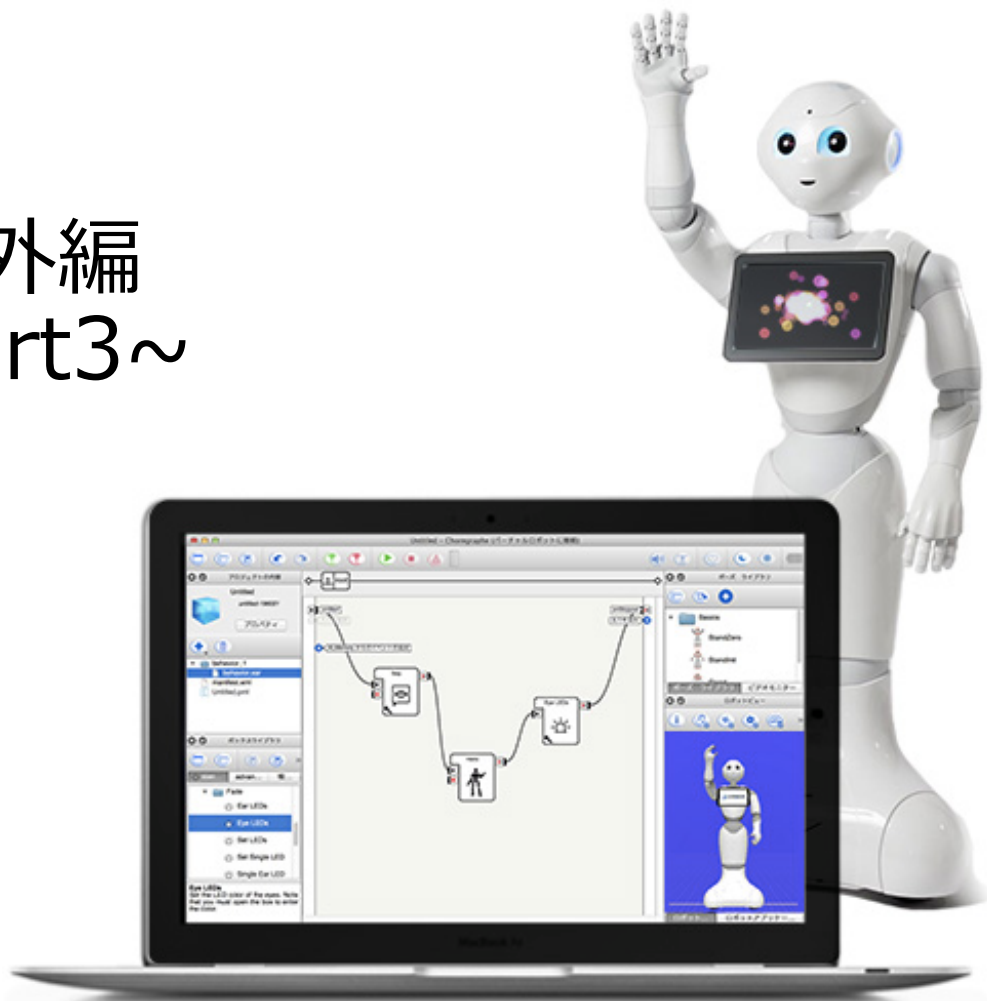


Atelier Akihabara

ワークショップ 番外編 Python入門~Part3~

2017/7/11

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

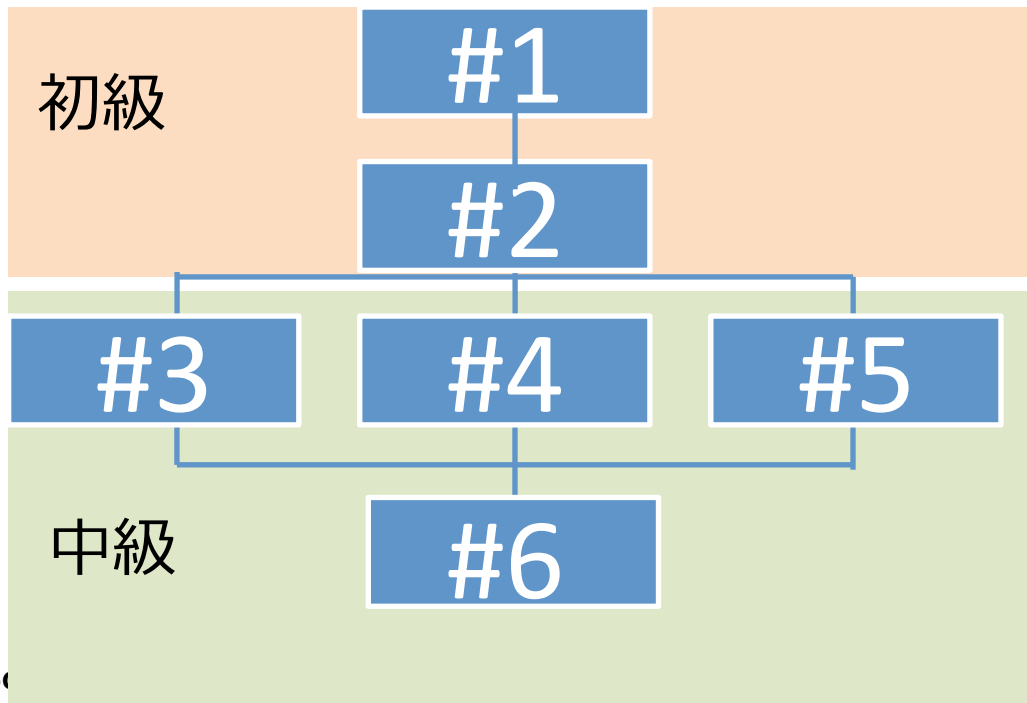


✓ 経験や知見を共有

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

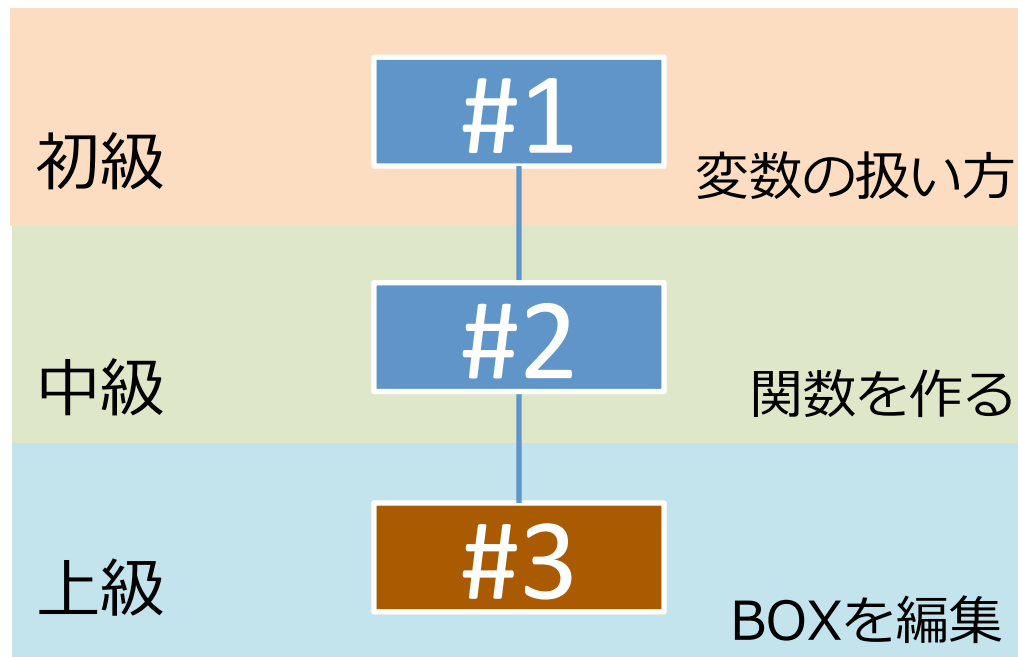
有志でPepperと開発スペースを
提供している
企業、大学、コミュニティスペース

秋葉原で回答できない質問は
各サテライトへ



スタッフが製作したオリジナルワークショップ°

pythonワークショップ



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

例：

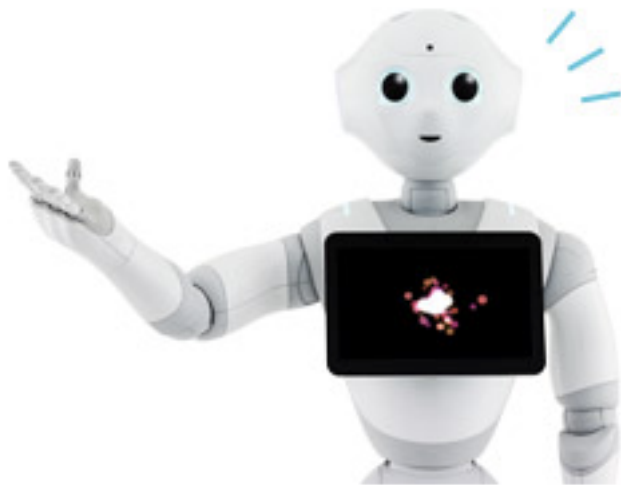
本日の案内を勤めさせていただきます、
大山 まりほ(Mariho Oyama)と申します。

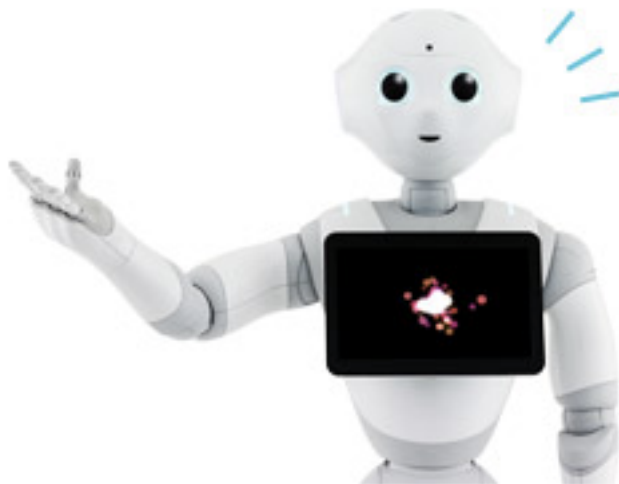
1. BOXを読んで基礎を学ぼう

ALProxy ・ 入力 ・ 変数 ・ 出力

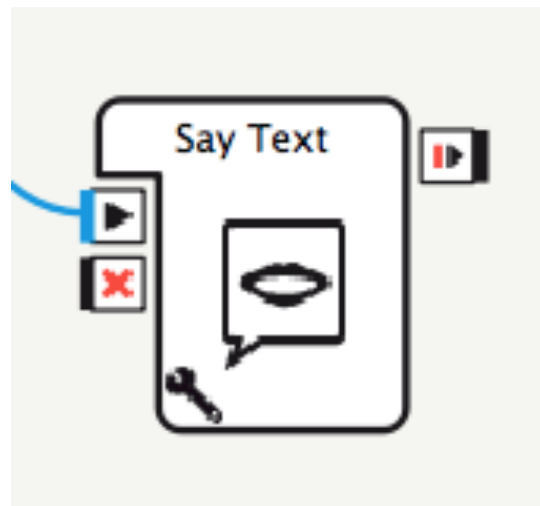
1. BOXを編集しよう

BOXを読んで
基礎を学ぼう



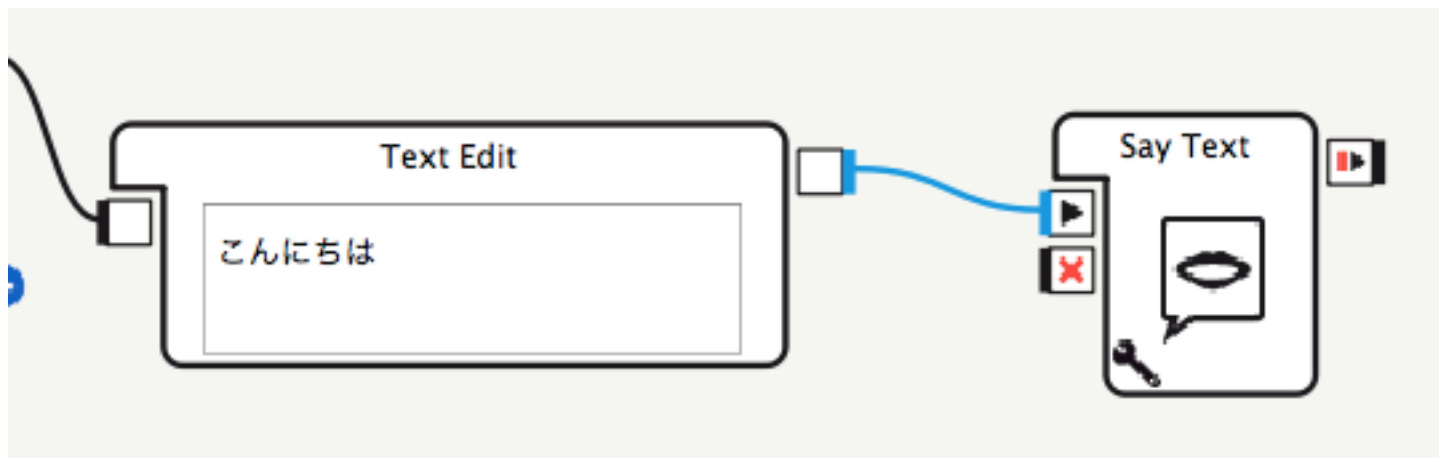


使うボックスはこれ！



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

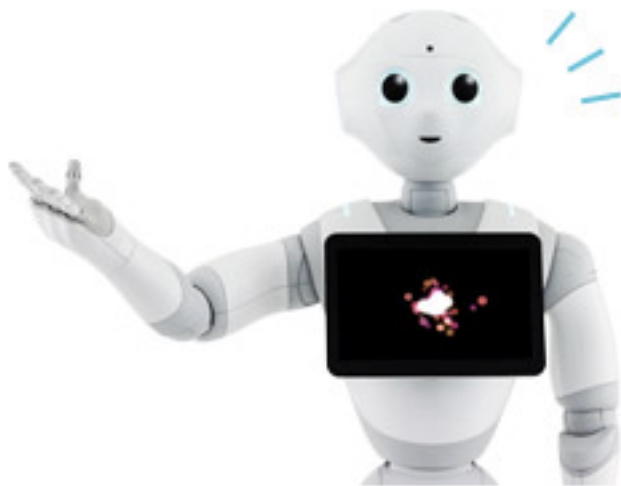
- 使うAPIによって色々な関数がある

class ALTextToSpeechProxy

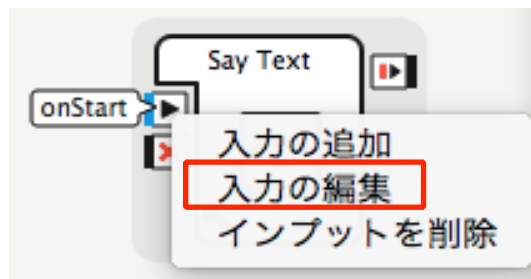
- ALTextToSpeechProxy::getAvailableLanguages()
- ALTextToSpeechProxy::getAvailableVoices()
- ALTextToSpeechProxy::getLanguage()
- ALTextToSpeechProxy::getParameter()
- ALTextToSpeechProxy::getSupportedLanguages()

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

パラメーター

- self.getParameter(“変数名”) で値を取得



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

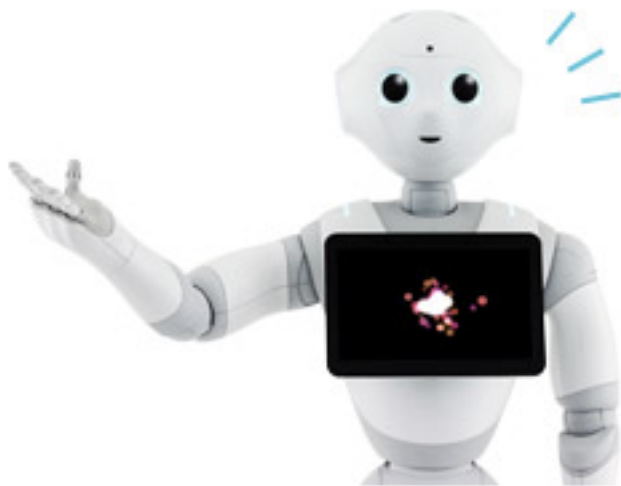
- self.変数名(値)で出力される



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
```


例外処理



例外処理

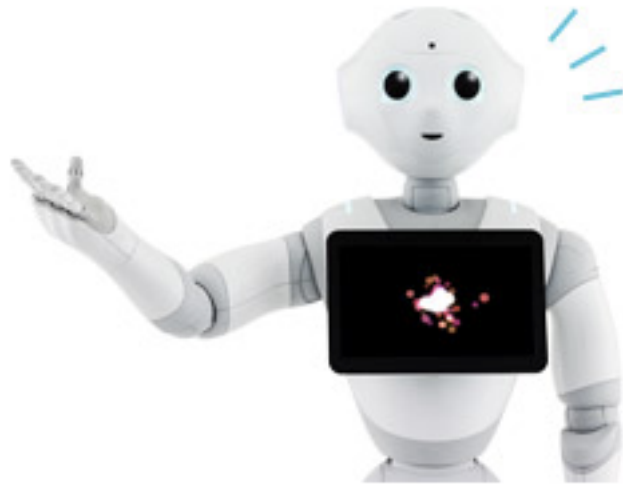
- 例外(実行中に検出されたエラー)処理コードを書くことで予期せぬ事故を未然に防ぐ
- try句：例外が起こりそうな処理
- except句：例外が起こった場合の処理
- else句：例外が検出されなかった場合の処理(except句の後ろにおく)
- finally句：どの場合にも実行する処理

～例～

```
try:  
    self.logger.info(x/y)  
except:  
    self.logger.info("Error!")  
else:  
    self.logger.info("Success!")  
finally:  
    self.logger.info("Completed!")
```

～実行結果～

```
x = 10,y = 0の時  
Error!  
Completed!  
  
x = 10,y = 2の時  
5  
Success!  
Completed!
```



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)
```

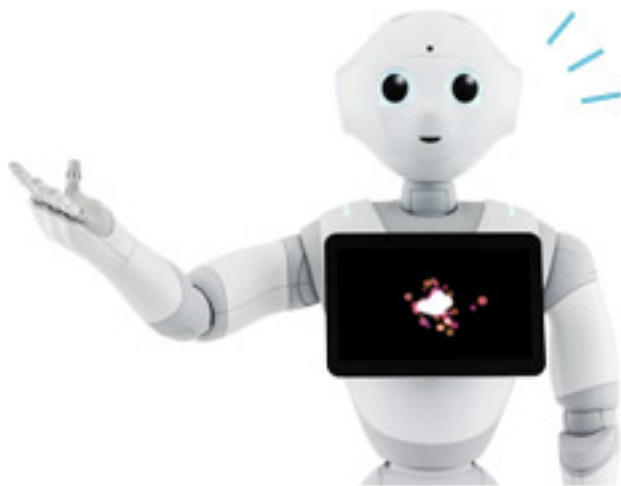
文字列に
変換する関数

変数の値を取得

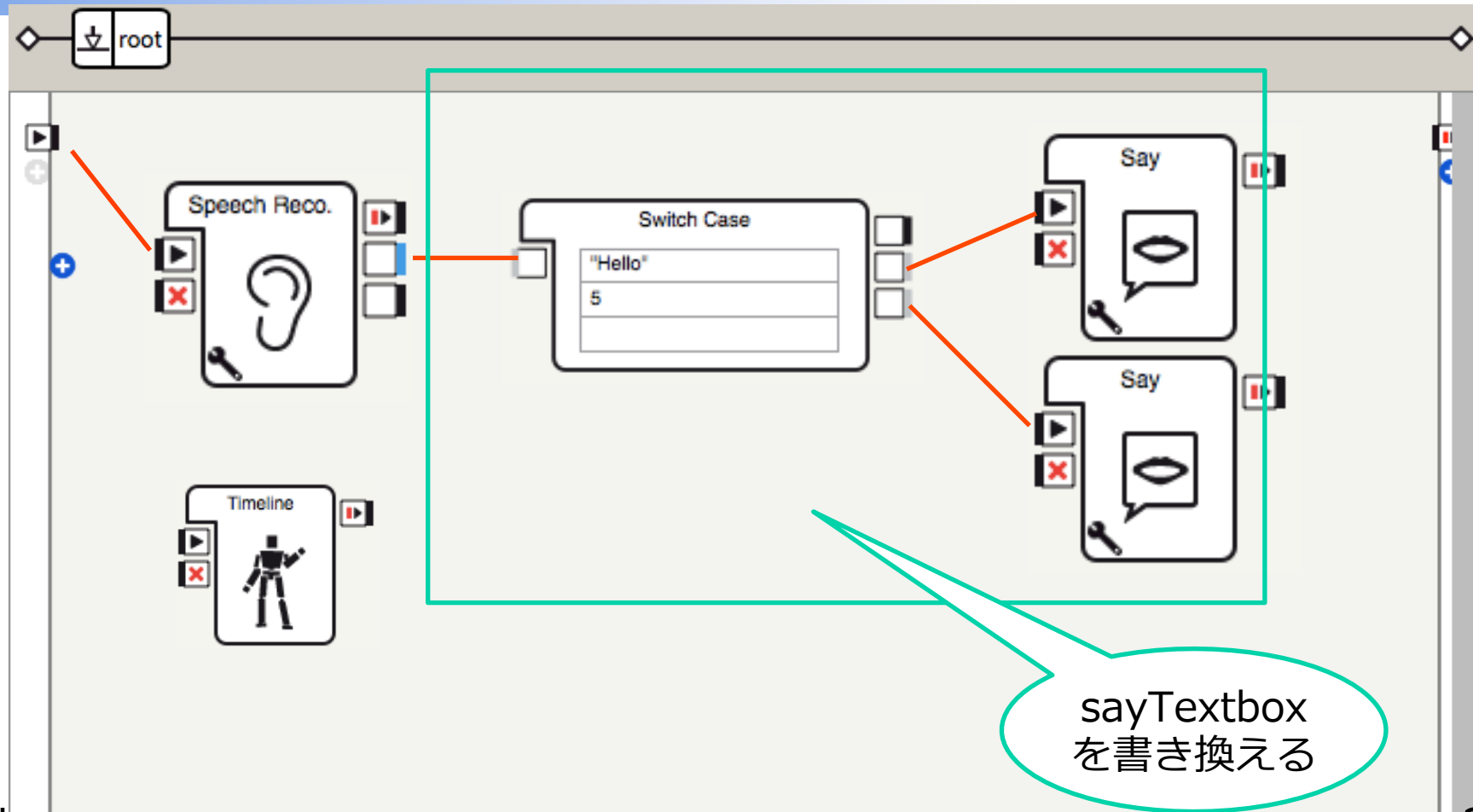
sentence = "\RSPD=100\ \VCT=100 \こんにちは\RST\"

ALTextToSpeechの
sayという関数を使用

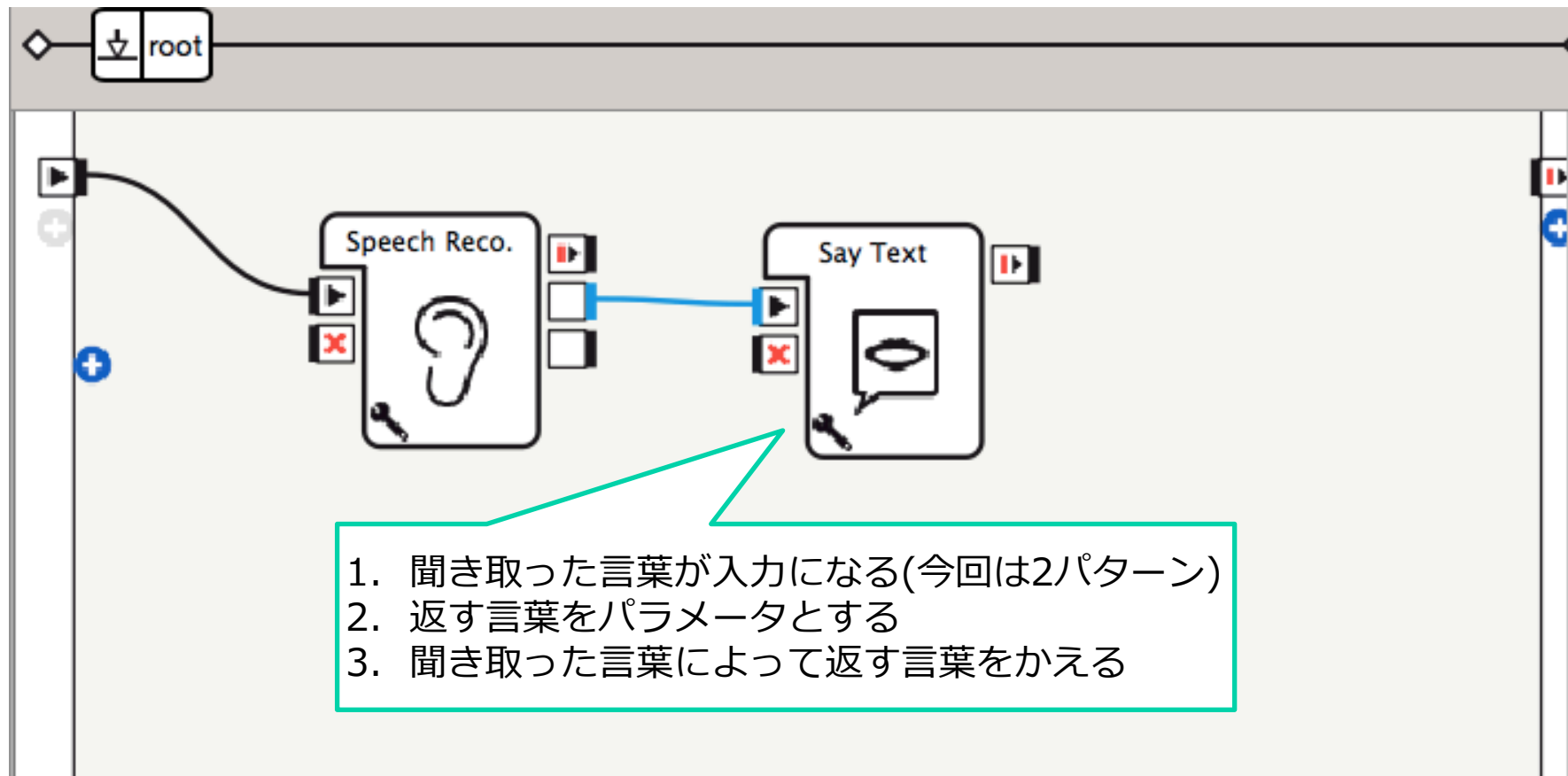
演習！
BOXを編集しよう



会話をするBOX before

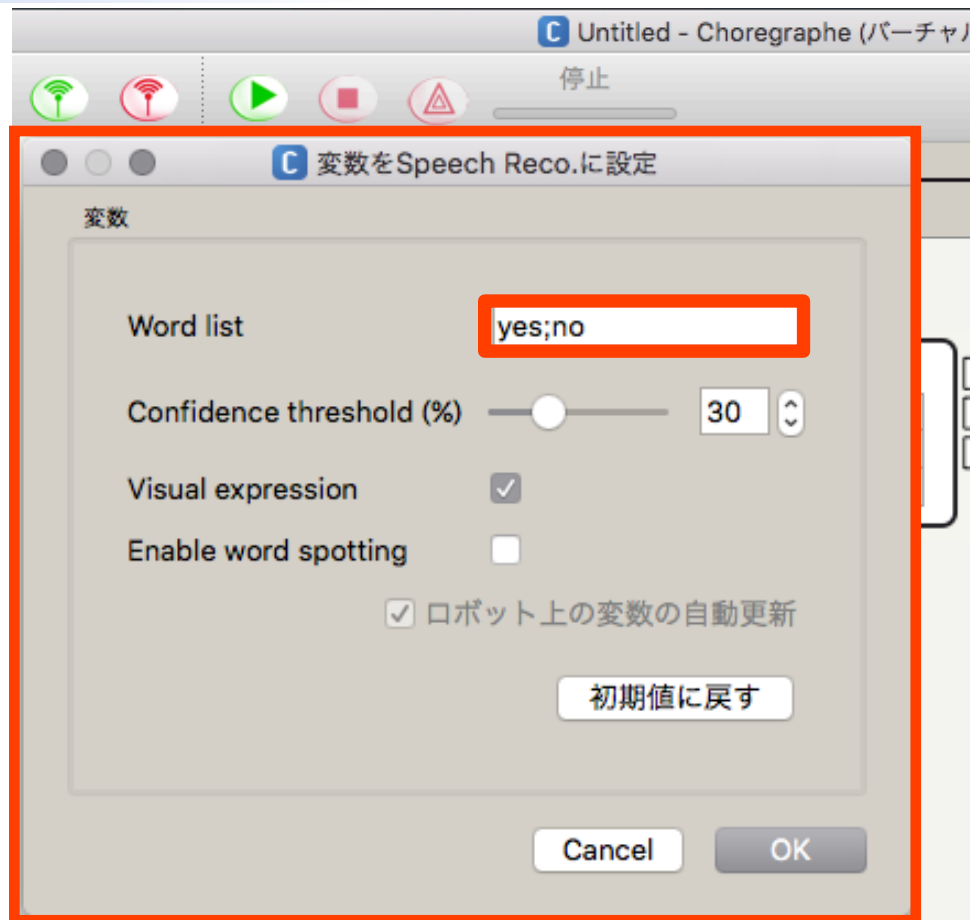


会話をするBOX after

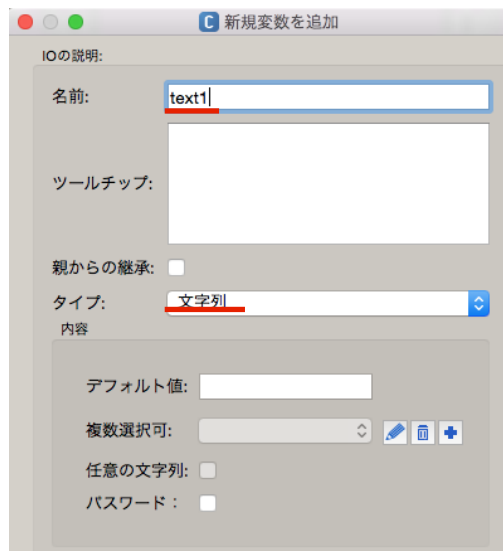
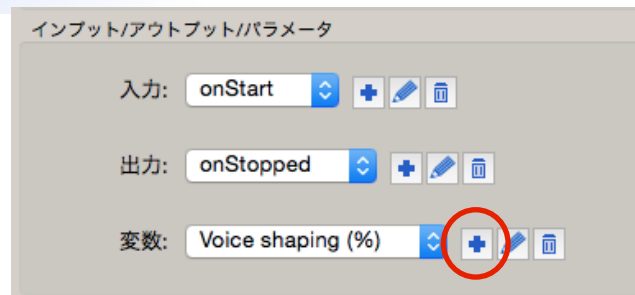
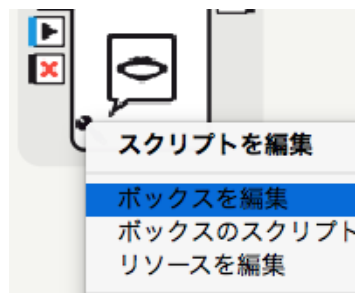


会話をする “Speech Reco.” の設定

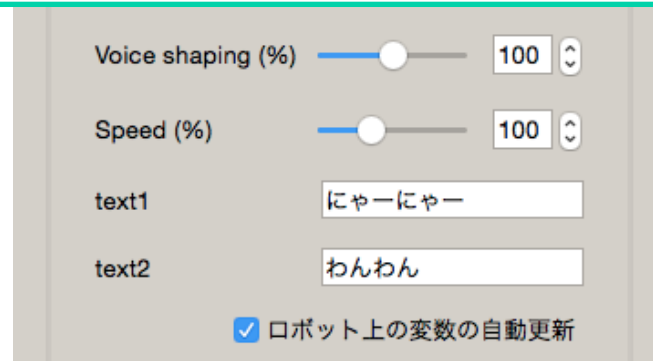
1. **Speech Reco** の
パラメーターボタンで設定
2. 聞かせたい言葉を **Wordlist** を設定
今回は「こんにちは;こんばんは」
3. **単語;単語** というように
半角セミコロンを挟めると
いくつもの単語を認識できる
(今回は2つ)



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)

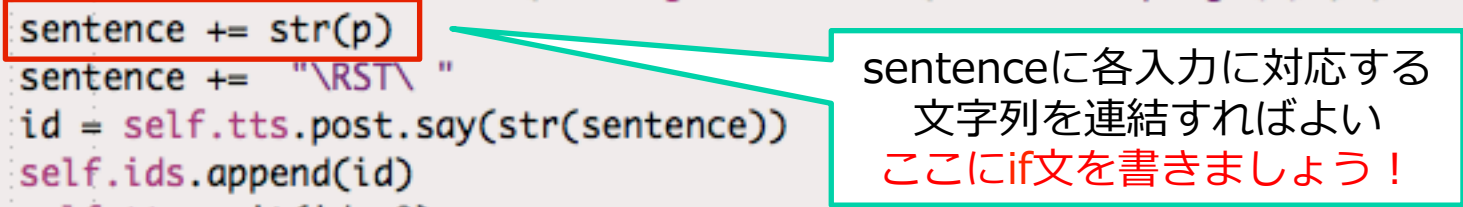


どこを書き換えたら良いか？

```
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
3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
    human = self.getParameter("human").split(";")
    pepper = self.getParameter("pepper").split(";")
    try:
        sentence = "\RSPD="+ str( self.getParameter("Speed (%)") ) + "\ "
        sentence += "\VCT="+ str( self.getParameter("Voice shaping (%)") ) + "\ "
        for i in range(len(human)):
            if p == human[i]:
                sentence += pepper[i]
        sentence += "\RST\ "
        id = self.tts.post.say(str(sentence))
        self.ids.append(id)
        self.tts.wait(id, 0)
    finally:
        try:
            self.ids.remove(id)
```


**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

