

免責事項

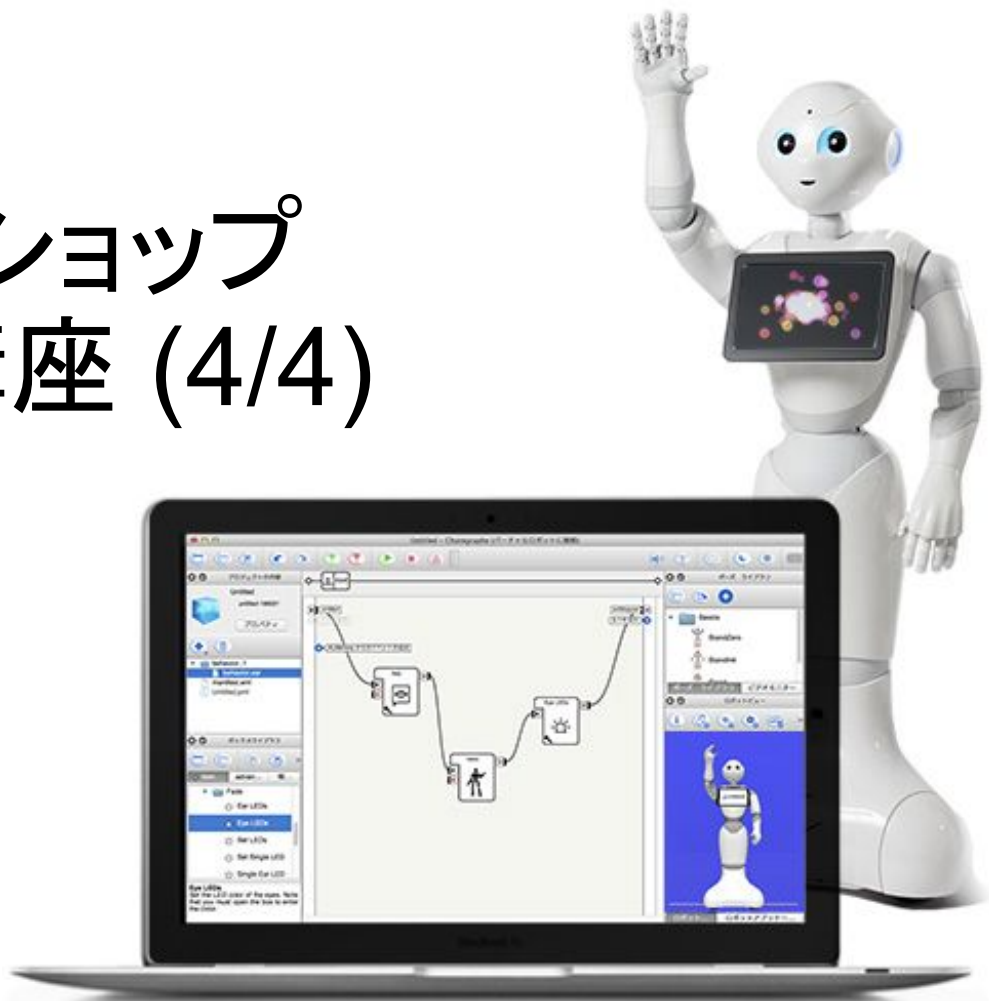
このワークショップは
アトリエのスタッフが作成したものであり
ソフトバンクロボティクス(株)公式のものでは
ないことをご承知ください。



アトリエ秋葉原

Pepper ワークショップ Python 講座 (4/4)

最終更新: 2018/12/22



実体験とコミュニティで開発を促進する

アトリエ



相互
促進

コミュニティ

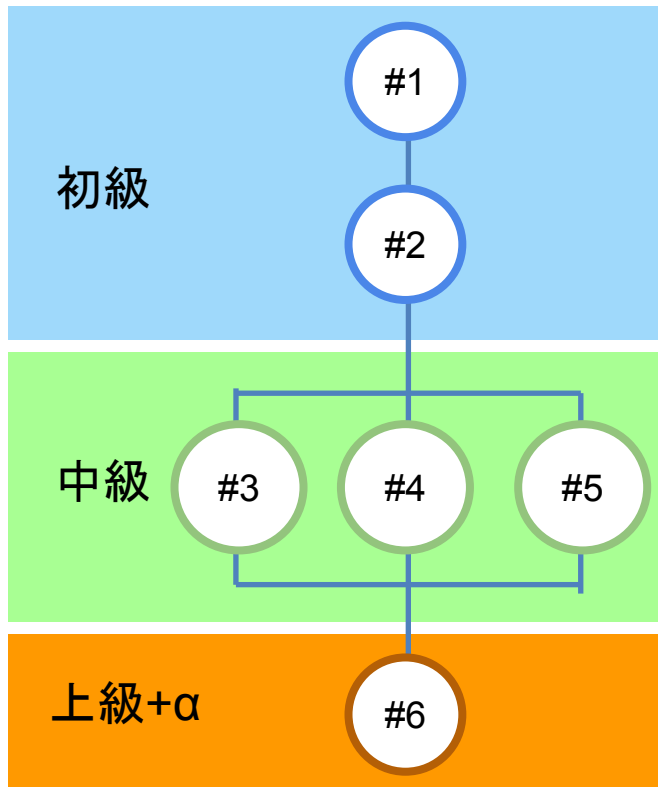


- Pepperのアプリ開発を実体験

- 経験や知見を共有

アトリエ秋葉原のサービス

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
検証や
打ち合わせの利用も可

1週間の予定

月	タッチアンドトライ
火	貸し切り(有料)
水	Pepper for Biz説明会 & タッチアンドトライ
木	貸し切り(有料)
金	タッチアンドトライ & ワークショップ
土日	タッチアンドトライ & ワークショップ

ワークショップ番外編について

アトリエスタッフが製作したオリジナルワークショップ

- ・外部APIとの連携を試そう(天気とTwitter)
- ・Pepperのディレクトリ構造を知ろう
- ・ペッパーリモコンを作ろう
- ・NAOqi2.5.5とNAOqi2.4.3の違い
- ・Pepperで学ぶPython基礎講座その1(変数の扱い方)
- ・Pepperで学ぶPython基礎講座その2(制御文を知る)
- ・Pepperで学ぶPython基礎講座その3(関数を作る)
- ・Pepperで学ぶPython基礎講座その4(BOXを編集)
- ・Azure Face APIで顔認証 ハンズオン
- ・Pepperで学ぶ、はじめてのWatson (Visual Recognition編)
- ・Pepper x TensorFlow 入門

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、

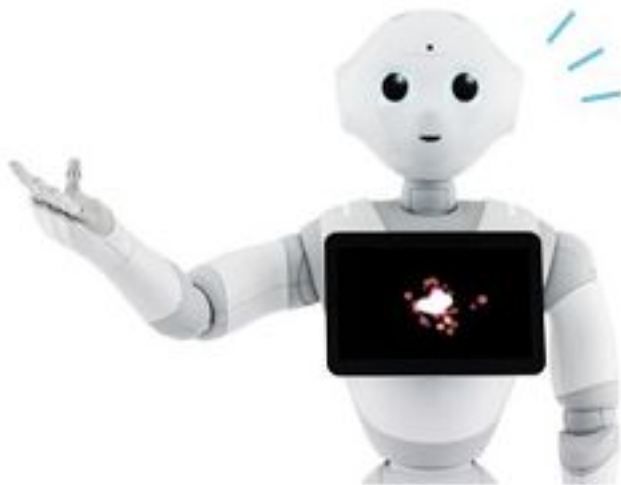
＊ ＊ と申します。



1. 例外処理について
2. BOXを読んで基礎を学ぼう
3. ALProxy・入力・変数・出力
4. BOXを編集しよう



例外处理



例外処理

- 例外(実行中に検出されたエラー)処理コードを書くことで予期せぬ事故を未然に防ぐ
- try句: 例外が起こりそうな処理
- except句: 例外が起こった場合の処理
- else句: 例外が検出されなかった場合の処理(except句の後ろにおく)
- finally句: どの場合にも実行する処理



例外処理

～例～

```
try:
    print(x/y)
except:
    print("Error!")
else:
    print("Success!")
finally:
    print("Completed!")
```

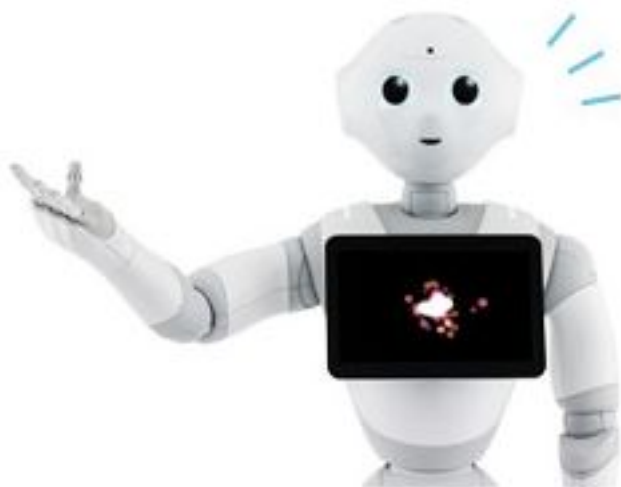
～実行結果～

x = 10,y = 0の時
Error!
Completed!

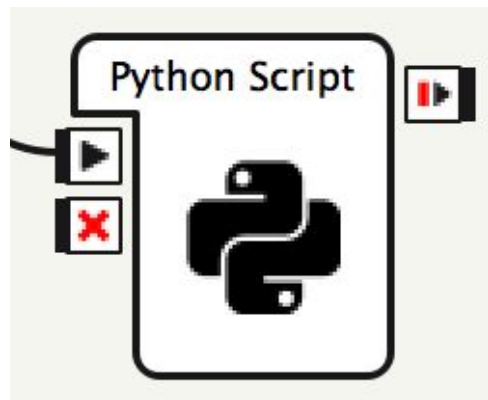
x = 10,y = 2の時
5
Success!
Completed!



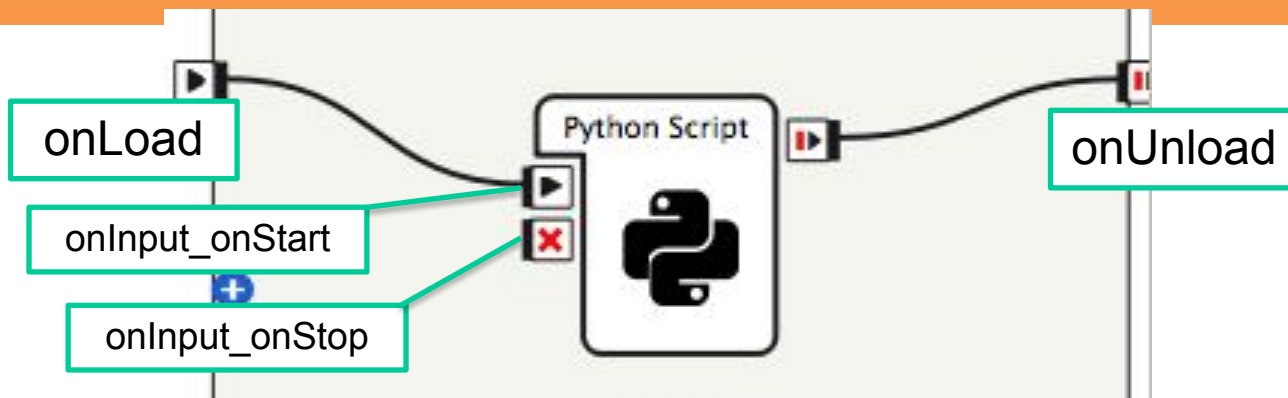
ChoregrapheでPythonを記述する場合について(おさらい)



使うボックスはこれ



Python Script boxの関数(おさらい)



__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(おさらい)

```
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

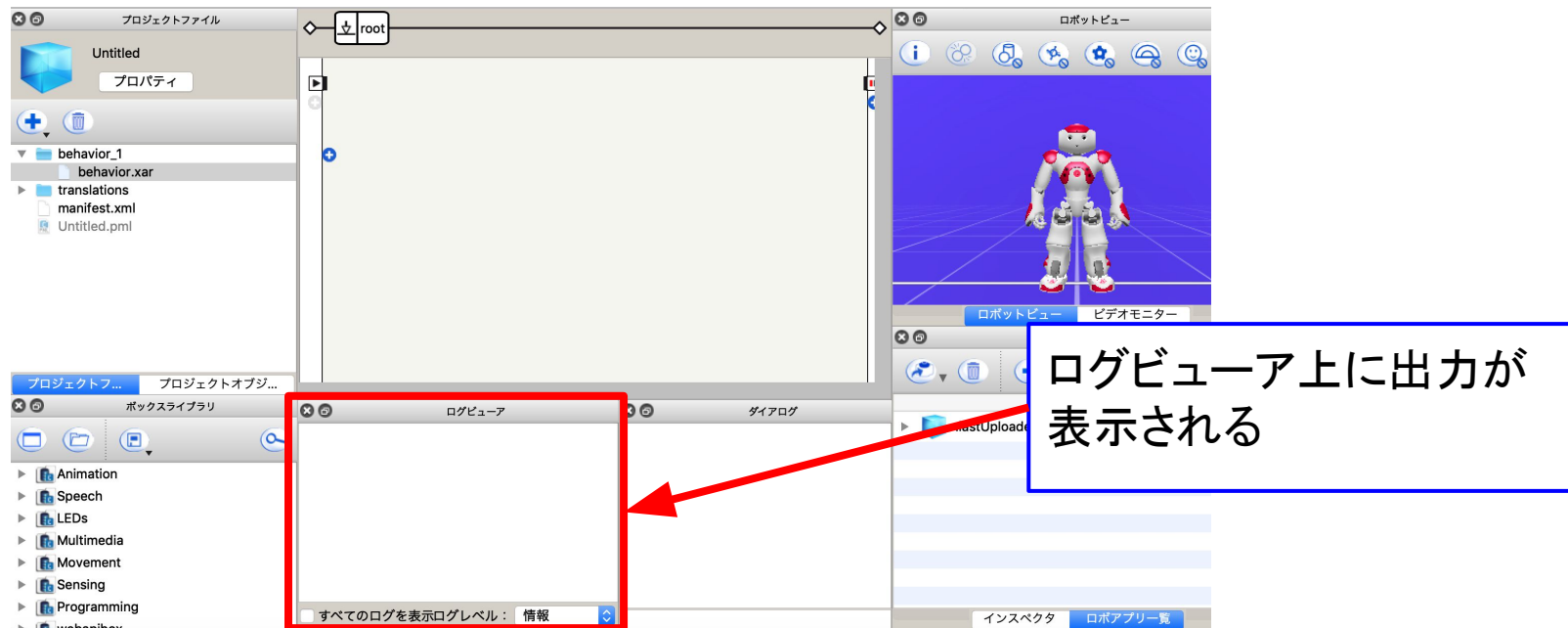
    def onInput_onStart(self):
        #self.onStopped() #activate the output of the box
        pass

    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
```

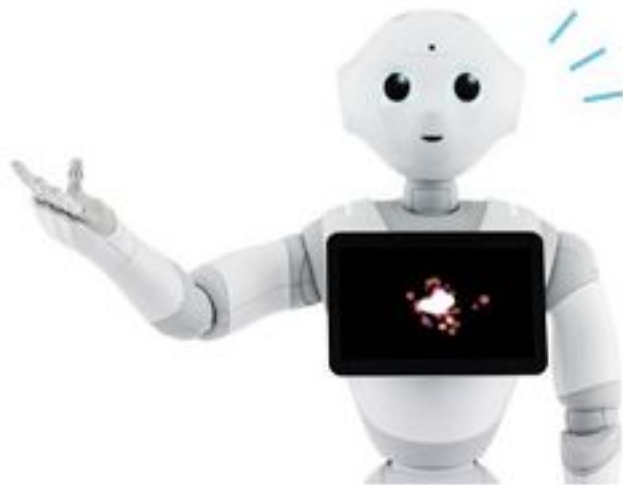
ここに書く

Choregraphe上で出力を得るに(おさらい)

Pythonでは標準的な出力を得るためには`print()`関数を用いるのが一般的であるが、Choregraphe上で出力を得るためには`self.logger.info()`関数を用いる。

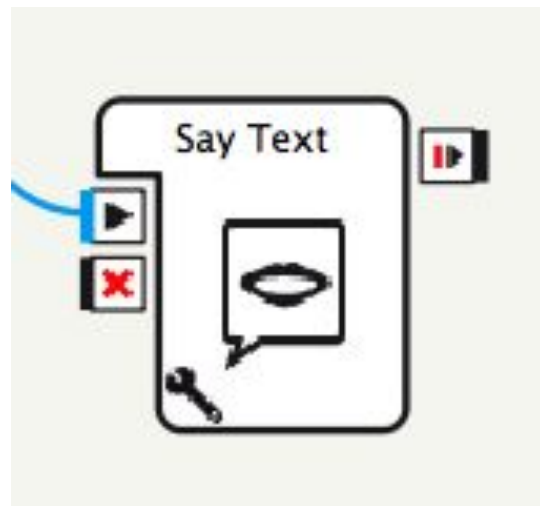


BOXを読んで
基礎を学ぼう



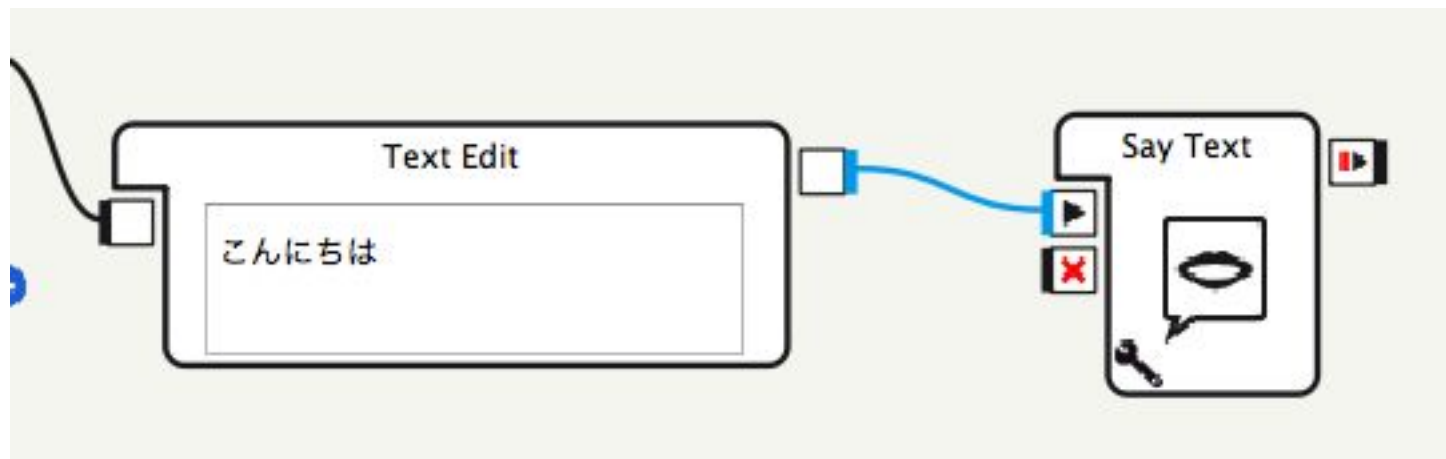


使うボックスはこれ！



Say Text boxとは

- 入力された言葉を話す



ALProxyとは

- PythonでPepperの機能にアクセスするためのnaoqiのモジュール

```
def __init__(self):  
    GeneratedClass.__init__(self, False)  
    self.tts = ALProxy('ALTextToSpeech')
```

使うAPI

- 使うAPIによって色々な関数がある

`class ALTextToSpeechProxy`

- `ALTextToSpeechProxy::getAvailableLanguages()`
- `ALTextToSpeechProxy::getAvailableVoices()`
- `ALTextToSpeechProxy::getLanguage()`
- `ALTextToSpeechProxy::getParameter()`
- `ALTextToSpeechProxy::getSupportedLanguages()`

検索するとチュートリアル
がでてくる



入力と パラメータと 出力



- 入力がデータを持つ場合は関数の中に引数として記述する



ダイナミック(灰)	数や文字列のリストなど
“バン”(黒)	信号のみが送られる
数(黄)	数字を持った状態で送られる
文字列(青)	文字列を持った状態で送られる

```
def onInput_onStart(self, p):  
    self.bIsRunning = True  
    try:  
        sentence = "\RSPD=" + str( self.getParameter(  
        sentence += "\VCT=" + str( self.getParameter(  
        sentence += str(p)  
        sentence += "\RST\ "
```

入力=引数なので
文字で置き換えて関数を作る



- self.getParameter(“変数名”)で値を取得



```
str( self.getParameter("Speed (%)") ) + "\ "  
str( self.getParameter("Voice shaping (%)") ) + "\ "  
"
```

- self.変数名(値)で出力される



onStoppedから信号を出力したい
→self.onStopped()で出力

```
def onInput_onStart(self, p):
    self.bIsRunning = True
    try:
        sentence = "\RSPD=" + str( self.getParameter("Speed
        sentence += "\VCT=" + str( self.getParameter("Voice
        sentence += str(p)
        sentence += "\RST\ "
        id = self.tts.post.say(str(sentence))
        self.ids.append(id)
        self.tts.wait(id, 0)
    finally:
        try:
            self.ids.remove(id)
        except:
            pass
    if( self.ids == [] ):
        self.onStopped() # activate output of the box
        self.bIsRunning = False
```



読んでみよう



—Say Text boxを読む—

```
def __init__(self):  
    GeneratedClass.__init__(self, False)  
    self.tts = ALProxy('ALTextToSpeech')  
    self.ttsStop = ALProxy('ALTextToSpeech', True)
```

ALTextToSpeech
という機能を使用

```
def onInput_onStart(self, p):  
    self.bIsRunning = True  
    try:  
        sentence = "\RSPD="+ str( self.getParameter("Speed (%)") ) + "\ "  
        sentence += "\VCT="+ str( self.getParameter("Voice shaping (%)") ) + "\ "  
        sentence += str(p)  
        sentence += "\RST\ "  
        id = self.tts.post.say(str(sentence))  
        self.ids.append(id)  
        self.tts.wait(id, 0)
```

文字列に
変換する関数

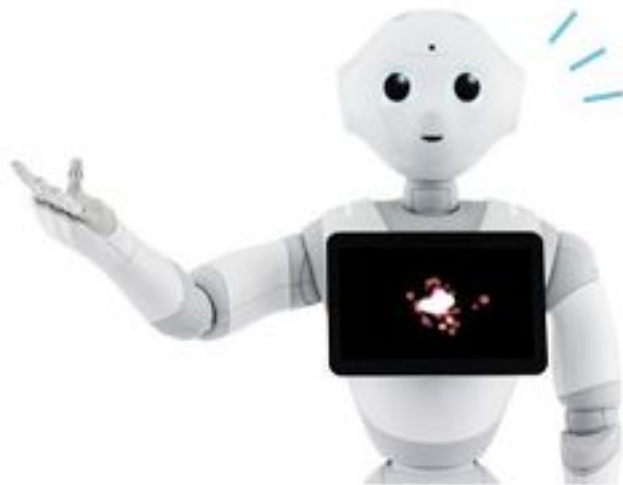
変数の値を取得

sentence = “\RSPD=100\ \VCT=100 \こんにちは\RST\”

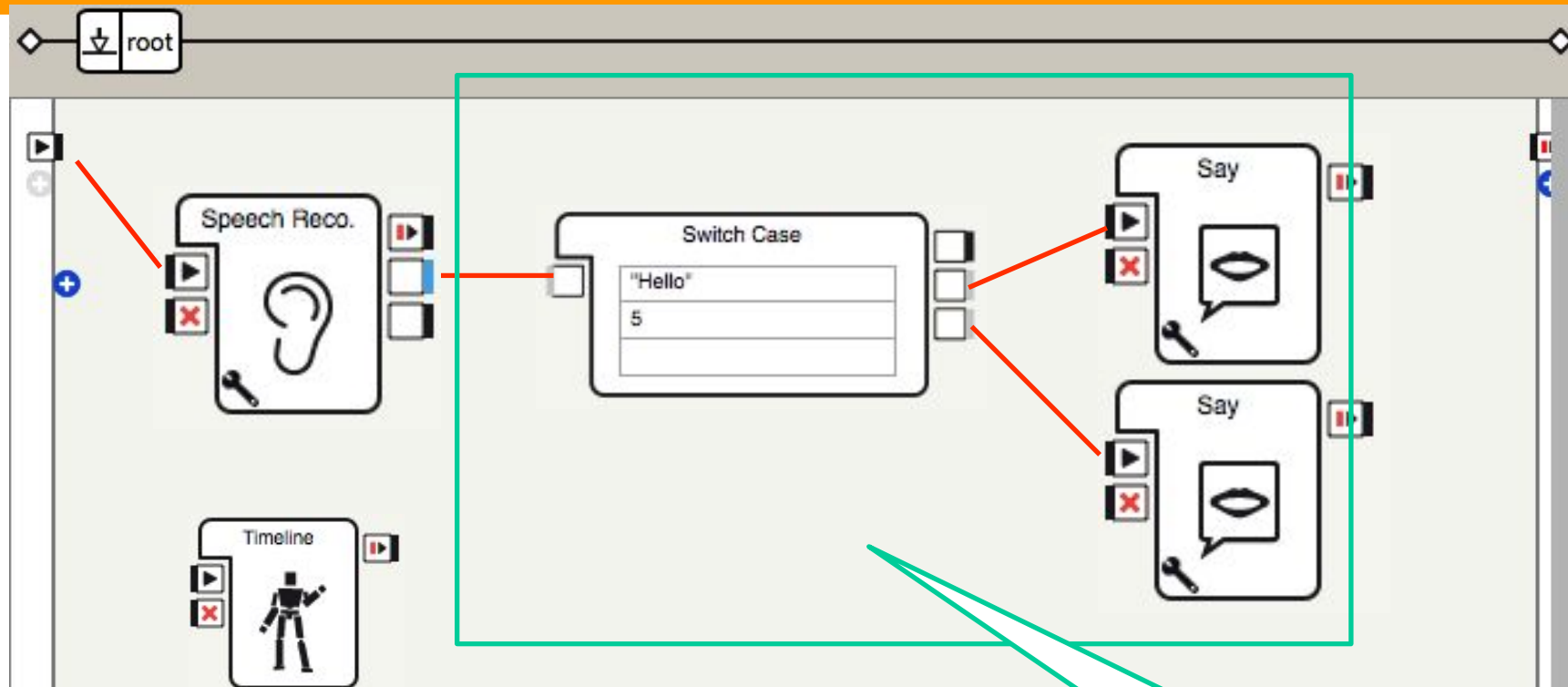
ALTextToSpeechの
sayという関数を使用



演習！
BOXを編集しよう

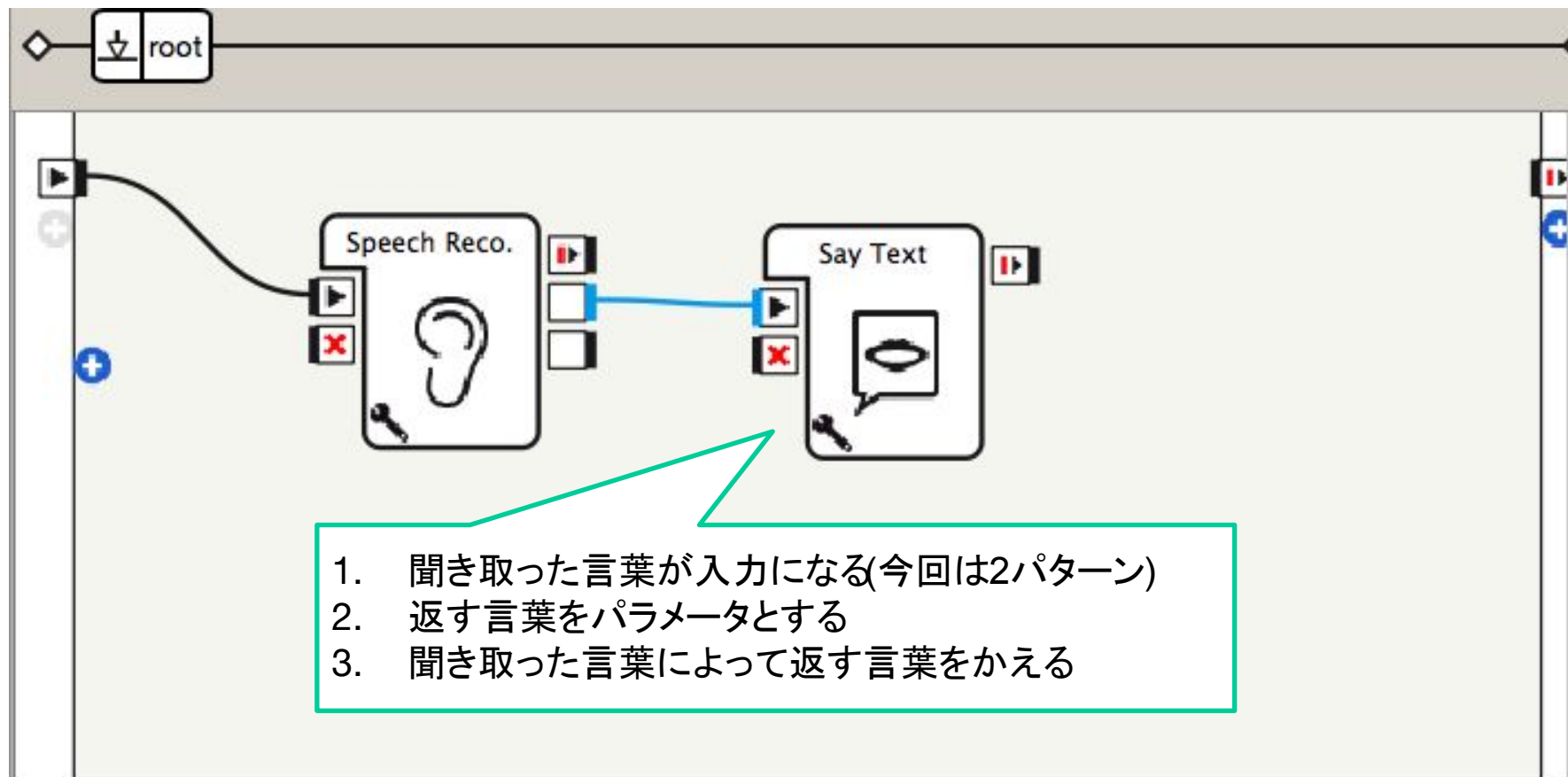


会話をするBOX before



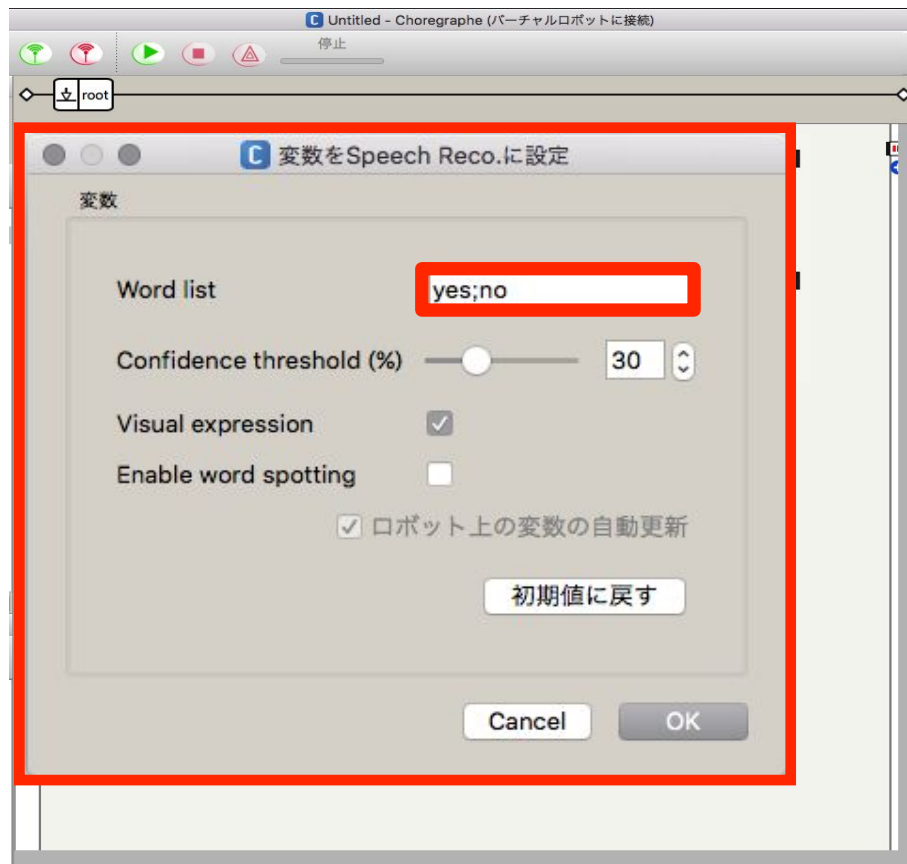
sayTextboxを
書き換える

会話をするBOX after

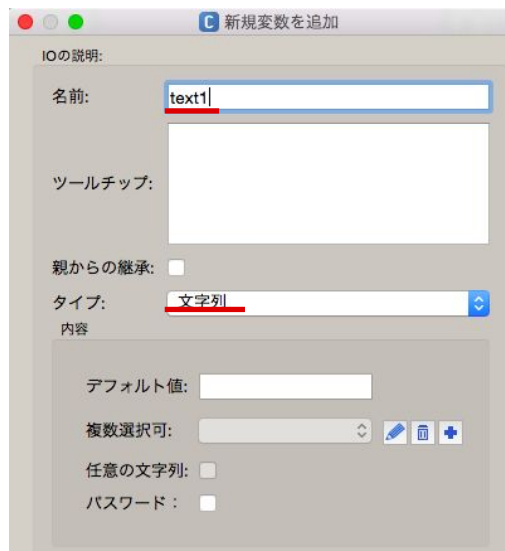
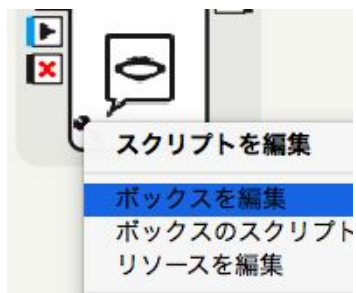


会話をする “Speech Reco.” の設定

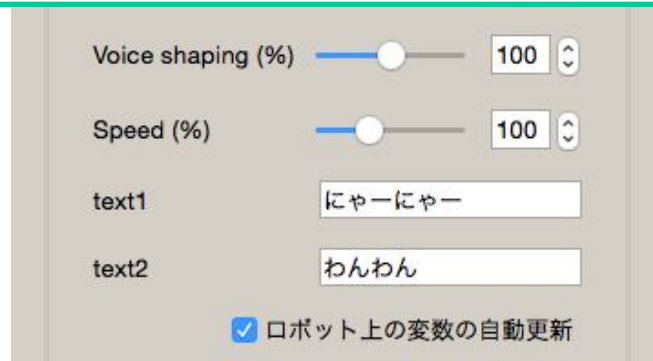
1. **Speech Reco**の
パラメーターボタンで設定
2. 聞かせたい言葉を**Wordlist**を設定
今回は「ねこ;いぬ」
3. **単語;単語** いうように
半角セミコロンを挟めると
いくつもの単語を認識できる
(今回は2つ)



パラメーターの追加



！今回は2パターンの返答なので
パラメーターを2つ作る(text1,text2)



どこを書き換えたら良いか？

```
def onInput_onStart(self, p):
    self.bIsRunning = True
    try:
        sentence = "\RSPD=" + str( self.getParameter("Speed (%)") ) + "\ "
        sentence += "\VCT=" + str( self.getParameter("Voice shaping (%)") ) + "\ "
        sentence += str(p)
        sentence += "\RST\ "
        id = self.tts.post.say(str(sentence))
        self.ids.append(id)
        self.tts.wait(id, 0)
    finally:
        try:
            self.ids.remove(id)
        except:
            pass
    if( self.ids == [] ):
        self.onStopped() # activate output of the box
        self.bIsRunning = False
```



どこを書き換えたら良いか？

```
def onInput_onStart(self, p):
    self.bIsRunning = True
    try:
        sentence = "\RSPD=" + str( self.getParameter("Speed (%)") ) + "\ "
        sentence += "\VCT=" + str( self.getParameter("Voice shaping (%)") ) + "\ "
        sentence += str(p)
        sentence += "\RST\ "
        id = self.tts.post.say(str(sentence))
        self.ids.append(id)
        self.tts.wait(id, 0)
    finally:
        try:
            self.ids.remove(id)
        except:
            pass
        if( self.ids == [] ):
            self.onStopped() # activate output of the box
            self.bIsRunning = False
```

sentenceに各入力に対応する
文字列を連結すればよい
ここにif文を書きましょう！



書き換えよう！

```
def onInput_onStart(self, p):  
    self.bIsRunning = True  
    try:  
        sentence = "\RSPD="+ str( self.getParameter("Speed (%)") ) + "\ "  
        sentence += "\VCT="+ str( self.getParameter("Voice shaping (%)") ) + "\ "
```

ここに書く！

```
    else:  
        sentence += "なんていったかわかりません！"  
    sentence += "\RST\ "  
    id = self.tts.post.say(str(sentence))  
    self.ids.append(id)  
    self.tts.wait(id, 0)
```



書き換えよう！ 答え

```
def onInput_onStart(self, p):  
    self.bIsRunning = True  
    try:  
        sentence = "\RSPD=" + str( self.getParameter("Speed (%)") ) + "\ "  
        sentence += "\VCT=" + str( self.getParameter("Voice shaping (%)") ) + "\ "  
        if p == "ねこ":  
            sentence += str(self.getParameter("text1"))  
        elif p == "いぬ":  
            sentence += str(self.getParameter("text2"))  
        else:  
            sentence += "なんていったかわかりません！"  
        sentence += "\RST\ "  
        id = self.tts.post.say(str(sentence))  
        self.ids.append(id)  
        self.tts.wait(id, 0)
```



一応用問題一



左記のように変数を設定し、
pepperが人の発言に対し対応する返
答をするプログラムを作成せよ

```
self.getParameter("human").split(";")
```

...変数humanの文字列"ねこ;いぬ;にわとり"を";"で
区切って["ねこ","いぬ","にわとり"]とリストを生成

```
len(リスト)
```

...リストの長さを数字で出力

ヒント:
for文、if文、range関数を使います
その他の関数を用いると最短1行で書き終わります!



一応用問題一 答え

```
def onInput_onStart(self, p):  
    self.bIsRunning = True  
    humanlist = self.getParameter("human").split(";")  
    pepperlist = self.getParameter("pepper").split(";")  
    try:  
        sentence = "\RSPD=" + str( self.getParameter("Speed (%)") ) + "\ "  
        sentence += "\VCT=" + str( self.getParameter("Voice shaping (%)") ) + "\ "  
        for i in range(len(humanlist)):  
            if p == humanlist[i]:  
                sentence += pepperlist[i]  
        sentence += "\RST\ "  
        id = self.tts.post.say(str(sentence))  
        self.ids.append(id)  
        self.tts.wait(id, 0)
```

入力パラメータの名前を"Text"から変更

別解も参照!!



一応用問題一 答え(別解)

- ・最短一行

```
sentence += pepperlist[humanlist.index(p)]
```

- ・enumerate関数を用いた場合

```
for i,j in enumerate(humanlist):  
    if p == j:  
        sentence += pepperlist[i]
```

- ・zip関数を用いた場合

```
for i,j in zip(humanlist , pepperlist):  
    if p == i:  
        sentence += j
```





ホーム アトリエ秋葉原とは 利用予約 アトリエ サテライト SDK FAQ リンク集

「Arduinoファンもくもく会#019 with アトリエ秋葉原(Pepper開発体験)」 イベントレポート



先日アトリエ秋葉原にてArduinoファンもくもく会#019 with アトリエ秋葉原(Pepper開発体験)を開催... [Read More »](#)

いいね！ 0

Tweet

イベント イベントレポート

AtelierStaff

検索

ダウンロード

ワークショップ教材
80点のモーションライブラリ

最近の投稿

「Arduinoファンもくもく会#019 with アトリエ秋葉原(Pepper開発体験)」 イベントレポート

今回は温度センサとPepperを連動させたラズパイハンズオン！

Pepper アトリエ秋葉原 with SoftBank

「アトリエ秋葉原 ブログ」で検索

- ・ワークショップのスライドをダウンロードできます

- ・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ます



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




435 投稿 661 フォロワー Pepperに関する情報が集まっています。現在435件の投稿があります。また661人のユーザーが Pepperタグをフォローしています。

最近いいねされた投稿

 doki_k が2018/05/21に投稿 👍 6
AWS IoT で Pepper と RaspberryPi 間を MQTT でやり取りしてみる
Python RaspberryPi Pepper awsIoT

 hws-hitorobo が2015/07/23に投稿 👍 18
PEPPER 目のLEDを複雑に光らせる
Pepper Choregraphe

 yuka_nm が2016/09/23に投稿 👍 26
Watson Speech to Text を使ってPepper同士は会話できるのか? ～シンプルな伝言からラップバトルへの挑戦まで
やってみた Pepper Watson SpeechToText

 JohnTomato が2015/12/15に投稿 👍 52
PepperとWatson SpeechToTextAPIを連携させて継続的な音声認識サービスを作ってみた
AdventCalendar Bluemix NAOqi Pepper Watson

 kakkey が2016/08/23に投稿 👍 1
長押ししてイベント発火させるボックスで、長押しのタイミングを分かるように改善してみた

Qiita

「Qiita pepper」で検索

・プログラミングに関する知識を
記録・共有するためのサービス

・Pepperタグに有志による
Pepperに関する様々な技術情報があります



アップロード済み すべて再生

≡ 並び替え



第17回 Pepperと外部センサーを繋いでみよう! (Mesh編)

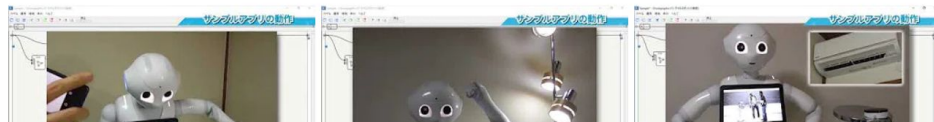
視聴回数 108 回・4 か月前

第16回 PepperにSlackから指示を送ってみよう!

視聴回数 149 回・6 か月前

第15回 Pepperでテレビ画面を制御してみよう! ...

視聴回数 47 回・7 か月前



Pepper Developer Network

「Pepper Developer Network」で検索

- ・SBR公式Youtubeチャンネル

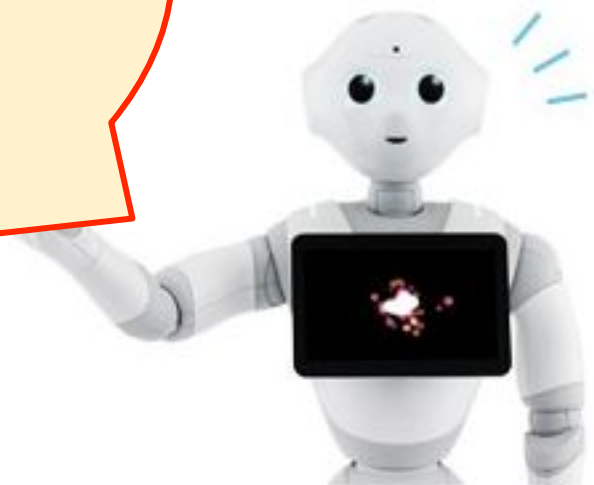
- ・ロボアプリ開発でよく使うパターンや知らないとハマっちゃうポイントについて説明しています。

お疲れ様でした

おつかれさまでした！
これにてPepperで学ぶPython講座その4
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

お帰りの際はアンケートの記入に
ご協力ください



<https://bitly.com/atelierakb>

