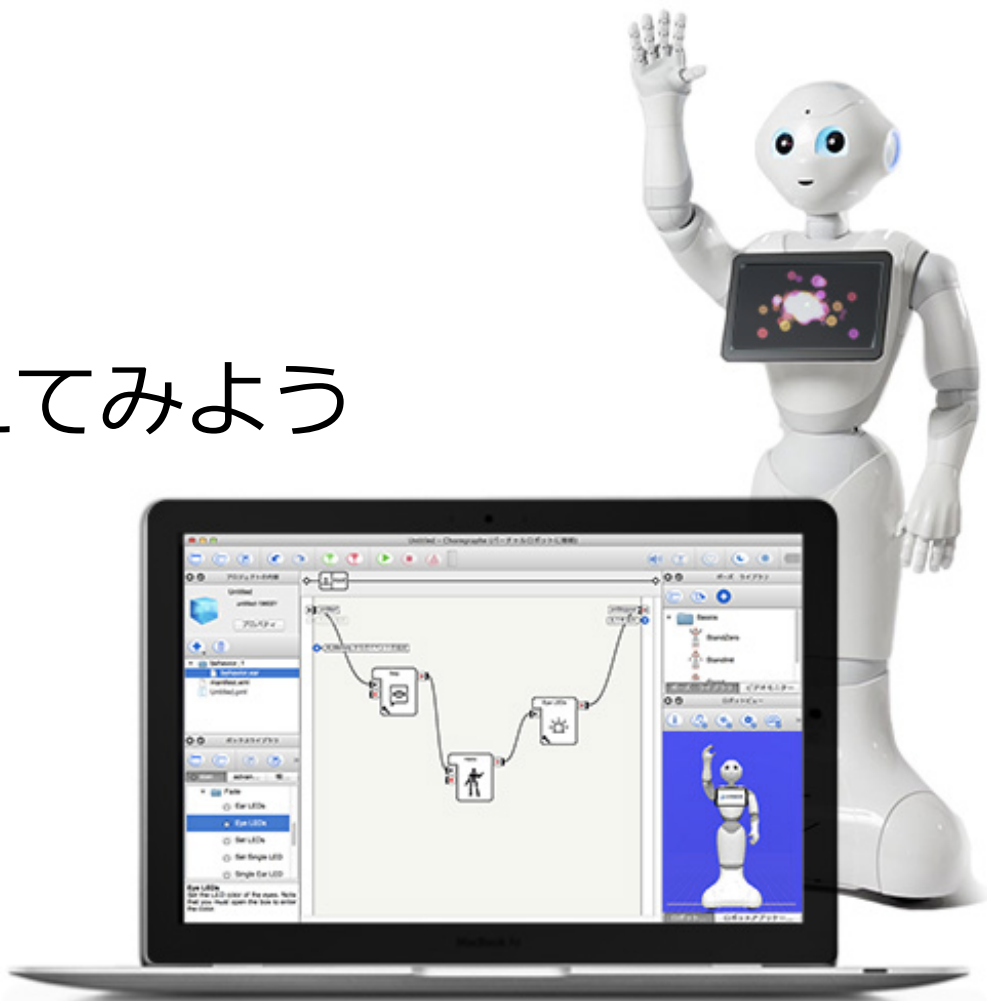


# Atelier Akihabara ワークショップ 既存のBoxを Pythonで書き換えてみよう

2017/6/26



## 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

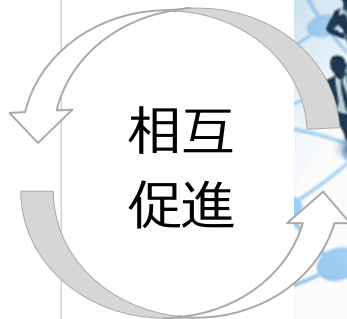


✓ Pepperのアプリ開発を実体験

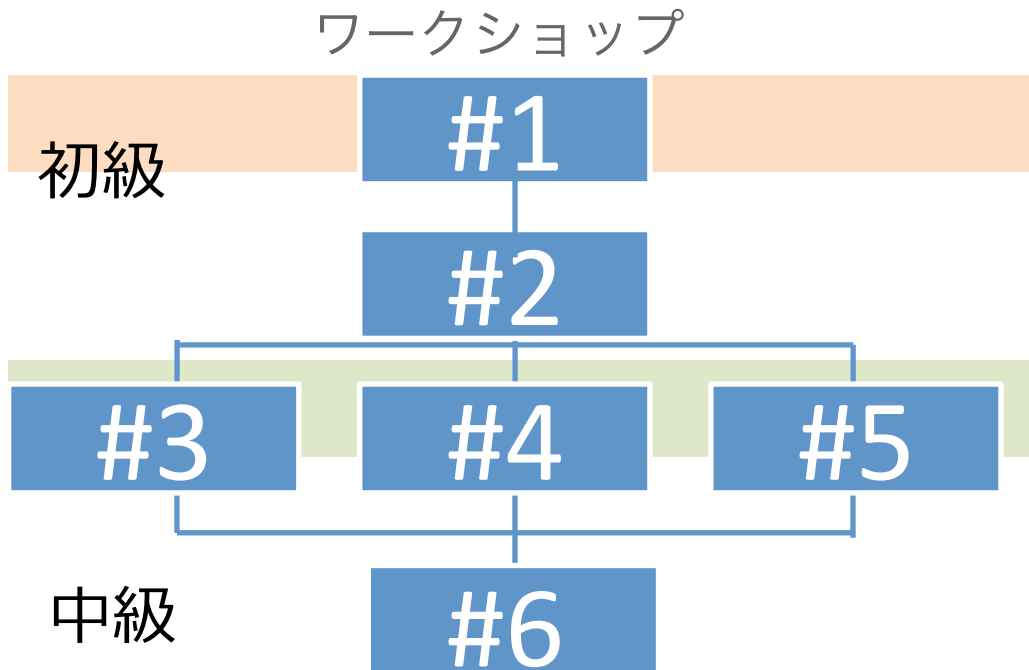
コミュニティ



✓ 経験や知見を共有



## 実体験とコミュニティで開発を促進する



## 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

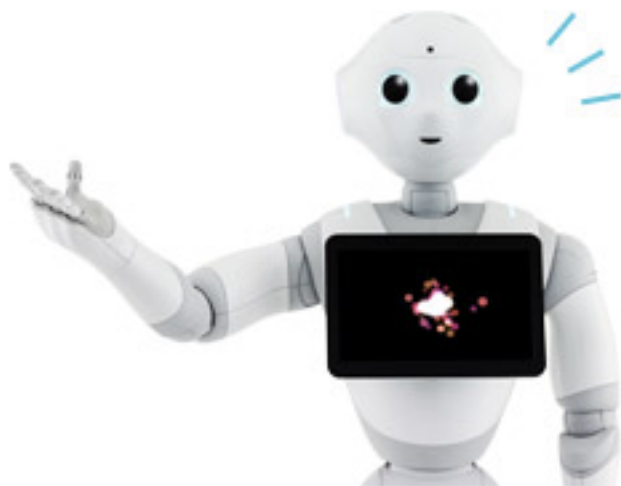


- ・ **お名前**
- ・ **所属**
- ・ **本日の意気込み**
- ・ **Pythonの知識**

例：

本日の案内を勤めさせていただきます、  
と申します。

1. BOXのしくみ
2. QRコード読み取り
3. メールを送る
4. 来客者を伝える簡単なアプリ
5. おまけ



Boxのしくみ



## OnlyOnceBoxを例に見てみる

```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False)
4
5     def onLoad(self):
6         self.bFirstTime = True
7
8     def onUnload(self):
9         self.bFirstTime = True
10
11    def onInput_onSignal(self):
12        if( self.bFirstTime ):
13            self.bFirstTime = False
14            self.firstTime()
```

event Input  
Name: onSignal

Type: "Bang"  
Tooltip:  
The signal that must pass only once.



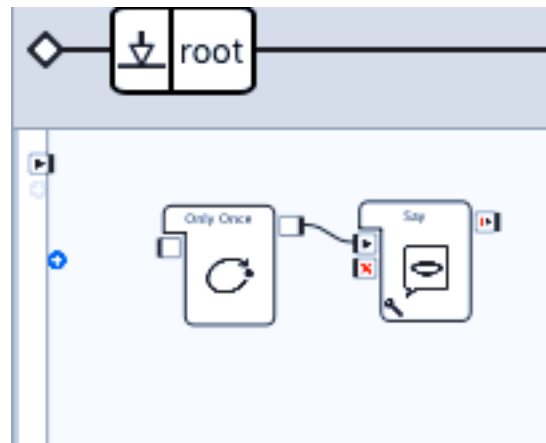
punctual Output  
Name: firstTime

Type: "Bang"  
Tooltip:  
Signal sent first time the onSignal input is stimulated.

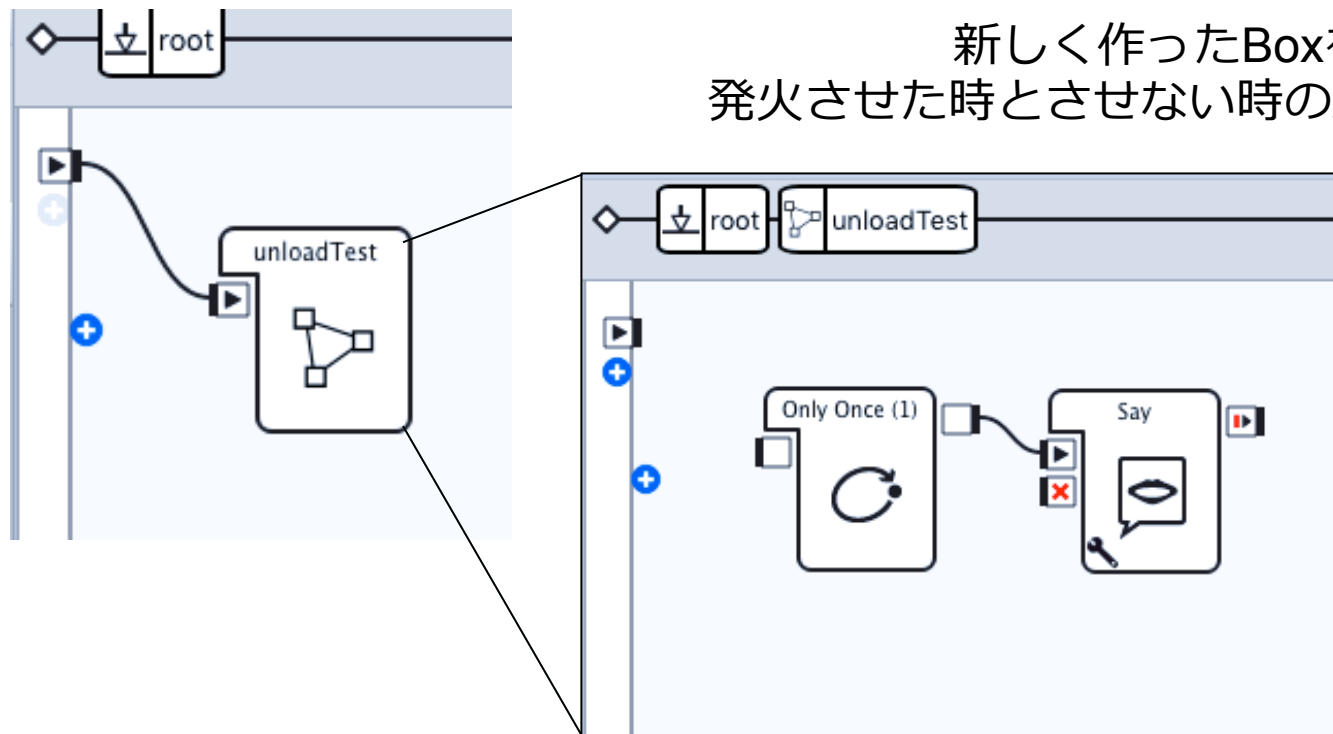
OnlyOnceBoxを改造して、  
onLoad関数がいつ実行されるのか見てみる。

```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False)
4
5     def onLoad(self):
6         self.bFirstTime = True
7
8         # 下の関数が実行されると出力から信号が出る
9         self.firstTime()
10
11     def onUnload(self):
12         self.bFirstTime = True
13
14     def onInput_onSignal(self):
15         if( self.bFirstTime ):
16             self.bFirstTime = False
17             self.firstTime()
```

左のように改造してあげると、  
onLoad関数が実行されると  
Sayが発火するはず



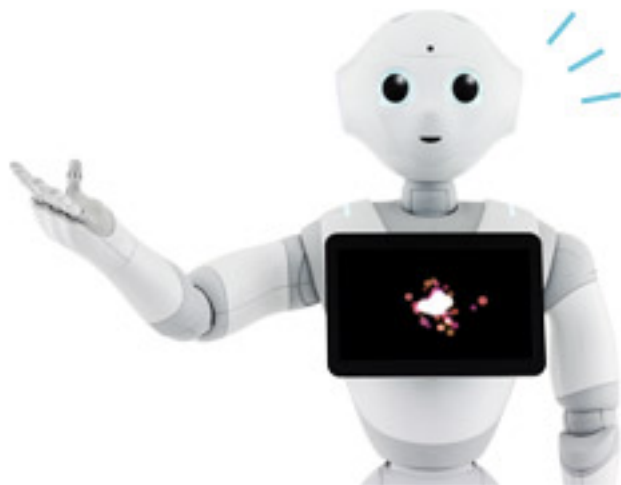
先ほどのOnlyOnceを下のようにつなぐ。



新しく作ったBoxを  
発火させた時とさせない時の違いを見る。

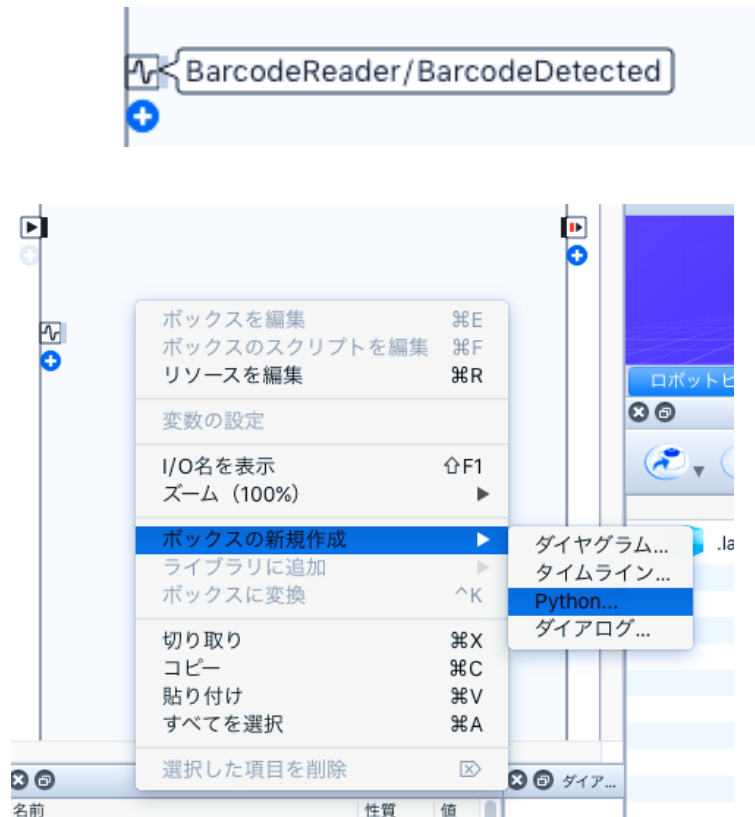
onload関数が  
実行される  
タイミングを  
見る。

# QRコード読み取り 新規PythonBoxの作成



# バーコードの読み取り(1/4)

1. メモリイベント”BarcodeReader/BarcodeDetected”を追加。
2. 背景を右クリックして  
“Boxの新規作成”の”Python”を選んで  
新しくPythonBoxを作る。



# バーコードの読み取り(2/4)

名前:  1

概要:

画像:

インプット/アウトプット/パラメータ

入力:  2

出力:  2

変数:

プラグイン

ライブラリの読み込み:   3

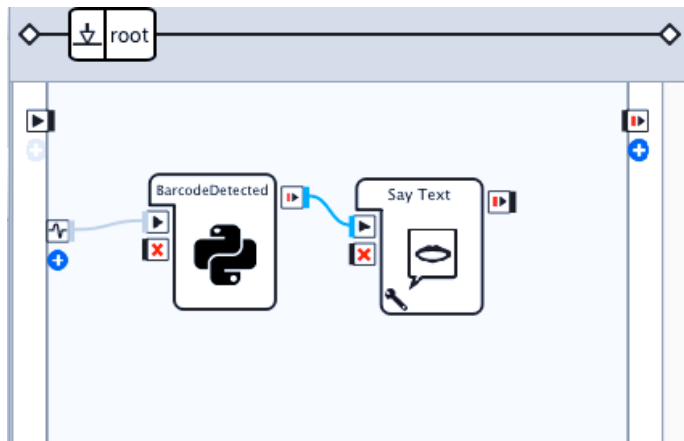
1. 名前を入力する。
2. 鉛筆マークを押してそれぞれ入力と出力の編集をする。  
入力:ダイナミック  
出力:文字列
3. OKを押す。
4. 以下のようなBoxが出来上がる。



色で信号の種類がわかる  
黒：バン  
青：文字列  
黄：数  
灰色：ダイナミック

# バーコードの読み取り(3/4)

左のように  
Pythonコードを書き、  
SayTextにつないで  
確認してみる。  
(ダイアログで確認)

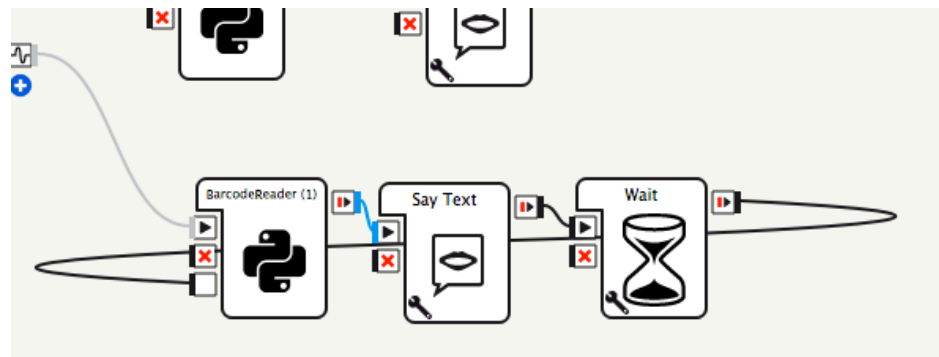


```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4
5     def onLoad(self):
6         #put initialization code here
7         self.flag = True
8         pass
9
10    def onUnload(self):
11        #put clean-up code here
12        pass
13
14    def onInput_onStart(self, p):
15        #self.onStopped(p) #activate the output of the box
16        if self.flag :
17            self.flag = False
18            self.onStopped(p[0][0])
19        pass
20
21    def onInput_onStop(self):
22        self.onUnload() #it is recommended to reuse the clean-up as
23        self.onStopped(p) #activate the output of the box
```

## 【余裕のある方向け】

喋った後、もう一度QRコードを読み取れるようにする。

- 入力を追加。  
(例はresetという入力を追加している。)
- 左のように繋ぐ。  
waitを入れるとエラーが起きない。
- 中身に以下を追加する。



```
def onInput_reset(self):
    self.flag = True
```



なぜ、

```
def onInput_onStart(self, p):  
    if self.flag:  
        self.flag = false  
        self.onStopped(p[0][0])
```

なのか。

出てくる値は

CodeData = [ Data, Position ]

また、

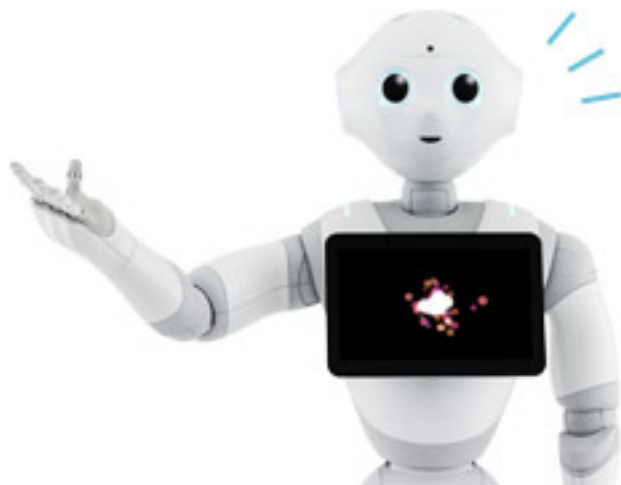
Position = [ [x0, y0], [x1, y1],  
 [x2, y2], [x3, y3] ]

となっている。

Dataが読み取ったデータで、  
Positionは読み取った映像中の  
QRコードの位置を表す。

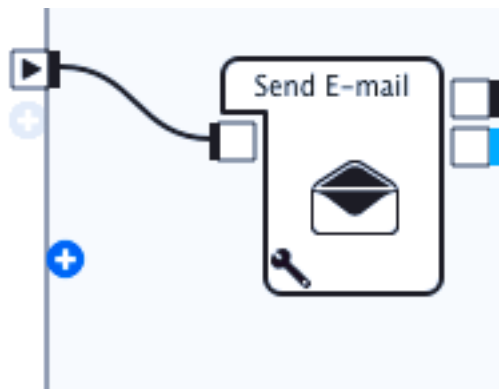
<http://doc.aldebaran.com/2-4/naoqi/vision/albarcodereader-api.html>

メールを送る



ペッパー用のメールアカウントを  
作ってあげないと  
セキュリティ上の問題あり。

テスト用アカウント  
test.mail.pep@gmail.com  
Pass:Pepper81  
SMTP address:  
smtp.gmail.com  
Port number:587



変数を Send E-mail に設定

変数

From	test.mail.pep@gmail
Password	.....
To	test.mail.pep@gmail
Subject	test
Contents	test
Attachment	
SMTP address	smtp.gmail.com
Port number	587

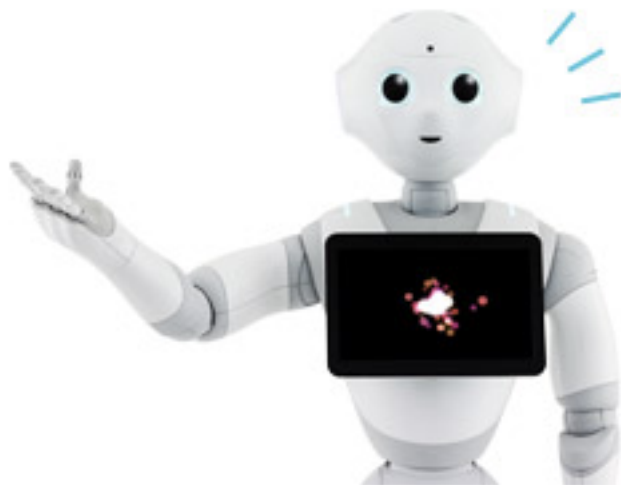
☒ ロボット上の変数の自動更新

初期値に戻す

キャンセル OK

SMTPやポート番号は調べてあげるとでてくる。  
Gmailの場合は連携を許可しないと使えない。

PythonでBoxを改造する

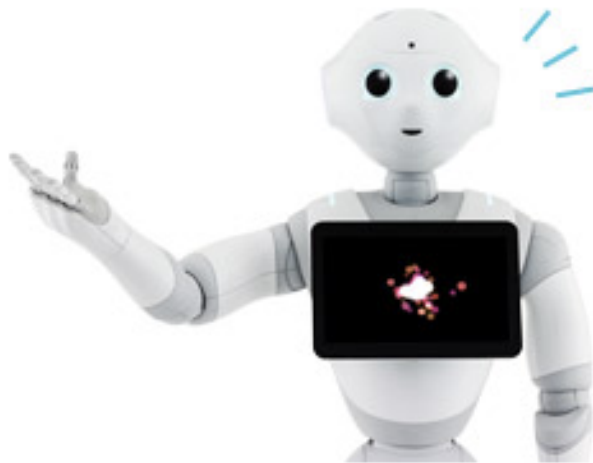


今まで紹介したものを使って来客者を伝える簡単なアプリを作る

1. QRコードで名前と会いに来た人のメールアドレスを読み取る。
1. メールで来客を伝える。

と言ったものを作る。

今回読み取るQRコードの情報は  
Pepper/test.mail.pep@gmail.com



## ヒント1 Send E-mailの中をしてみる。

スパナマークで設定した値はself.getParameter()で得ていることが分かる。

```
43 def onInput_onSend(self):  
44     sEmailUser = self.getParameter("From")  
45     aTo = self.getParameter("To").split(";")  
46     sSubject = self.getParameter("Subject")  
47     sText = self.getParameter("Contents")  
48     sAttachedFilePath = self.getParameter("Attachment")  
49     sPwd = self.getParameter("Password")  
50     sSmtp = self.getParameter("SMTP address")  
51     sPort = int( self.getParameter("Port number") )  
52     try:  
53         sPort = int( sPort )  
54         bValidPort = ( sPort >= 0 and sPort <= 65535 )
```

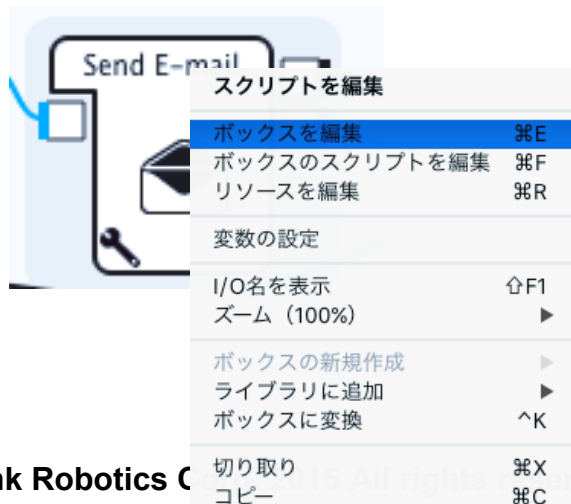
それぞれの変数に  
新しく値を  
入れられるように  
コードを  
書き換えれば  
良い。  
aToとsTextの  
右側を書き換える。

## ヒント2 情報の受け取り方

Send E-mail BoxのonSendの入力を文字列にして、バーコードで読み取った情報を受け取る。

Boxの上で右クリックして”ボックスを編集”から入力の種類を編集可能

バーコードに  
日本語があるときは  
エンコードが必要！



```
43 def onInput_onSend(self, p):
```

上のように、pを引数に加えてあげると  
入力から入ってきた値がpの中に入る。  
(必ずしも”p”である必要はない)



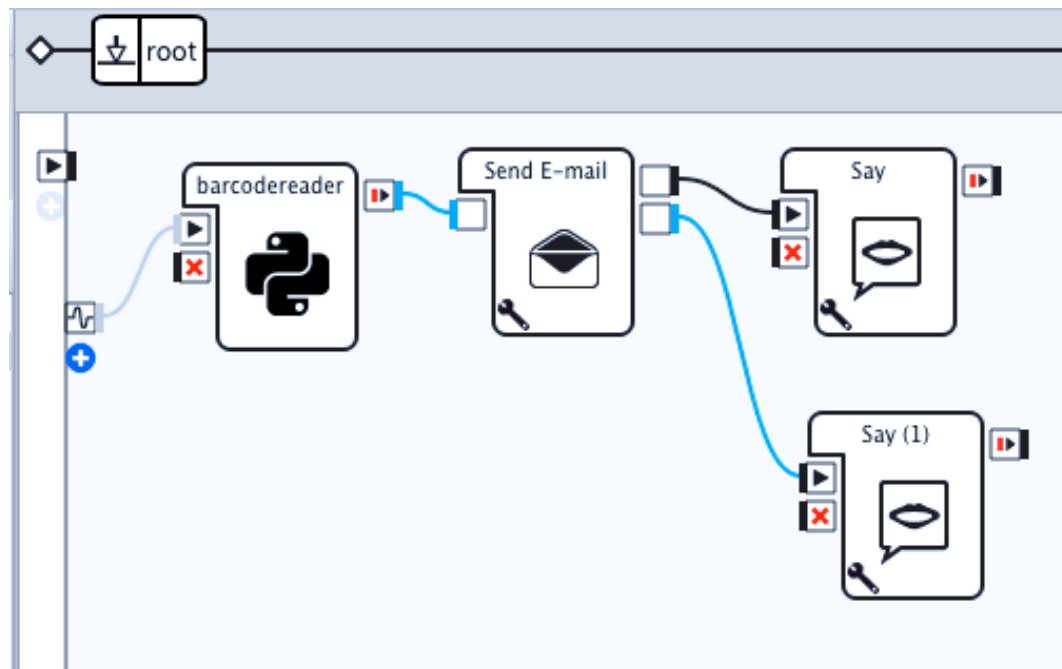
## ヒント3 Send E-mainの改造の一例

今回、バーコードから読み取った情報は  
"/"で区切られているため"/"でsplitすれば良い

```
43 def onInput_onSend(self, p):  
44     data = p.split("/")  
45     aTo = data[1].split(";")  
46     sEmailUser = self.getParameter("From")  
47     sSubject = self.getParameter("Subject")  
48     sText = data[0] + "さんがいらっしゃいました。"  
49     sAttachedFilePath = self.getParameter("Attachment")  
50     sPwd = self.getParameter("Password")  
51     sSmtip = self.getParameter("SMTP address")  
52     sPort = int( self.getParameter("Port number") )  
53     try:  
54         sPort = int( sPort )  
55         bValidPort = ( sPort >= 0 and sPort <= 65535 )  
56     except:  
57         bValidPort = False  
58     if( not bValidPort ):
```

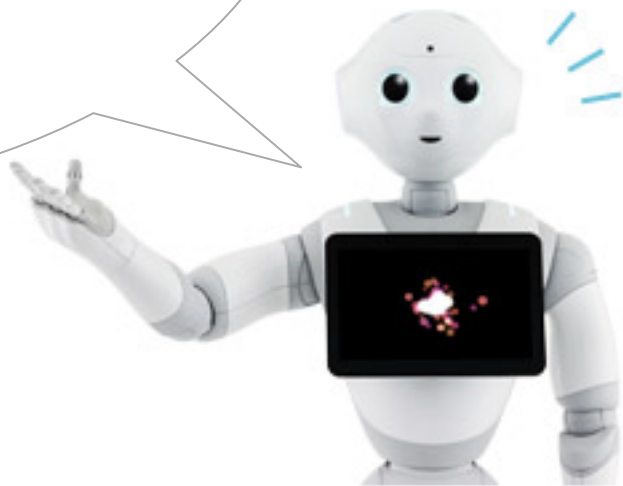
コードから  
宛先メールアドレスは  
";"で区切れば  
何人でも送れることが  
わかる

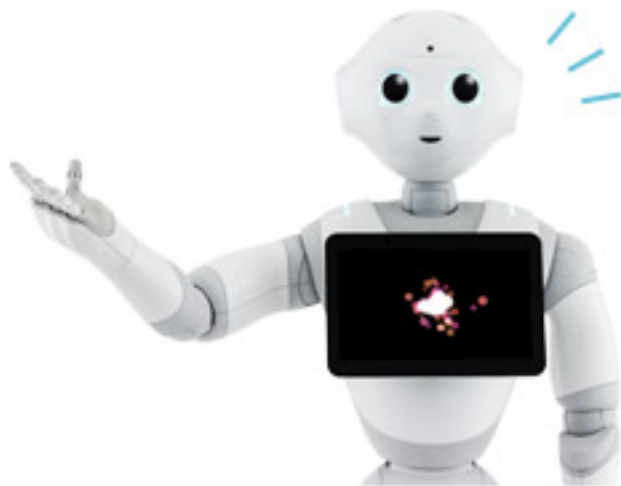
## ヒント4 完成の一例



おつかれさまでした！  
これにて  
**WS"既存のBoxをPythonで書き換えてみよう”  
は終わりになります。**

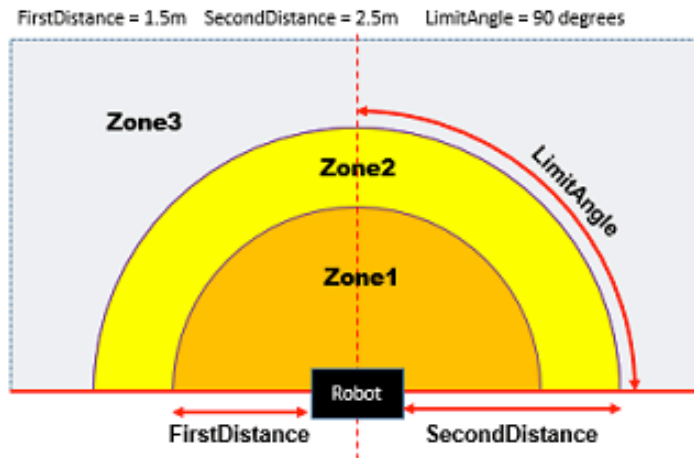
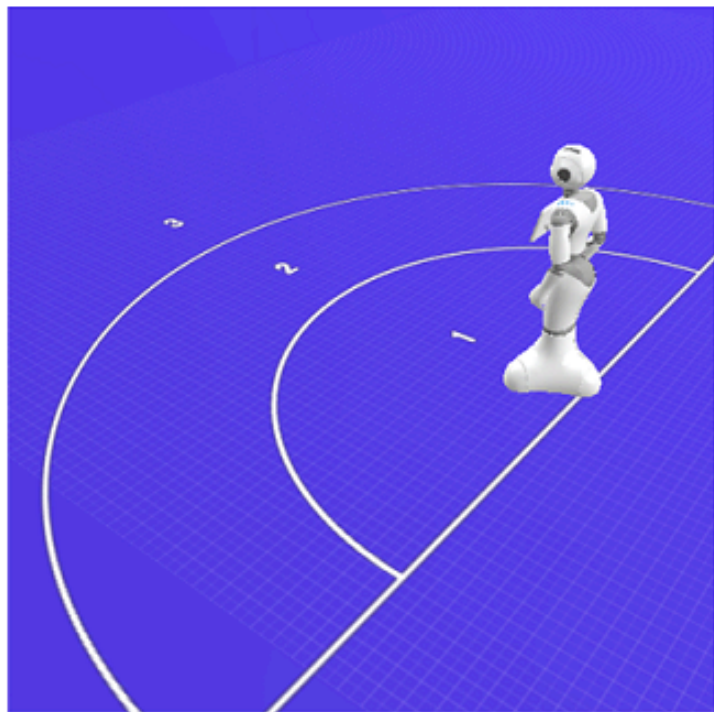
タッチアンドトライで質問もしてみてください





おまけ  
エンゲージメント  
ゾーンとは?

# エンゲージメントゾーン(1/3)



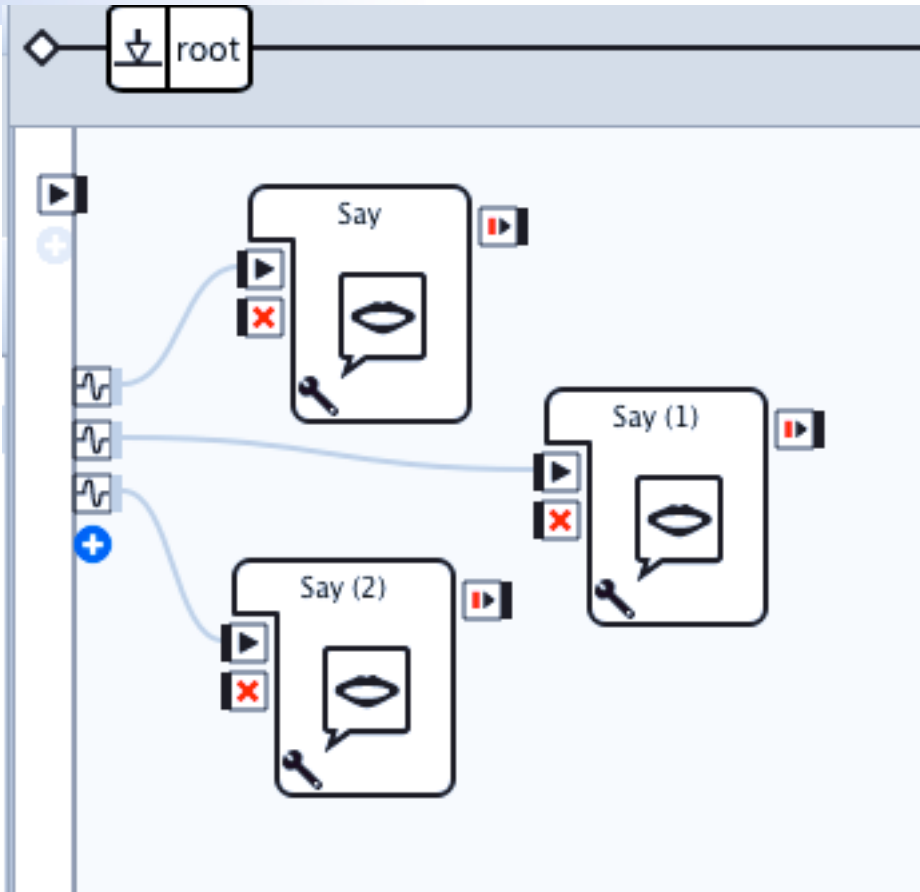
デフォルトでは...

FirstDistance --- Pepperから1.5m

SecondDistance --- Pepperから2.5m

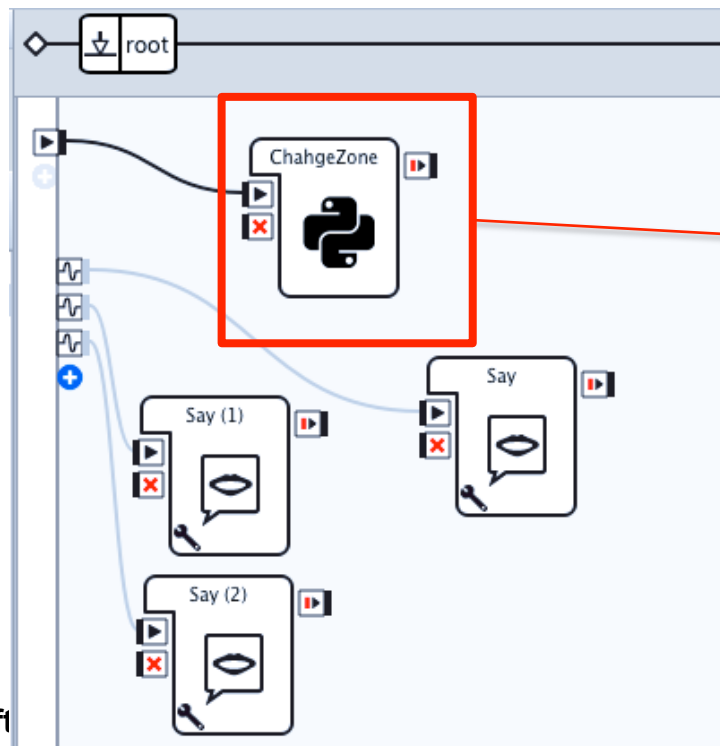
LimitAngle --- 90°

1. メモリイベント
  - PersonEnteredZone1
  - PersonEnteredZone2
  - PersonEnteredZone3を追加する。
2. SayBoxをそれぞれつなぐ
3. SayBoxの中にそれぞれに対応する言葉を入れてあげる。
4. 実行してPepperの周りを歩き回る。



## エンゲージメントゾーン(3/3)

エンゲージメントゾーンの定義をAPIによりカスタマイズしてみる。



```
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
        zone = ALProxy("ALEngagementZones")
        zone.setFirstLimitDistance(1.0)
        zone.setSecondLimitDistance(2.0)
        zone.setSecondLimitDistance(4.0)

        self.onStopped()
        pass
```

```
    def onInput_onStop(self):
        self.onUnload() #it is recommended to reuse the cle
        self.onStopped() #activate the output of the box
```