zはstr型と認識

```
int x = 5;  
float y = 1.5;  
str z = “Apple”;
```

# フォーマット指定子

## 使い方

型	指定子
int	%d
float	%f
str	%s

文字列にint型の変数を  
代入すると宣言

" x = %d " % x

代入する変数

## —format関数— 文字列に変数を埋め込む

- 文字列に変数を埋め込み出力することができる

“任意の文字列1{}任意の文字列2”.format(変数)

⇒ 任意の文字列1変数任意の文字列2

“文字列1{0}文字列2{1}文字列3”.format(変数1,変数2)

⇒ 文字列1変数1文字列2変数2文字列3

## —print文— 文字・変数を出力する

```
x = 10  
y = 1.5  
z = "apple"  
self.logger.info(y+0.5)  
self.logger.info(z+"banana")  
self.logger.info("x : {0},y : {1},z : {2}".format(x,y,z))
```

```
2.0  
applebanana  
x : 10, y : 1.5,z : apple
```

# ーリストー 複数の値を一つの値として扱う

- 値を複数格納することができる(配列という)
  - 値1つ1つのデータを要素という
  - データのいる場所(何番目にいるか)をindexという
  - リスト名[index(数字)]で要素を取り出すことができる

```
data = [5,19,-8,3,22,93]
```

index	0	1	2	3	4	5
data	5	19	-8	3	22	93

↑ data[4]

## ーリストー 文字列も配列

`z = "apple"`



`z`

a	p	p	l	e
---	---	---	---	---

```
z = "apple"
```

```
list = [5,3.2,"melon",'b']
```

```
self.logger.info(z[3])
```

```
self.logger.info(list[2])
```

```
self.logger.info(list[0]+3)
```

```
self.logger.info(list[3]+"anana")
```

- 1組の値(keyとvalue)を複数格納することができる
  - 同じ辞書内に同じkeyは使えない
  - 順序は考慮されていない

```
pepper =  
{"height":121,"weight":29,"atelier":"akihabara"}
```

	key	value
pepper	height	121
	weight	29
	atelier	"akihabara"



```
pepper = {"height":121,"weight":  
29,"atelier":"akihabara"}  
self.logger.info(pepper["height"])           #値の取得  
pepper["company"]="softbank"  #要素の追加  
self.logger.info(pepper)
```

pepper	key	value	
	height	121	
	weight	29	
	atelier	"akihabara"	
	company	"softbank"	←追加

## 一演習問題一

①花子さんの身長・体重が格納されている辞書がある。  
BMIを小数点第2位まで求めて出力せよ。

hanako = {"height":150,"weight":42}

はなこさんのBMIは18.67です。

変数がfloat型の場合：  
(y: {:.2f}).format(y)と書くと  
小数点以下の桁数指定ができる。

BMI : 
$$\frac{\text{体重(kg)}}{\text{身長(m)} \times \text{身長(m)}}$$

ヒント

- ・辞書のvalueの取り出し方は？  
→辞書名[key]

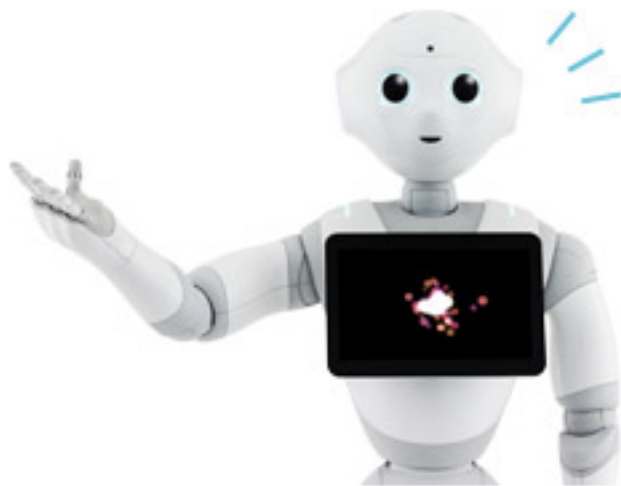
②花子さんと太郎さんの身長・体重が格納されている辞書のリストがある。BMIを小数点第2位まで求めて出力せよ。

```
data = [{"height":150,"weight":42}, {"height":170,"weight":60}]
```

はなこさんのBMIは18.67で太郎さんのBMIは20.76です。

リストの要素が辞書になっている！

→リストから辞書を取り出して、valueを取り出す



まとめ



# Pythonの基本的な記述の仕方を学ぶ

出力(print文)

四則演算

変数の型

リスト

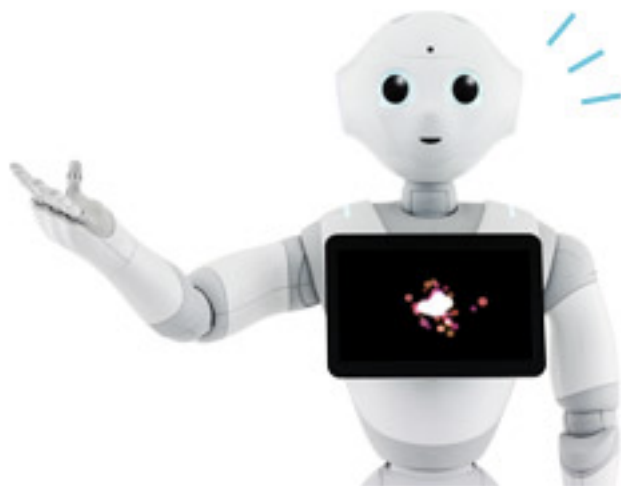
ここまで→ 辞書

条件分岐(if文)

繰り返し文(for文、while文)

関数とは

choregraphe特有のpythonの記述の仕方



**おまけ**



## Pepper デベロッパーポータル

「Pepper developer」で検索

<https://developer.softbankrobotics.com/jp-ja>

### Pepperに関するデベロッパー向けの情報を集約したポータルサイト

- ・ 技術ドキュメント
- ・ 事例を共有するショーケース
- ・ Pepper SDK for Android Studioのダウンロード
- ・ 最新ニュースの提供

## Pepper アトリエ秋葉原 with SoftBank

「アトリエ秋葉原 ブログ」で検索

- ・ ペッパー開発に役立つ記事を見ることができる
- ・ イベントの紹介とイベントのレポートが見ることができる
- ・ tipsの項目から開発に便利なツールを手に入れることができる



## アトリエ秋葉原FBグループ

「アトリエ秋葉原 FB」で検索

- ・アトリエ秋葉原のFacebookグループです
- ・情報共有や質問ができます

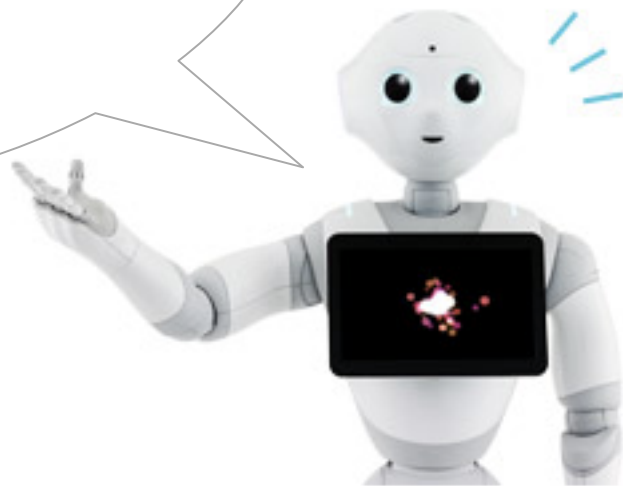
Qiita

「Qiita pepper」で検索

- ・ エンジニアの情報交換サイト
- ・ PepperタグでPepperに関する様々な技術情報がある

**おつかれさまでした！  
これにてWS番外編は終わりになります。**

WSは続けてぜひ受講してみてください  
タッチアンドトライで質問もしてみてください



①花子さんの身長・体重が格納されている辞書がある。  
BMIを小数点第2位まで求めて出力せよ。

```
hanako = {"height":150,"weight":42}  
bmi = hanako["weight"]*10000.0/  
(hanako["height"]*hanako["height"])  
self.logger.info("はなこさんのBMIは{:.2f}で  
す。".format(bmi))
```

②花子さんと太郎さんの身長・体重が格納されている辞書のリストがある。BMIを小数点第2位まで求めて出力せよ。

```
data = [{"height":150,"weight":42}, {"height":170,"weight":60}]
bmi1 = data[0]["weight"]*10000.0/(data[0]["height"]*data[0]["height"])
bmi2 = data[1]["weight"]*10000.0/(data[1]["height"]*data[1]["height"])
self.logger.info("はなこさんのBMIは{0:.2f}でたろうさんのBMIは{1:.2f}です。".format(bmi1,bmi2))
```